Knowledge Distillation with Geometric Approaches for Multimodal Data Analysis

by

Eun Som Jeon

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2023 by the
Graduate Supervisory Committee:

Pavan Turaga, Chair
Baoxin Li
Hyunglae Lee
Suren Jayasuriya

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

This thesis presents robust and novel solutions using knowledge distillation with geometric approaches and multimodal data that can address the current challenges in deep learning, providing a comprehensive understanding of the learning process involved in knowledge distillation. Deep learning has attained significant success in various applications, such as health and wellness promotion, smart homes, and intelligent surveillance. In general, stacking more layers or increasing the number of trainable parameters causes deep networks to exhibit improved performance. However, this causes the model to become large, resulting in an additional need for computing and power resources for training, storage, and deployment. These are the core challenges in incorporating such models into small devices with limited power and computational resources. In this thesis, robust solutions aimed at addressing the aforementioned challenges are presented. These proposed methodologies and algorithmic contributions enhance the performance and efficiency of deep learning models. The thesis encompasses a comprehensive exploration of knowledge distillation, an approach that holds promise for creating compact models from high-capacity ones, while preserving their performance. This exploration covers diverse datasets, including both time series and image data, shedding light on the pivotal role of augmentation methods in knowledge distillation. The effects of these methods are rigorously examined through empirical experiments. Furthermore, the study within this thesis delves into the efficient utilization of features derived from two different teacher models, each trained on dissimilar data representations, including time-series and image data. Through these investigations, I present novel approaches to knowledge distillation, leveraging geometric techniques for the analysis of multimodal data. These solutions not only address real-world challenges but also offer valuable insights and recommendations for modeling in new applications.

Dedicated to my parents Guho Jeon, Youngsun Kim, and my younger sister Eunbi Jeon, for their endless love and support. Thanks to their love, I am able to be here.

ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my supervisor Dr. Pavan Turaga. With his patience, wisdom, and faith, I have learned invaluable research skills and the wisdom needed to navigate life. His philosophy as a scholar and researcher has inspired me greatly and has been of great assistance in my professional growth. I will continue to engrave his teachings in my mind and never forget the gratitude, cherishing the time I was able to study alongside him.

I would like to thank my committee members, Dr. Baoxin Li, Dr. Hyunglae Lee, and Dr. Suren Jayasuriya. Thanks to your dedication of time and effort for the review of this research and your guidance and feedback, this study was able to progress further.

Especially, I want to express my gratitude to Dr. Kang Ryoung Park at Dongguk University for a lot of support and mentoring provided as I embark on my Ph.D. journey in the United States.

I express my heartfelt gratitude to all my labmates in the Geometric Media Lab (GML) at Arizona State University. Dr. Rushil Anirudh (LLNL), Dr. Suhas Lohit (MERL), and Dr. Anirudh Som (SRI), thank you for providing research guidance and the opportunity to collaborate. Dr. Hongjun Choi, I am grateful for your assistance in both research collaboration and helping me adapt well to life in the United States. Dr. Kowshik Thopalli, Dr. Rajhans Singh, Niccolo Meniconi, Sinjini Mitra, Jisoo Lee, and Dr. Ankita Shukla, thank you for the opportunity to have many conversations and spend time together both as friends and as research collaborators.

I would like to express my gratitude to the many friends I met in graduate school. I am thankful to all my friends who shared conversations, spent time together, celebrated achievements, and embarked on this journey with me. I gained a lot of courage and strength thanks to their support. And, I also want to express my gratitude to the

many long-time friends in Korea. Their support has been a great source of strength and determination for me.

Finally, I deeply thank my beloved parents and my younger sister. Thanks to their unconditional love, dedication, and support, I was able to achieve personal and academic growth and complete this thesis successfully. They have sacrificed so much for me and have always given me courage and strength. Expressing my gratitude beyond words, I dedicate this thesis to them.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

Table Page

Table Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Deep learning has achieved great success and widely utilized in many different applications, including computer vision (He *et al.* (2016); Huang *et al.* (2017)), speech recognition (Abdel-Hamid *et al.* (2014); Xiong *et al.* (2018)), and wearable sensor analysis (Wan *et al.* (2020); Fawaz *et al.* (2019)). Various architectures that go beyond convolutional methods have also been developed. However, to obtain better performance, a greater number of layers and parameters are utilized as a solution, resulting in increasing the complexity of networks and requiring large computational time and resources. Further, the demand for using deep models on small devices, such as mobile and IoT devices, has increased. This thesis studies and proposes robust methodologies to address these concerns with multimodal data using time-series and image data. Also, effective strategies to improve the performance of deep learning models in generating a small model are proposed.

1.1 Knowledge Distillation

To utilize lightweight forms of models, many studies have developed techniques for generating small models, such as network pruning (Molchanov *et al.* (2017); Han *et al.* (2016)), quantization (Han *et al.* (2016); Wu *et al.* (2016)), low-rank factorization (Tai *et al.* (2016)), and knowledge distillation (KD) (Hinton *et al.* (2015)) to compress deep learning models. Some of these methods aid in creating smaller deep learning models and reducing inference time on the edge device. However, post-training or fine-tuning techniques are generally applied to recover the lost classification performance, which

is cumbersome and can slow down development. (Han *et al.* (2016); Wu *et al.* (2016)). Whereas, KD does not require any additional processing, which saves time and costs on development and computational resources.

KD generates a small model by using the learned weights from a larger and more complex model. There are many variants of KD using the augmentation method, intermediate representations, and multiple trained models for better knowledge transfer. In order to maximize the performance of knowledge distillation, understanding the behavior of KD with different strategies, exploring the effects of the augmentation method in KD, and using an advanced method to extract better quality features are required. In Chapter 2, model performance in KD with time-series data is explored. In the chapter, strategies for using augmentation methods in KD and the optimal network choice are investigated. The efficacy of early-stopped teachers in KD on time-series data is also addressed, and the effects of several augmentation methods with different capacities of teachers are explained. The exploration helps provide a comprehensive understanding of the behavior of augmentation in KD and the relationships between properties for training and testing sets.

1.2 Geometric Approaches in Knowledge Distillation

Geometric approaches are to consider non-linearity and non-Euclidean processes. These aid in finding the optimal solution and solving distorted distance problems that cannot be solved by the conventional methods using linear distance or unimodal data. This can be applied to feature space for KD procedures, and then two combined methods can create great synergies to improve the performance significantly.

In this thesis, geometric approaches are utilized to improve distillation performance. In Chapter 3, a geometric method is utilized to obtain a better representation of intermediate features to improve the performance of distillation using image data.

In the chapter, features are first projected onto a hypersphere to compute the angular distribution. Based on angular features, the gap between positive and negative features is enlarged by inserting an angular margin, which aids in obtaining better quality features to improve the performance of knowledge transfer. In Chapter 4, a geometric approach is applied to obtain knowledge of the relationship between similarity features for distillation using multimodal data, including time-series and image data. To improve performance, topological data analysis (TDA) is adopted to generate persistence images from time-series data, representing topological features that have robustness under time-series perturbations. To accommodate different modalities (persistence image and raw time-series data), a new knowledge form is designed, leveraging feature relationships from orthogonal properties. The knowledge, reflecting feature relationships, is more expressive and disentangled than the original one, which helps distill the better student. The proposed method shows significant improvements over other baselines.

1.3 Multimodal Data Analysis with Multiple Teachers in Knowledge Distillation

To improve the performance of a model, multiple models with multimodal data have been utilized (Som *et al.* (2020)). Specifically, topological features from persistence images have been adopted in machine learning to complement time-series features and improve performance. However, these methods increase model complexity and computational resources. Even though KD using multiple teachers has been studied (Gou *et al.* (2021); Liu *et al.* (2020); Zhang *et al.* (2022)), there are still limitations because of the knowledge gap between teachers and students. Furthermore, most cases focus on unimodal data analysis.

In this thesis, to maximize the advantage and performance of KD, multiple teachers with multimodal data (time-series and image data) are incorporated to train a

single student using time-series data as an input alone. In Chapter 4, utilizing topological knowledge by a geometric method in KD is introduced. To use two teachers trained with different data, applying an annealing strategy in KD is proposed to consider different contributions and reduce the knowledge gap between them. To transfer knowledge effectively, instead of transferring features directly, orthogonality properties are utilized to extract feature relationships and learn a student model. In Chapter 5, diverse feature maps and a constrained adaptive weighting mechanism are proposed. In general, richer knowledge can be leveraged to improve the performance of a student in KD (Gou *et al.* (2021)). However, combining different statistical characteristics of features from different teachers and training a model in a unified framework are challenges. To extract useful features in distillation, batch and channel similarities within a mini-batch are utilized, which help to match different dimensional sizes of features. Further, to integrate features from different teachers effectively, a constrained weighting mechanism is developed to control the effects of teachers adaptively, which is computed based on the entropy values of their output logits. Through feature visualization, the differences in activation maps of teachers are shown, which implies a knowledge gap between them. In Chapter 6, uncertain feature rectification for KD is addressed. Previous studies introduced the idea of leveraging multiple teachers in KD (You *et al.* (2017); Kwon *et al.* (2020)); however, most of them focused on feature matching and did not consider inherent or different features from teachers. Even though teachers learn with different types of data, their target is to implement the same task. That is, when multiple teachers are incorporated into the KD learning process with a framework, they generate common and uncommon features simultaneously. Also, teachers are not always guaranteed to produce high-quality knowledge, improving the KD process. To improve the learning process of KD using different teachers, an uncertainty-aware feature rectification for

KD is devised. Firstly, features from teachers are separated into common or different features. And, based on the entropy loss values of teachers, features of teachers are rectified to transfer high-quality knowledge to a student. When teachers have different architectural networks, this framework shows significantly improved performance compared to baselines.

Through all chapters in this thesis, the effectiveness of the proposed methods is investigated in various aspects, such as feature map visualization, measuring expected calibration errors to evaluate the generalizability and reliability of models, and combining with existing algorithms to verify compatibility. Based on these analyses, I give recommendations for modeling in new applications on various datasets.

Finally, in Chapter 7, I conclude this thesis and highlight the future directions.

Chapter 2

ROLE OF DATA AUGMENTATION STRATEGIES IN KNOWLEDGE DISTILLATION FOR WEARABLE SENSOR DATA

2.1 Introduction

Deep Learning has achieved state-of-the-art performance in various fields, including computer vision (He *et al.* (2016); Huang *et al.* (2017); Dalal and Triggs (2005); Lowe (2004)), speech recognition (Abdel-Hamid *et al.* (2014); Xiong *et al.* (2018)), and wearable sensors analysis (Wan *et al.* (2020); Fawaz *et al.* (2019)). In general, stacking more layers or increasing the number of learnable parameters causes deep networks to exhibit improved performance (Huang *et al.* (2017); Dalal and Triggs (2005); Lowe (2004); Fawaz *et al.* (2019); Khan *et al.* (2020); Gil-Martín *et al.* (2020)). However, this causes the model to become large resulting in additional need for compute and power resources, for training, storage, and deployment. These challenges can hinder the ability to incorporate such models into edge devices. Many studies have explored techniques such as network pruning (Molchanov *et al.* (2017); Han *et al.* (2016)), quantization (Han *et al.* (2016); Wu *et al.* (2016)), low-rank factorization (Tai *et al.* (2016)), and Knowledge Distillation (KD) (Hinton *et al.* (2015)) to compress deep learning models. At the cost of lower classification accuracy, some of these methods help to make the deep learning model smaller and increase the speed of inference on the edge devices. Post-training or fine-tuning strategies can be applied to recover the lost classification performance (Han *et al.* (2016); Wu *et al.* (2016)). On the contrary, KD does not require fine-tuning nor is subjected to any post-training processes.

KD is a simple and popular technique that is used to develop smaller and efficient models by distilling the learnt knowledge/weights from a larger and more complex model. The smaller and larger models are referred to as student and teacher models, respectively. KD allows the student model to retain the classification performance of the larger teacher model. Recently, different variants of KD have been proposed (Yim *et al.* (2017); Heo *et al.* (2019)). These variations rely on different choices of network architectures, teacher models, and various features used to train the student model. Alongside, teacher models trained by early stopping for KD (ESKD) have been explored, which have helped improving the efficacy of KD (Cho and Hariharan (2019)). However, to the best of my knowledge, there is no previous study that explores the effects, challenges, and benefits of KD for human activity recognition using wearable sensor data.

In this chapter, I firstly study KD for human activity recognition from time-series data collected from wearable sensors. Secondly, I also evaluate the role of data augmentation techniques in KD. This is evaluated by using several time domain data augmentation strategies for training as well as for testing phase. The key highlights and findings from this study are summarized below:

- I compare and contrast several KD approaches for time-series data and conclude that EKSD performs better as compared to other techniques.

- I perform KD on time-series data with different sizes of teacher and student networks. I corroborate results from previous studies that suggest that the performance of a higher capacity teacher model is not necessarily better.

- I study the effects of data augmentation methods on both teacher and student models. I do this to identify which combination of augmentation methods give the most benefit in terms of classification performance.

- This study is evaluated on human activity recognition task and is conducted on a small scale publicly available dataset as well as a large scale dataset. This ensures the observations are reliable irrespective of the dataset sizes.

The rest of the chapter is organized as follows. In section 2.2, I provide a brief overview of KD techniques as well as data augmentation strategies. In section 2.3, I present which augmentation methods are used and its effects on time-series data. In section 2.4, I describe the experimental results and analysis. In section 2.5, I discuss the findings and conclusions.

2.2 Background

Knowledge Distillation

The goal of KD is to supervise a small student network by a large teacher network, such that the student network achieves comparable or improved performance over teacher model. This idea was firstly explored by Buciluǎ *et al.* (Buciluǎ *et al.* (2006)) followed by several developments like Hinton *et al.* (Hinton *et al.* (2015)). The main idea of KD is to use the soft labels which are outputs, soft probabilities, of a trained teacher network and contain more information than just a class label, which is illustrated in Figure 2.1. For instance, if two classes have high probabilities for a data, the data has to lie close to a decision boundary between these two classes. Therefore, mimicking these probabilities helps student models to get knowledge of teachers that have been trained with labeled data (hard labels) alone.

During training, the loss function $\mathcal{L}$ for a student network is defined as:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\mathcal{C}} + \lambda\mathcal{L}_{\mathcal{K}} \tag{2.1}$$

where $\mathcal{L}_{\mathcal{C}}$ is the standard cross entropy loss, $\mathcal{L}_{\mathcal{K}}$ is KD loss, and λ is hyper-parameter; $0 < \lambda < 1$.

In supervised learning, the error between the output of the softmax layer of a student network and ground-truth label is penalized by the cross-entropy loss:

$$\mathcal{L}_{\mathcal{C}} = \mathcal{H}(softmax(a_s), y_g) \tag{2.2}$$

where $\mathcal{H}(\cdot)$ denotes a cross entropy loss function, a_s is logits of a student (inputs to the final softmax), and y_g is a ground truth label. In the process of KD, instead of using peaky probability distributions which may produce less accurate results, Hinton *et al.* (Hinton *et al.* (2015)) proposed to use probabilities with temperature scaling, *i.e.*, output of a teacher network given by $f_t = softmax(a_t/\tau)$ and a student $f_s = softmax(a_s/\tau)$ are softened by hyperparameter τ, where $\tau > 1$. The teacher and student try to match these probabilities by a KL-divergence loss:

$$\mathcal{L}_{\mathcal{K}} = \tau^2 KL(f_t, f_s) \tag{2.3}$$

where $KL(\cdot)$ is the KL-divergence loss function.

There has been lots of approaches to improve the performance of distillation. Previous methods focus on adding more losses on intermediate layers of a student network to be closer to a teacher (Zagoruyko and Kmodakis (2017); Tung and Mori (2019)). Averaging consecutive student models tends to produce better performance of students (Tarvainen and Valpola (2017)). By implementing KD repetitively, the performance of KD is improved, which is called sequential knowledge distillation (Zhang *et al.* (2018b)).

Recently, learning procedures for improved efficacy of KD has been presented. Goldblum *et al.* (Goldblum *et al.* (2020)) suggested adversarially robust distillation (ARD) loss function by minimizing dependencies between output features of a teacher. The method used perturbed data as adversarial data to train the student network. Interestingly, ARD students even show higher accuracy than their teacher.

I adopt augmentation methods to create data which is similar to adversarial data of ARD. Based on ARD, the effect of using adversarial data for KD can be verified, however, which data augmentation is useful for training KD is not well explored. Unlike ARD, to figure out the role of augmentation methods for KD and which method improves the performance of KD, I use augmentation methods generating different kinds of transformed data for teachers and students. In detail, by adopting augmentation methods, I can generate various combinations of teachers and students which are trained with the same or different augmentation method. It provides to understand which transformation and combinations can improve the performance of KD. I explain the augmentation method for KD in section 2.3 with details. Additionally, KD tends to show an efficacy with transferring information from early stopped model of a teacher, where training strategy is called ESKD (Cho and Hariharan (2019)). Early stopped teachers produce better students than the standard knowledge distillation (Full KD) using fully-trained teachers. Cho *et al.* (Cho and Hariharan (2019)) presented the efficacy of ESKD with image datasets. I implement ESKD on time-series data and investigate its efficacy on training with data transformed by various augmentation methods. I explain more details in section 2.3 and discuss the efficiency of ESKD in later sections.

In general, many studies focus on the structure of networks and adding loss functions to existing framework of KD (Furlanello *et al.* (2018); Yang *et al.* (2019)). However, the performance of most approaches depends on the capacity of student models. Also, availability of sufficient training data for teacher and student models can affect to the final result. In this regard, the factors that have an affect on the distillation process need to be systematically explored, especially on time-series data from wearable sensors.

Data Augmentation

Data augmentation methods have been used to boost the generalizability of models and avoid over-fitting. They have been used in many applications such as time-series forecasting (Han *et al.* (2019)), anomaly detection (Chalapathy and Chawla (2019)), classification (Fawaz *et al.* (2019); Le Guennec *et al.* (2016)), and so on. There are many data augmentation approaches for time-series data, which can be broadly grouped under two categories (Wen *et al.* (2020)). The first category consists of transformations in time, frequency, and time-frequency domains (Wen *et al.* (2020); Park *et al.* (2019a)). The second group consists of more advanced methods like decomposition (Kegel *et al.* (2018)), model-based (Cao *et al.* (2014)), and learning-based methods (Esteban *et al.* (2017); Wen *et al.* (2020)).

Time-domain augmentation methods are straightforward and popular. These approaches directly manipulate the original input time-series data. For example, the original data is transformed directly by injecting Gaussian noise or other perturbations such as step-like trend and spikes. Window cropping or sloping also has been used in time domain transformation, which is similar to computer vision method of cropping samples (Cui *et al.* (2015)). Other transformations include window warping that compresses or extends a randomly chosen time range and flipping the signal in time-domain. Additionally, one can use blurring and perturbations in the data points, especially for anomaly detection applications (Gao *et al.* (2020)). A few approaches have focused on data augmentation in the frequency domain. Gao *et al.* (Gao *et al.* (2020)) proposed perturbations for data augmentation in frequency domain, which improves the performance of anomaly detection by convolutional neural networks. The performance of classification was found to be improved by amplitude adjusted Fourier transform and iterated amplitude adjusted Fourier transform

which are transformation methods in frequency domain (Eileen *et al.* (2019)). Time-frequency augmentation methods have also been recenlty investigated. SpecAugment is a Fourier-transform based method that transforms in Mel-Frequency for speech time-series data (Park *et al.* (2019a)). The method was found to improve the performance of speech recognition. In (Steven Eyobu and Han (2018)), a short Fourier transform is proposed to generate a spectrogram for classification by LSTM neural network.

Decomposition-based, model-based, and learning-based methods are used as advanced data augmentation methods. For decomposition, time-series data are disintegrated to create new data (Kegel *et al.* (2018)). Kegel *et al.* firstly decomposes the time-series based on trend, seasonality, and residual. Then, finally new time-series data are generated with a deterministic and a stochastic component. Bootstrapping methods on the decomposed residuals for generating augmented data was found to help the performance of a forecasting model (Bergmeir *et al.* (2016)). Model-based approaches are related to modeling the dynamics, using statistical model (Cao *et al.* (2014)), mixture models (Kang *et al.* (2020)), and so on. In (Cao *et al.* (2014)), model-based method were used to address class imbalance for time-series classification. Learning-based methods are implemented with learning frameworks such as generative adversarial nets (GAN) (Esteban *et al.* (2017)) and reinforcement learning (Cubuk *et al.* (2019)). These methods generate augmented data by pre-trained models and aim to create realistic synthetic data (Esteban *et al.* (2017); Cubuk *et al.* (2019)).

Finally, augmentation methods can be combined together and applied simultaneously to the data. Combining augmentation methods in time-domain helps to improve performance in classification (Um *et al.* (2017)). However, combining various aug-

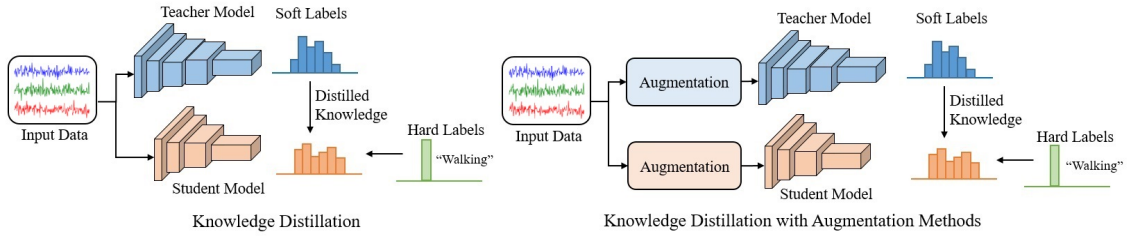


Figure 2.1: An Overview of Knowledge Distillation Framework (Left) and Proposed Knowledge Distillation with Data Augmentation Method (Right). A Capacity Network Known as Teacher Is Used to Guide the Learning of a Smaller Network Known as Student.

mentation methods may results in a large amount of augmented data, increasing training-time, and may not always improve the performance (Wen *et al.* (2020)).

2.3 Strategies for Knowledge Distillation with Data Augmentation

I would like to investigate strategies for training KD with time-series data and identify augmentation methods for teachers and students that can provide better performance. The strategies include two scenarios on KD. Firstly, I apply augmentation methods only when a student model is trained based on KD with a teacher model trained by the original data. Secondly, augmentation methods are applied not only to students, but also to teacher. When a teacher model is trained from scratch, an augmentation method is used, where the model is to be used as a pre-trained model for distillation. And, when a student is trained on KD, the same/different augmentation methods are used. The set of augmentation approaches on KD are illustrated in Figure 2.1, and described in further detail later in this section. Also, I explore the effects of ESKD on time-series data – ESKD uses a teacher which is obtained in the early training process. ESKD generates better students rather than using the fully-trained teachers from Full KD (Cho and Hariharan (2019)). The strategy is derived from

the fact that the accuracy is improved initially. However, the accuracy towards the end of training begins to decrease, which is lower than the earlier accuracy. I adopt early stopped teachers with augmentation methods for the experiments presented in section 2.4.

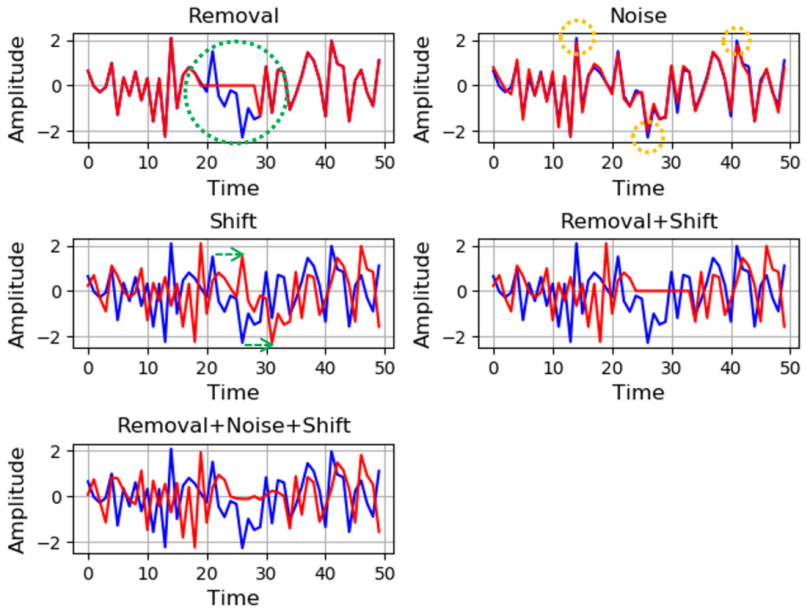


Figure 2.2: Illustration of Different Augmentation Methods Used in the Knowledge Distillation Framework. The Original Data Is Shown in Blue and the Corresponding Transformed Data with Data Augmentation Method Is Shown in Red.

In order to see effects of augmentation on distillation, I adopt time-domain augmentation methods which are removal, adding noise with Gaussian noise, and shifting. The original pattern, length of the window, and periodical points can be preserved by this transformation. I use transformation methods in time domain so that I can analyze the results from each method, and combinations, more easily. These methods also have been used popularly for training deep learning networks (Wen *et al.* (2020)). I apply combinations of augmentation methods, combined with removal and shifting, and with all methods to a data to see the relationships between each property of

datasets for teachers and students of KD. An example of different transformation used for data augmentation is shown in Figure 2.2. I describe each of the transforms below:

- **Removal:** is used to erase amplitude values of sequential samples. The values of chosen samples to be erased are transformed to the amplitude of the first point. For example, I assume that n samples are chosen as $(X_{t+1}, X_{t+2}, \cdots, X_{t+n})$ and their amplitudes are $(A_{t+1}, A_{t+2}, \cdots, A_{t+n})$ to be erased. A_{t+1} is the amplitude of the first sample X_{t+1} and is assigned to $(A_{t+1}, A_{t+2}, \cdots, A_{t+n})$. That is, values $(A_{t+1}, A_{t+2}, \cdots, A_{t+n})$ are mapped to $(A_{t+1}, A_{t+1}, \cdots, A_{t+1})$. The first point and the number of samples to be erased are chosen randomly. The result of removal is shown in Figure 2.2 with a green dashed circle.

- **Noise Injection:** To inject noise, I apply Gaussian noise with mean 0 and a random standard deviation. The result of adding noise is shown in Figure 2.2 with yellow dashed circles.

- **Shifting:** For shifting data, to keep the characteristics such as values of peak points and periodic patterns in the signal, I adopt index shifting and rolling methods to the data for generating new patterns, which means the 100% shifted signal from the original signal by this augmentation corresponds to the original one. For example, assuming the total number of samples are 50 and 10 time-steps (20% of the total number of samples) are chosen to be shifted. The values for amplitude of samples $(X_1, X_2, \cdots, X_{11}, \cdots X_{50})$ are $(A_1, A_2, \cdots, A_{11}, \cdots, A_{49}, A_{50})$. By shifting 10 time-steps, $(A_{41}, A_{42}, \cdots, A_1, \cdots, A_{39}, A_{40})$ are newly assigned to the samples of $(X_1, X_2, \cdots, X_{11}, \cdots, X_{49}, X_{50})$. The number of time-steps to be shifted is chosen randomly. Shifting is shown in Figure 2.2 with green dashed arrows.

- **Mix1:** Applies removal as well as shifting to the same data.
- **Mix2:** Applies removal, Gaussian noise injection, and shifting simultaneously to the data.

2.4 Experiments and Analysis

In this section, I describe datasets, settings, ablations, and results of the experiments.

2.4.1 Dataset Description

I perform experiments on two datasets: GENEActiv (Wang *et al.* (2016)) and PAMAP2 (Reiss and Stricker (2012)), both of which are wearable sensors based activity datasets. I evaluate multiple teachers and students of various capacities for KD with data augmentation methods.

GENEActiv

GENEActiv dataset (Wang *et al.* (2016)) consists of 29 activities over 150 subjects. The dataset was collected with a GENEActiv sensor which is a light-weight, waterproof, and wrist-worn tri-axial accelerometer. The sampling frequency of the sensors is 100Hz. In this experiments, I used 14 activities which can be categorized as daily activities such as walking, sitting, standing, driving, and so on. Each class has over approximately 900 data samples and the distribution and details for activities are illustrated in Figure 2.3. I split the dataset for training and testing with no overlap in subjects. The number of subjects for training and testing are over 130 and 43, respectively. A window size for a sliding window is 500 time-steps or 5 seconds and the process for temporal windows is full-non-overlapping sliding windows. The number of windows for training is approximately 16000 and testing is 6000.

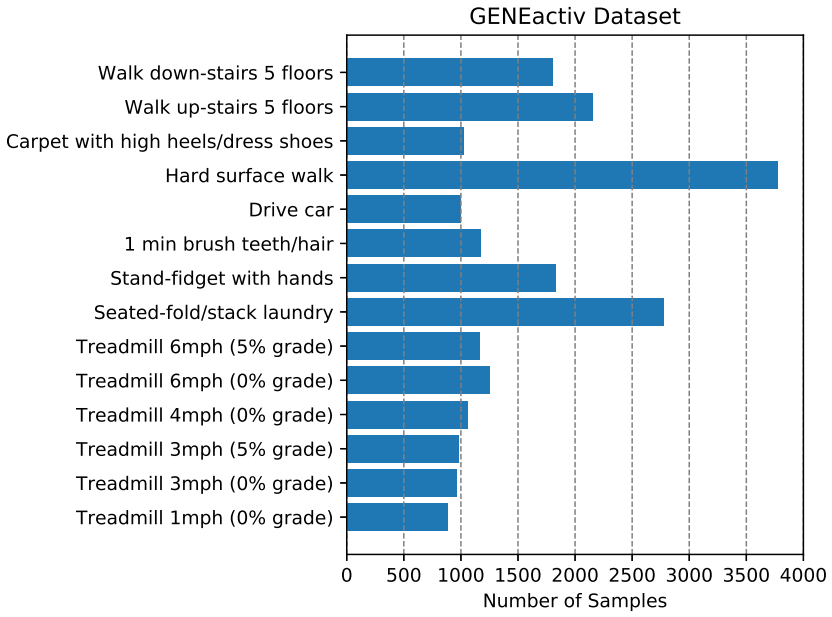


Figure 2.3: Distribution of GENEActiv Data Across Different Activities. Each Sample Has 500 Time-steps.

PAMAP2

PAMAP2 dataset (Reiss and Stricker (2012)) consists of 18 physical activities for 9 subjects. The 18 activities are categorized as 12 daily activities and 6 optional activities. The dataset was obtained by measurements of heart rate, temperature, accelerometers, gyroscopes, and magnetometers. The sensors were placed on hands, chest, and ankles of the subject. The total number of dimensions in the time-series is 54 and the sampling frequency is 100Hz. To compare with previous methods, in experiments on this dataset, I used leave-one-subject-out combination for validation comparing the i^{th} subject with the i^{th} fold. The input data is in the form of time-series from 40 channels of 4 IMUs and 12 daily activities. To compare with previous methods, the recordings of 4 IMUs are downsampled to 33.3Hz. The 12 action classes are: lying, sitting, standing, walking, running, cycling, nordic walking, ascending

stairs, descending stairs, vacuum cleaning, ironing, and rope jumping. Each class and subject are described in Table 2.1. There is missing data for some subjects and the distribution of the dataset is imbalanced. A window size for a sliding window is 100 time-steps or 3 seconds and step size is 22 time-steps or 660 ms for segmenting the sequences, which allows semi-non-overlapping sliding windows with 78% overlapping (Reiss and Stricker (2012)).

Table 2.1: Details of PAMAP2 Dataset. The Dataset Consists of 12 Activities Recorded for 9 Subjects.

	Sbj.101	Sbj.102	Sbj.103	Sbj.104	Sbj.105	Sbj.106	Sbj.107	Sbj.108	Sbj.109	Sum	Nr. of subjects
Lying	407	350	329	344	354	349	383	361	0	2877	8
Sitting	352	335	432	381	402	345	181	342	0	2770	8
Standing	325	383	307	370	330	365	385	377	0	2842	8
Walking	333	488	435	479	481	385	506	474	0	3481	8
Running	318	135	0	0	369	341	52	246	0	1461	6
Cycling	352	376	0	339	368	306	339	382	0	2462	7
Nordic walking	302	446	0	412	394	400	430	433	0	2817	7
Ascending stairs	233	253	147	243	207	192	258	168	0	1701	8
Descending stairs	217	221	218	206	185	162	167	137	0	1513	8
Vacuum cleaning	343	309	304	299	366	315	322	364	0	2622	8
Ironing	353	866	420	374	496	568	442	496	0	3995	8
Rope jumping	191	196	0	0	113	0	0	129	92	721	6

2.4.2 Analysis of Distillation

For experiments on GENEActiv, I run 200 epochs for each model using SGD with momentum 0.9 and the initial learning rate $lr = 0.1$. The lr drops by 0.5 after 10 epochs and drops down by 0.1 every $[\frac{t}{3}]$ where t is the total number of epochs. For experiments on PAMAP2, I run 180 epochs for each model using SGD with momentum 0.9 and the initial learning rate $lr = 0.05$. The lr drops down by

0.2 after 10 epochs and drops down 0.1 every $[\frac{t}{3}]$ where t is the total number of epochs. The results are averaged over 3 runs for both the datasets. To improve the performance, feature engineering (Zheng and Casari (2018); Bengio *et al.* (2013)), feature selection, and reducing confusion by combining classes (Dutta *et al.* (2016)) can be applied additionally. However, to focus on the effects of KD which is based on feature-learning (Bengio *et al.* (2013)), feature engineering/selection methods to boost performance are not applied and all classes as specified in section 2.4.1 are used in the following experiments.

Training from scratch to find a Teacher

Table 2.2: Accuracy for Various Models Trained from Scratch on GENEActiv

Model	# Parameters	Accuracy (%)	Model	# Parameters	Accuracy (%)	Model	# Parameters	Accuracy (%)
ResNet18(8)	62,182	63.75±0.42	WRN16-1	61,374	67.66±0.37	WRN28-1	126,782	68.63±0.48
ResNet18(16)	244,158	65.84±0.69	WRN16-2	240,318	67.84±0.36	-	-	-
ResNet18(24)	545,942	66.47±0.21	WRN16-3	536,254	68.89±0.56	WRN28-2	500,158	69.15±0.24
ResNet18(32)	967,534	66.33±0.12	WRN16-4	949,438	69.00±0.22	WRN28-3	1,119,550	69.23±0.27
ResNet18(48)	2,170,142	68.13±0.22	WRN16-6	2,127,550	70.04±0.05	WRN28-4	1,985,214	69.29±0.51
ResNet18(64)	3,851,982	68.17±0.21	WRN16-8	3,774,654	69.02±0.15	WRN28-6	4,455,358	70.99±0.44

To find a teacher for KD, I conducted experiments with training from scratch based on two different network architectures: ResNet (He *et al.* (2016)) and WideResNet (Zagoruyko and Komodakis (2016)). These networks have been popularly used in various state-of-the-art studies for KD (Yim *et al.* (2017); Heo *et al.* (2019); Goldblum *et al.* (2020); Cho and Hariharan (2019)). I modified and compared the structure having the similar number of trainable parameters. As described in Table 2.2, for training from scratch, WideResNet (WRN) tends to show better performance than

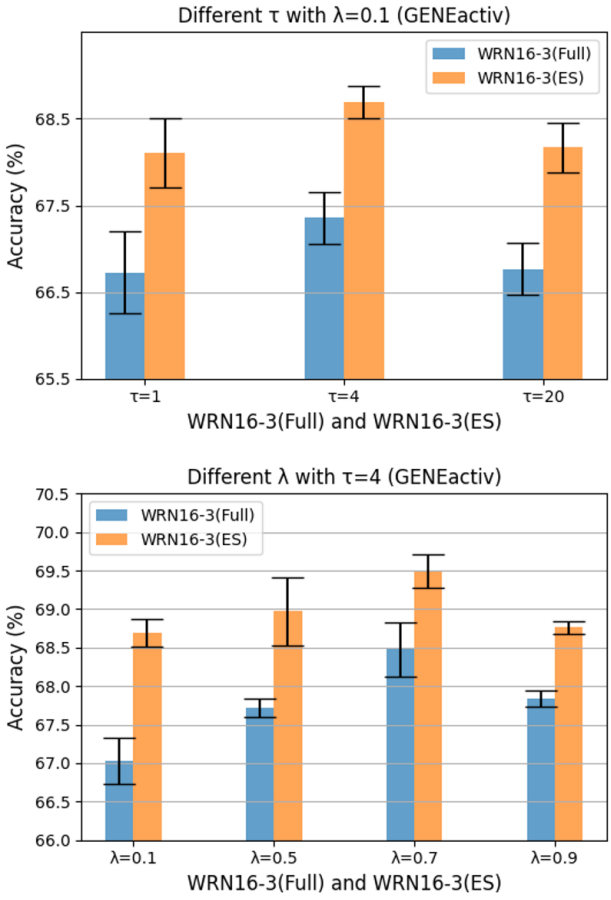


Figure 2.4: Effect of Hyperparmeters τ and λ on the Performance of Full KD and ESKD Approaches. The Results Are Reported on GENEActiv Dataset with WRN16-3 and WRN16-1 Networks for Teacher and Student Models Respectively.

ResNet18(k) where k is the dimension of output from the first layer. The increase in accuracy with the dimension of each block is similar to the basic ResNet.

Setting hyperparameters for KD

For setting hyperparameters in KD, I conducted several experiments with different temperature τ as well as lambda λ. I investigated distillation with different hyperparameters as well. I set WRN16-3 as a teacher network (Cho and Hariharan (2019))

and WRN16-1 as a student network, which is shown in Figure 2.4. For temperature τ, in general, $\tau \in \{3, 4, 5\}$ are used (Cho and Hariharan (2019)). High temperature mitigated the peakiness of teachers and helped to make the signal to be softened. In this experiments, according to the results from different τ, high temperature did not effectively help to increase the accuracy. When I used $\tau = 4$, the results were better than other choices for both datasets with Full KD and ESKD (Cho and Hariharan (2019)). For $\lambda = 0.7$ and 0.99, I obtained the best results with Full KD and ESKD for GENEActiv and PAMAP2, respectively.

Analyzing Distillation with different size of Models

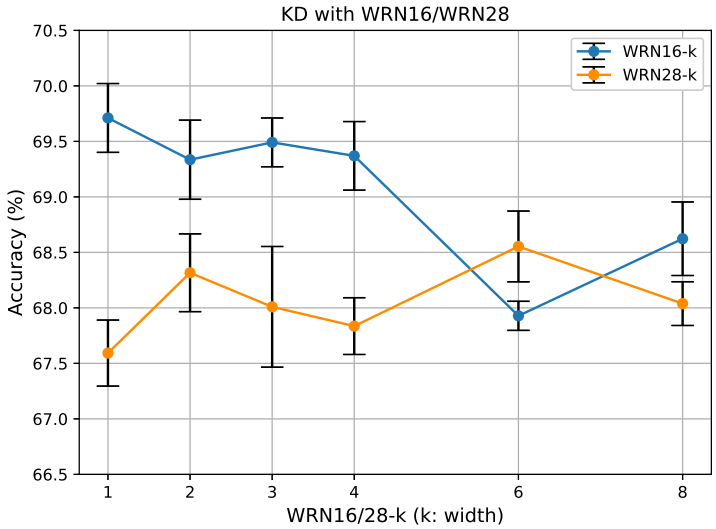


Figure 2.5: Results of Distillation from Different Teacher Models of WRN16-k and WRN28-k on GENEActiv Dataset. The Higher Capacity of Teachers Does Not Always Increase the Accuracy of Students.

To analyze distillation with different size of models, WRN16-k and WRN28-k were used as teacher networks having different capacity and structures in depth and width k. WRN16-1 and WRN28-1 were used as student networks, respectively. As

mentioned in the previous section, in general, a higher capacity network trained from scratch shows better accuracy for WRN16 and WRN28. However, as shown in Figure 2.5, in most of the cases, the results from WRN16-k shows better than the results of WRN28-k which has larger width. And the accuracy with teachers of WRN16-3 is higher than the one with teachers having larger width. Therefore, a teacher of higher capacity is not always guaranteed to generate a student whose accuracy is better.

Knowledge Distillation based on Fully Iterated and Early Stopped Models

Table 2.3: Accuracy for Various Models on GENEActiv Dataset

Student	Teacher	Teacher Acc. (%)	Student Acc. (%)
	WRN16-2	69.06	69.34±0.36
	WRN16-3	69.99	**69.49**±0.22
WRN16-1	WRN16-4	69.80	69.37±0.31
(ESKD)	WRN16-6	70.24	67.93±0.13
	WRN16-8	70.19	68.62±0.33
WRN16-1	WRN16-3	69.68	68.62±0.22
(Full KD)	WRN16-8	69.28	68.68±0.17

I performed additional experiments with WRN16-k which gives the best results. Table 2.3 and Table 2.4 give detailed results for GENEActiv and PAMAP2, respectively. Compared to training from scratch, although the student capacity from KD is much lower, the accuracy is higher. For instance, for the result of GENEActiv with WRN16-8 by training from scratch, the accuracy is 69.02% and the number of trainable parameters is 3 million in Table 2.3. The number of parameters for WRN16-1 as a student for KD is 61 thousand which is approximately 1.6% of 3 million. How-

Table 2.4: Accuracy for Various Models on PAMAP2 Dataset

Student	Teacher	Teacher Acc. (%)	Student Acc. (%)
	WRN16-2	84.86	86.18±2.44
	WRN16-3	85.67	**86.38**±2.25
WRN16-1	WRN16-4	85.23	85.95±2.27
(ESKD)	WRN16-6	85.51	86.37±2.35
	WRN16-8	85.17	85.11±2.46
WRN16-1	WRN16-3	81.52	84.31±2.24
(Full KD)	WRN16-8	81.69	83.70±2.52

ever, the accuracy of a student with WRN16-2 teacher from ESKD is 69.34% which is higher than the result of training from scratch with WRN16-8. It shows a model can be compressed with conserved or improved accuracy by KD. Also, I tested with 7 classes on GENEActiv dataset which were used by the method in (Choi *et al.* (2018)). This work used over 50 subjects for testing set. Students of KD were WRN16-1 and trained with $\tau = 4$ and $\lambda = 0.7$. As shown in Table 2.5 where brackets denote the structure of teachers and their accuracy, ESKD from WRN16-3 teacher shows the best accuracy for 7 classes, which is higher than results of models trained from scratch, Full KD, and previous methods (Cortes and Vapnik (1995); Choi *et al.* (2018)). In most of the cases, students are even better than their teacher. In various sets of GENEActiv having different number of classes and window length, ESKD shows better performance than Full KD. In Table 2.4, the best accuracy on PAMAP2 is 86.38% from ESKD with teacher of WRN16-3, which is higher than results from Full KD. The result is even better than previous methods (Jordao *et al.* (2018)), which are described in Table 2.6 where brackets denote the structure of teachers and their ac-

curacy. Therefore, KD allows model compression and improves the accuracy across datasets. And ESKD tends to show better performance compared to Full KD. Also, the higher capacity models as teachers does not always generate better performing student models.

Table 2.5: Accuracy (%) for Related Methods on GENEActiv Dataset with 7 Classes

Method	Window length	
	1000	500
WRN16-1	89.29±0.32	86.83±0.15
WRN16-3	89.53±0.15	87.95±0.25
WRN16-8	89.31±0.21	87.29±0.17
ESKD (WRN16-3)	**89.88**±0.07 (89.74)	**88.16**±0.15 (88.30)
ESKD (WRN16-8)	89.58±0.13 (89.68)	87.47±0.11 (87.75)
Full KD (WRN16-3)	89.84±0.21 (88.95)	87.05±0.19 (86.02)
Full KD (WRN16-8)	89.36±0.06 (88.74)	86.38±0.06 (85.08)
SVM (Cortes and Vapnik (1995))	86.29	85.86
Choi *et al.* (Choi *et al.* (2018))	89.43	87.86

2.4.3 Effect of Augmentation on Student Model Training

To understand distillation effects based on the various capacity of teachers and augmentation methods, WRN16-1, WRN16-3, and WRN16-8 are selected as "Small",

Table 2.6: Accuracy for Related Methods on PAMAP2 Dataset

Method	Accuracy (%)
WRN16-1	82.81±2.51
WRN16-3	84.18±2.28
WRN16-8	83.39±2.26
ESKD (WRN16-3)	**86.38**±2.25 (85.67)
ESKD (WRN16-8)	85.11±2.46 (85.17)
Full KD (WRN16-3)	84.31±2.24 (81.52)
Full KD (WRN16-8)	83.70±2.52 (81.69)
Chen and Xue (2015)	83.06
Ha *et al.* (2015)	73.79
Ha and Choi (2016)	74.21
Kwapisz *et al.* (2011)	71.27
Catal *et al.* (2015)	85.25
Kim *et al.* (2012)	81.57

"Medium", and "Large" models, respectively. ESKD is used for this experiment which tends to show better performance than the Full KD and requires three-fourths of the total number of epochs for training (Cho and Hariharan (2019)).

In order to find augmentation methods impacting KD on students for training, I first trained a teacher from scratch with the original datasets. Secondly, I trained stu-

dents from the pre-trained teacher with augmentation methods which have different properties including removal, adding noise, shifting, Mix1, and Mix2. For experiments on GENEActiv, for removal, the number of samples to be removed is less than 50% of the total number of samples. The first point and the exact number of samples to be erased are chosen randomly. To add noise, the value for standard deviation of Gaussian noise is chosen uniformly at random between 0 and 0.2. For shifting, the number of time-steps to be shifted is less than 50% of the total number of samples. For Mix1 and Mix2, the same parameters are applied. For experiments on PAMAP2, the number of samples for removal is less than 10% of the total number of samples and standard deviation of Gaussian noise for adding noise is less than 0.1. The parameter for shifting is less than 50% of the total number of samples. The same parameters of each method are applied for Mix1 and Mix2. The length of the window for PAMAP2 is only 100 which is 3 seconds and downsampled from 100Hz data. Compared to GENEActiv whose window size is 500 time-steps or 5 seconds, for PAMAP2, a small transformation can affect the result very prominently. Therefore, lower values are applied to PAMAP2. The parameters for these augmentation methods and the sensor data for PAMAP2 to be transformed are randomly chosen. These conditions for applying augmentation methods are used in the following experiments as well.

Analyzing augmentation methods on training from scratch and KD

The accuracy of training scratch with different augmentation methods on WRN16-1 is presented in Table 2.7. Most of the accuracies from augmentation methods, except adding noise which can alter peaky points and change gradients, are higher than the accuracy obtained by learning with the original data. Compared to other methods, adding noise may influence classification between similar activities such as walking, which is included in both datasets as detailed sub-categories.

Table 2.7: Accuracy (%) of Training from Scratch on WRN16-1 with Different Augmentation Methods

Method	Dataset	
	GENEActiv	PAMAP2
Original	68.60±0.23	82.81±2.51
Removal	69.20±0.32	83.34±2.41
Noise	67.60±0.36	82.80±2.66
Shift	68.69±0.22	83.91±2.18
Mix1(R+S)	69.31±0.96	83.59±2.37
Mix2(R+N+S)	67.89±0.11	83.64±2.76

The validation accuracy of scratch and Full KD learning on GENEActiv dataset is presented in Figure 2.6. Training from scratch with the original data shows higher accuracy than KD with original data in very early stages before 25 epochs. However, KD shows better accuracy than the models trained from scratch after 40 epochs. KD with augmentation tends to perform better in accuracy than models trained from scratch and KD learning with the original data alone. That is, data augmentation can help to boost the generalization ability of student models for KD. Mix1 shows the highest accuracy among the results. The highest accuracies are seen in early stages, which are less than 120 epochs for all methods, where 120 epochs is less than three-fourths of the total number of epochs. On closer inspection, I find that the best accuracies are actually seen in less than 20 epochs for training from scratch and Full KD, less than 60 epochs for shifting, Mix1, and Mix2, and less than 120 epochs for adding noise, respectively. This implies that not only early stopped teachers but also early stopped students are able to perform better than fully iterated models.

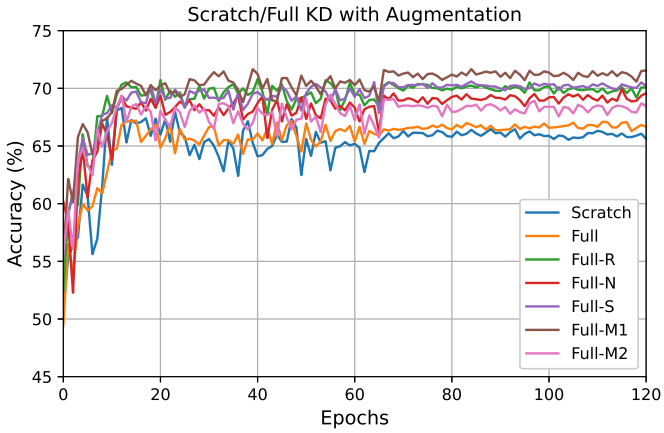


Figure 2.6: The Validation Accuracy for Training from Scratch and Full KD. WRN16-1 Is Used for Training from Scratch. For Full KD, WRN16-3 Is a Teacher Network and WRN16-1 Is a Student Network. R, N, S, M1, and M2 in the Legend Are Removal, Adding Noise, Shifting, Mix1, and Mix2, Respectively.

In training based on KD with augmentation methods, the accuracy goes up in early stages, however, the accuracy suffers towards to the end of training. These trends on KD are similar to the previous ESKD study (Cho and Hariharan (2019)). For the following experiments, I restrict the analyses to ESKD.

Analyzing Augmentation Methods on Distillation

The accuracy of each augmentation method with KD is summarized in Table 2.8 and 2.9 for GENEActiv and Table 2.10 and 2.11 for PAMAP2. The results were obtained from small-sized students of ESKD. The gray colored cells of these tables are the best accuracy for the augmentation method among the different capacity teachers of KD. When a higher λ is used, distillation from teachers is improved, and the best results are obtained when the teacher capacity is smaller. Also, the best performance of students, when learning with augmentation methods and the original

Table 2.8: Accuracy (%) of KD from Variants of Teacher Capacity and Augmentation Methods on GENEActiv ($\lambda = 0.7$)

Method	Teacher		
	Small	Medium	Large
	68.87	69.99	70.19
Original	69.71±0.31	69.61±0.17	68.62±0.33
Removal	69.80±0.34	70.23±0.41	70.28±0.68
Noise	69.26±0.08	69.12±0.19	69.38±0.39
Shift	70.63±0.19	70.43±0.89	70.00±0.20
Mix1(R+S)	70.56±0.57	**71.35**±0.20	70.22±0.10
Mix2(R+N+S)	69.27±0.31	69.51±0.28	69.62±0.21

Table 2.9: Accuracy (%) of KD from Variants of Teacher Capacity and Augmentation Methods on GENEActiv ($\lambda = 0.99$)

Method	Teacher		
	Small	Medium	Large
	68.87	69.99	70.19
Original	69.44±0.19	67.80±0.36	68.67±0.20
Removal	69.48±0.22	69.75±0.40	70.01±0.27
Noise	69.99±0.14	70.20±0.06	70.12±0.14
Shift	**70.96**±0.10	70.42±0.06	70.16±0.24
Mix1(R+S)	70.40±0.27	70.07±0.38	69.36±0.16
Mix2(R+N+S)	70.56±0.23	69.88±0.16	69.71±0.30

Table 2.10: Accuracy (%) of KD from Variants of Teacher Capacity and Augmentation Methods on PAMAP2 ($\lambda = 0.7$)

Method	Teacher		
	Small	Medium	Large
	85.42	85.67	85.17
Original	84.75±2.64	84.47±2.32	84.90±2.38
Removal	85.16±2.46	85.51±2.27	85.02±2.47
Noise	84.96±2.59	85.52±2.26	84.85±2.43
Shift	85.21±2.21	85.45±2.19	**85.66**±2.26
Mix1(R+S)	85.54±2.51	85.60±2.19	84.71±2.53
Mix2(R+N+S)	85.17±2.39	85.27±2.33	83.76±2.77

Table 2.11: Accuracy (%) of KD from Variants of Teacher Capacity and Augmentation Methods on PAMAP2 ($\lambda = 0.99$)

Method	Teacher		
	Small	Medium	Large
	85.42	85.67	85.17
Original	86.37±2.35	86.38±2.25	85.11±2.46
Removal	84.66±2.67	85.70±2.40	84.81±2.52
Noise	84.77±2.65	85.21±2.41	85.05±2.40
Shift	86.08±2.42	**86.65**±2.13	85.53±2.28
Mix1(R+S)	84.93±2.71	85.88±2.28	84.73±2.54
Mix2(R+N+S)	82.94±2.76	83.94±2.70	83.28±2.50

data, is achieved with similar teacher capacities. For example, for GENEActiv with $\lambda = 0.7$, the best results are generated from various capacity of teachers. But, with $\lambda = 0.99$, the best results tend to be seen with smaller capacity of teachers. Even though the evaluation protocol for PAMAP2 is leave-one-subject-out with an imbalanced distribution of data, with $\lambda = 0.7$, the best results are obtained from larger capacity of teachers as well. Furthermore, results from both datasets verify that larger and more accurate teachers do not always result in better students. Also, the best result from shifting is seen at the same capacity of the teacher with the original data. It might be because shifting includes the same time-series 'shapes' as the original data. The method for shifting is simple but is an effectively helpful method for training KD. For all teachers on PAMAP2 with $\lambda = 0.99$, the accuracies from training by shifting are even higher than other combinations. Compared to previous methods (Jordao *et al.* (2018)) with PAMAP2, the result by shifting outperforms others. Furthermore, although the student network of KD has the same number of parameters of the network trained from scratch (WRN16-1), the accuracy is much higher than the latter one; the result of Mix1 from GENEActiv and shifting from PAMAP2 by the medium teacher is approximately 2.7% points and 3.8% points better than the result from original data by training from scratch, respectively. These accuracies are even better than the results of their teachers. It also verifies that KD with an augmentation method including shifting has benefits to obtain improved results.

To investigate the difference in performance with a model trained from scratch and KD with augmentation methods, statistical analysis was conducted by calculating p-value from a t-test with a confidence level of 95%. Table 2.12 and 2.13 show averaged accuracy, standard deviation, and calculated p-value for WRN16-1 trained from scratch with original training set and various student models of WRN16-1 trained with KD and augmentation. That is, student models in KD have the same structure

Table 2.12: p-value and (Accuracy (%), Standard Deviation) for Training from Scratch and KD on GENEActiv Dataset

Scratch	KD (Teacher: Medium)	p-value
Original (68.60±0.23)	Original (ESKD) (69.61±0.17)	0.030
	Original (Full) (68.62±0.22)	0.045
	Removal (70.23±0.41)	0.006
	Noise (69.12±0.19)	0.012
	Shift (70.43±0.89)	0.025
	Mix1(R+S) (71.35±0.20)	0.073
	Mix2(R+N+S) (69.51±0.28)	0.055

of the model trained from scratch and teachers for KD are WRN16-3 ($\tau = 4$, $\lambda = 0.7$). For GENEActiv, in five out of the seven cases, the calculated p-values are less than 0.05. Thus, the results in the table show statistically-significant difference between training from scratch and KD. For PAMAP2, in all cases, p-values are less than 0.05. This also represents statistically-significant difference between training from scratch and KD. Therefore, I can conclude that KD training with augmentation methods,

Table 2.13: p-value and (Accuracy (%), Standard Deviation) for Training from Scratch and KD on PAMAP2 Dataset

Scratch	KD (Teacher: Medium)	p-value
Original (82.81±2.51)	Original (ESKD) (84.47±2.32)	0.0298
	Original (Full) (84.31±2.24)	0.0007
	Removal (85.51±2.27)	0.0008
	Noise (85.52±2.26)	0.0002
	Shift (85.45±2.19)	0.0034
	Mix1(R+S) (85.60±2.19)	0.0024
	Mix2(R+N+S) (85.27±2.33)	0.0013

which shows better results in classification accuracy, performs significantly different from training from scratch, at a confidence level of 95%.

Finally, the expected calibration error (ECE) (Guo *et al.* (2017)) is calculated to measure the confidence of performance for models trained from scratch and KD ($\tau = 4$, $\lambda = 0.7$) with augmentation methods. As shown in Table 2.14 and 2.15, in all cases, ECE values for KD are lower than when models are trained from scratch, indicating

Table 2.14: ECE (%) of Training from Scratch and KD on GENEActiv Dataset

Scratch	ECE	KD (Teacher: Medium)	ECE
Original	3.22	Original (ESKD)	2.96
Removal	3.56	Removal	2.90
Noise	3.45	Noise	2.85
Shift	3.24	Shift	2.78
Mix1(R+S)	3.72	Mix1(R+S)	2.79
Mix2(R+N+S)	3.67	Mix2(R+N+S)	2.86

Table 2.15: ECE (%) of Training from Scratch and KD on PAMAP2 dataset

Scratch	ECE	KD (Teacher: Medium)	ECE
Original	2.28	Original (ESKD)	2.16
Removal	3.64	Removal	3.09
Noise	5.83	Noise	3.01
Shift	2.87	Shift	2.22
Mix1(R+S)	4.39	Mix1(R+S)	2.96
Mix2(R+N+S)	5.55	Mix2(R+N+S)	4.17

that models trained with KD have higher reliability. Also, results of KD including shifting are lower than results from other augmentation methods. This additionally verifies that KD improves the performance and shifting helps to get improved models.

Analyzing training for KD with augmentation methods

The loss values of each method, for the medium-sized teacher, are shown in Table 2.16 and 2.17. The loss values were obtained from the final epoch while training student models based on Full KD. As shown in these tables, for both cross entropy and KD loss values, training with shifting-based data augmentation results in lower loss, compared to other augmentation strategies and the original model. The loss value for noise augmentation is higher than the values of shifting. On the other hand, the KD loss value for Mix1 is higher than the values for removal and shifting. However, the training loss is for these two methods and its value of testing is lower. Compared to other methods, Mix2 shows higher loss for training, which may be because this method generates more complicated patterns. However, the testing KD loss value of Mix2 is lower than the value of original and adding noise. These findings imply that the data of original and shifting have very similar patterns. And data based on Mix1 and Mix2 are not simply trainable data for distillation, however, these methods have an effect of preventing a student from over-fitting or degradation for classification. The contrast of results from GENEActiv between each method is more prominent than the one from PAMAP2. This is due to the fact that smaller parameters for augmentation are applied to PAMAP2. Also, the dataset is more challenging to train on, due to imbalanced data and different channels in sensor data.

2.4.4 Analysis of Teacher and Student Models with a Variant Properties of Training Set

To discuss properties of training set for teacher and student models, I use the same parameter ($\tau = 4$, $\lambda = 0.7$) in this experiment on two datasets. In this section, I try to train a medium teacher and a small student by training set having the same or

Table 2.16: The Loss Value ($\mathbf{10^{-2}}$) for KD (Teacher: Medium) from Various Methods on GENEActiv

Method (λ=0.7)	CE Train	KD Train	KD Test
Original	3.774	0.617	1.478
Removal	3.340	0.406	1.246
Noise	11.687	**1.172**	1.358
Shift	2.416	**0.437**	1.119
Mix1(R+S)	5.475	0.475	**1.108**
Mix2(R+N+S)	17.420	1.337	1.338

Table 2.17: The Loss Value ($\mathbf{10^{-2}}$) for KD (Teacher: Medium) from Various Methods on PAMAP2 (Subject 101)

Method (λ=0.7)	CE Train	KD Train	KD Test
Original	0.832	0.156	1.783
Removal	1.237	0.146	1.038
Noise	1.066	**0.138**	1.284
Shift	0.468	**0.129**	1.962
Mix1(R+S)	1.267	0.150	**0.895**
Mix2(R+N+S)	1.853	0.177	1.065

different properties to take into account relationships between teachers and students. Testing set is not transformed or modified. The medium teacher is chosen because

the teacher showed good performance in the prior experiments discussed in previous sections. Further, distillation from a medium model to a small model is an preferable approach (Cho and Hariharan (2019)). Also, I analyze which augmentation method is effective to achieve higher accuracy. I use adding noise, shifting, and Mix1 methods which transform data differently.

Table 2.18: Accuracy (%) of Training from Scratch on WRN16-3 with Different Augmentation Methods

Dataset	Original	Noise	Shift	Mix1(R+S)
GENEActiv	69.53±0.40	68.59±0.05	72.08±0.20	71.64±0.26
GENEActiv (Top-1)	69.99	68.68	72.48	72.17
PAMAP2	84.65±2.28	83.08±2.51	82.54±2.42	82.39±2.62
PAMAP2 (Top-1)	85.67	85.31	84.38	84.09

To obtain a medium teacher model, the model is trained from scratch with augmentation methods. These results are shown in Table 2.18. For GENEActiv, shifting based data augmentation gives the best performance. However, for PAMAP2, original data achieves the best performance. Mix1 shows slightly lower accuracy than shifting. In these experiments, the student model is trained using the teacher model that achieves best performance over several trials.

I also evaluated different combinations of data augmentation strategies for teacher-student network pairs. A pair is obtained by using one or no data augmentation strategy to train the teacher network by training from scratch, and the student network is trained by ESKD under different, same, or no augmentation strategy. The

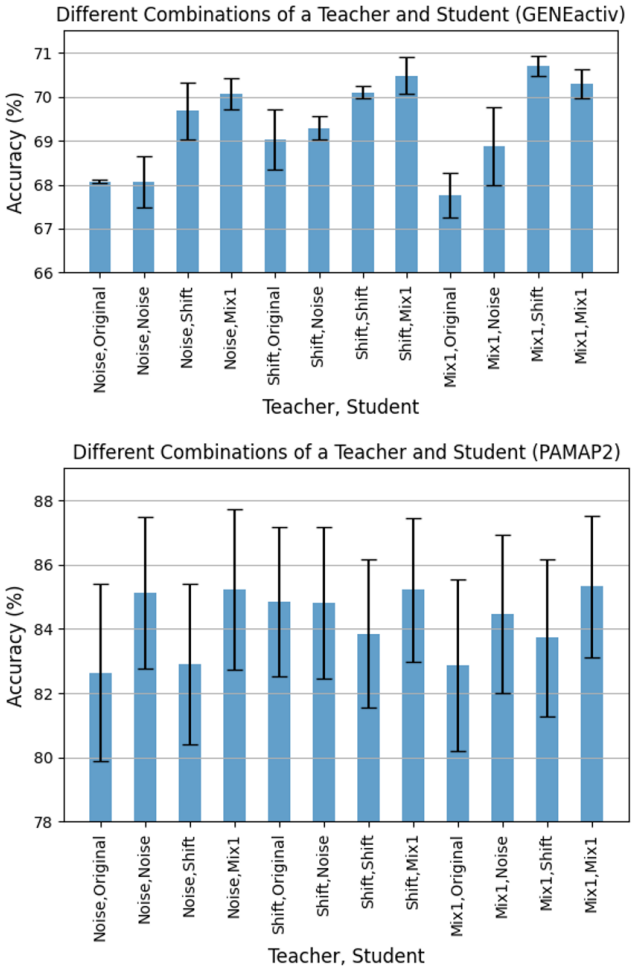


Figure 2.7: The Results for Students Trained by Different Combinations of Training Sets for Teachers and Students. The Teacher and Student Both Are Learned by Augmentation Methods. WRN16-3 (Medium) and WRN16-1 (Small) Are Teacher and Student Networks, Respectively.

results are shown in Figure 2.7. I found that KD with the same data augmentation strategy for training teachers and students may not be the right choice to get the best performance. When a teacher is trained by shifting and a student is trained by Mix1 which showed good performance as a student in the previous sections, the results are better than other combinations for both datasets. Also, when a student is learned by

Mix1 including shifting transform, in general, the performance are also good for all teachers. It implies that the method chosen for training a student is more important than choosing a teacher; KD with a medium teacher trained by the original data and a student trained with shift or Mix1 outperforms other combinations. Using the same strategy for training data for teachers and students does not always present the best performance. When the training set for students is more complicated than the set for teachers, the performance in accuracy tends to be better. That is, applying a transformation method to students can help to increase the accuracy. It also verifies that better teachers do not always lead to increased accuracy of students. Even if the accuracies from these combinations of a teacher and student are lower than models trained from scratch by WRN16-3, the number of parameters for the student is only about 11% of the one for WRN16-3. Therefore, the results still are good when considering both performance and computation.

2.4.5 Analysis of Student Models with Different Data Augmentation Strategies for Training and Testing Set

In this section, I study the effect of students on KD from various augmentation methods for training and testing, while a teacher is trained with the original dataset. I use the same parameter ($\tau = 4$, $\lambda = 0.7$) and ESKD for this experiment on two datasets. A teacher is selected with a medium model trained by the original data. I use adding noise, shifting and Mix1 methods which transform data differently.

After training the teacher network on original data, a student network is trained with different data augmentation strategies and is evaluated on test data transformed with different data augmentation strategies. The results are illustrated in Figure 2.8. For GENEActiv, most often, training student networks with Mix1 show better performance on different testing sets. However, if the testing set is affected by adding

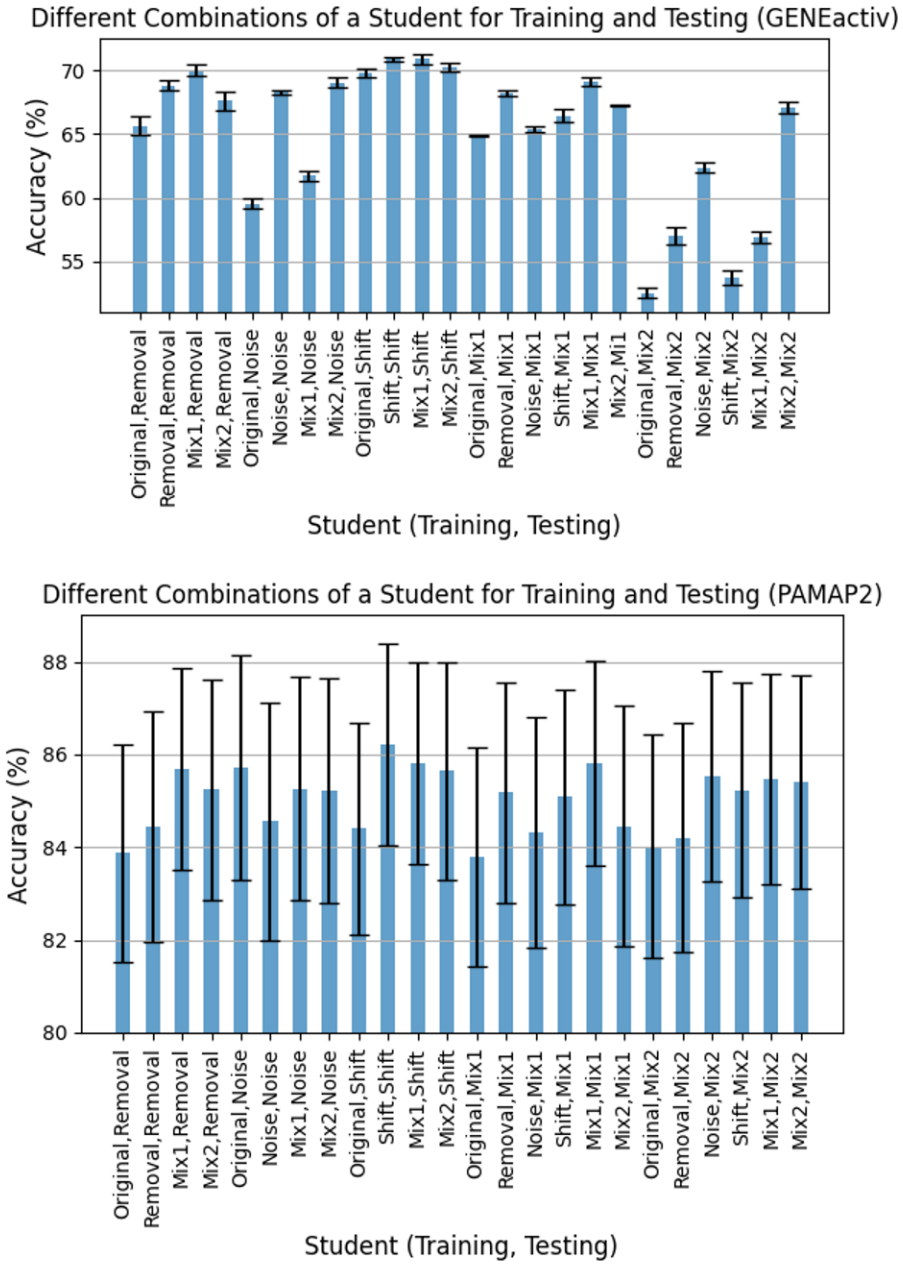


Figure 2.8: Effect on Classification Performance of Student Network with Different Augmentation Methods for Training and Testing Sets. WRN16-3 (Medium) and WRN16-1 (Small) Are Teacher and Student Networks, Respectively.

noise, training students with adding noise and Mix2 shows much better performance than training with shifting and Mix1. From the results on PAMAP2, in most of the cases, training students with Mix1 shows better performance to many different testing set. However, when the testing set is augmented by adding noise, training with original data shows the best performance. This is likely attributable to the window size, which has about a hundred samples, and the dataset includes the information of 4 kinds of IMUs. Therefore, injecting noise, which can affect peaky points and change gradients, creates difficulties for classification. Also, these issue can affect the both training and testing data. Thus, if the target data includes noise, training set and augmentation methods have to be considered along with the length of the window and intricate signal shapes within the windows.

2.4.6 Analysis of Testing Time

Here, I compare the evaluation time for various models on the GENEActiv dataset. I conducted the test on a desktop with a 3.50 GHz CPU (Intel® Xeon(R) CPU E5-1650 v3), 48 GB memory, and NVIDIA TITAN Xp (3840 NVIDIA® CUDA® cores and 12 GB memory) graphic card. I used a batch size of 1 and approximately 6000 data samples for testing. Four different models were trained from scratch with WRN16-k (k=1, 3, 6, and 8). To test with ESKD and Mix1, WRN16-3 was used as a teacher and WRN16-1 was used for student network. As expected, larger models take more time for testing, as shown in Table 2.19. WRN16-1 as a student trained by ESKD with Mix1 augmentation achieves the best accuracy, 71.35%, where the model takes the least amount of time on both GPU and CPU. The results on CPU reiterate the reason why model compression is required for many applications, especially on edge devices, wearables, and mobile devices, which have limited computational and

power resources and are generally implemented in real time with only CPU. The gap in performance would be higher if an edge device had lower computational resources.

Table 2.19: Processing Time of Various Models for GENEActiv Dataset

Model (WRN16-k)	Acc. (%)	Total GPU (sec)	Avg. GPU (ms)	Total CPU (sec)	Avg. CPU (ms)
k=1	67.66				
k=1 (ESKD)	69.61	15.226	2.6644	**16.655**	2.8920
k=1 (ESKD+Mix1)	**71.35**				
k=3	68.89	16.426	2.8524	21.333	3.7044
k=6	70.04	16.663	2.8934	33.409	5.8012
k=8	69.02	16.885	2.9320	46.030	7.9928

2.5 Conclusion

In this chapter, I studied many relevant aspects of knowledge distillation (KD) for wearable sensor data as applied to human activity analysis. I conducted experiments with different sizes of teacher networks to evaluate their effect on KD performance. I show that a high capacity teacher network does not necessarily ensure better performance of a student network. I further showed that training with augmentation methods and early stopping for KD (ESKD) is effective when dealing with time-series data. I also establish that the choice of augmentation strategies has more of an impact on the student network training as opposed to the teacher network. In most cases, KD training with the Mix1 (Removal+Shifting) data augmentation strategy for students showed robust performance. Further, I also conclude that a single augmen-

tation strategy is not conclusively better all the time. Therefore, I recommend using a combination of augmentation methods for training KD in general. In summary, these findings provide a comprehensive understanding of KD and data augmentation strategies for time-series data from wearable devices of human activity. These conclusions can be used as a general set of recommendations to establish a strong baseline performance on new datasets and new applications.

Chapter 3

LEVERAGING ANGULAR DISTRIBUTIONS FOR IMPROVED KNOWLEDGE DISTILLATION

3.1 Introduction

In the past decade, convolutional neural networks (CNN) have been widely deployed into many commercial applications. Various architectures that go beyond convolutional methods have also been developed. However, a core challenge in all of them is that they are accompanied by high computational complexity, and large storage requirements (Gou *et al.* (2021); Cho and Hariharan (2019)). For this reason, application of deep networks is still limited to environments that have massive computational support. In emerging applications, there is growing demand for applying deep nets on edge, mobile, and IoT devices (Li *et al.* (2018); Plastiras *et al.* (2018); Jang *et al.* (2020); Wu *et al.* (2016)). To move beyond these limitations, many studies have developed a lightweight form of neural models which assure performance while 'lightening' the network scale (Cho and Hariharan (2019); Li *et al.* (2018); Plastiras *et al.* (2018); Jang *et al.* (2020); Wu *et al.* (2016); Han *et al.* (2016); Hinton *et al.* (2015)).

Knowledge distillation (KD) is one of the promising solutions that can reduce the network size and develop an efficient network model (Gou *et al.* (2021); Cho and Hariharan (2019); Yim *et al.* (2017)) for various fields including wearable sensor data (Jeon *et al.* (2022b)), sound (Tripathi and Paul (2022); Li *et al.* (2021b)), and image classification (Wen *et al.* (2021b); Chen *et al.* (2021)). The concept of knowledge distillation is that the network consists of two networks, a larger one called teacher

and a smaller one called student (Hinton *et al.* (2015)). During training the student, the teacher transfers its knowledge to the student, using the logits from the final layer. So, the student can retain the teacher model's classification performance.

Recent insights have shown that features learnt in deep-networks often exhibit an angular distribution, usually leveraged via a hyperspherical embedding (Choi *et al.* (2020); Liu *et al.* (2016, 2017)). Such embeddings lead to improved discriminative power, and feature separability. In terms of loss-functions, these can be implemented by using angular features that correspond to the geodesic distance on the hypersphere and incorporating a preset constant margin. In this work, I show that leveraging such spherical embeddings also improves knowledge distillation. Firstly, to get more activated features, spatial attention maps are computed and decoupled into two parts: positive and negative maps. Secondly, I construct a new form of knowledge by projecting the features onto the hypersphere to reflect the angular distance between them. Then, I introduce an angular margin to the positive feature to get a more attentive feature representation. Finally, during the distillation, the student tries to mimic the more separated decision regions of the teacher to improve the classification performance. Therefore, the proposed method effectively regularizes the feature representation of the student network to learn informative knowledge of the teacher network.

The contributions of this chapter are:

- I propose an angular margin based distillation loss (named as AMD) which performs knowledge distillation by transferring the angular distribution of attentive features from the teacher network to the student network.

- I experimentally show that the proposed method results in significant improvements with different combinations of networks and outperforms other attention-

based methods across four datasets of different complexities, corroborating that the performance of a higher capacity teacher model is not necessarily better.

- I rigorously validate the advantages of the proposed distillation method with various aspects using visualization of activation maps, classification accuracy, and reliability diagrams.

The rest of the chapter is organized as follows. In section 3.2 and 3.3, I describe related work and background, respectively. In section 3.4, I provide an overview of the proposed method. In section 3.5, I describe the experimental results and analysis. In section 3.6, I discuss the findings and conclusions.

3.2 Related Work

Knowledge Distillation. Knowledge distillation, a transfer learning method, trains a smaller model by shifting knowledge from a larger model. KD is firstly introduced by Buciluǎ *et al.* (Buciluǎ *et al.* (2006)) and is further explored by Hinton *et al.* (Hinton *et al.* (2015)). The main concept of KD is using soft labels by a trained teacher network. That is, mimicking soft probabilities helps students get knowledge of teachers, which improves beyond using hard labels (training labels) alone. Cho *et al.* (Cho and Hariharan (2019)) explore which combination of student-teacher is good to obtain the better performance. They show that using a teacher trained by early stopping the training improves the efficacy of KD. KD can be categorized into two approaches that use the outputs of the teacher (Gou *et al.* (2021)). One is response-based KD, which uses the posterior probabilities with softmax loss. The other is feature-based KD using the intermediate features with normalization. Feature-based methods can be performed with the response-based method to complement traditional KD (Gou *et al.* (2021)). Recently, feature-based distillation methods for KD

have been studied to learn richer information from the teacher for better-mimicking and performance improvement (Gou *et al.* (2021); Wen *et al.* (2021b); Wang and Yoon (2021)). Romero *et al.* (Romero *et al.* (2015)) firstly introduced the use of intermediate representations in FitNets using feature-based distillation. This method enables the student to mimic the teacher's feature maps in intermediate layers.

Attention Transfer. To capture the better knowledge of a teacher network, attention transfer (Gou *et al.* (2021); Zagoruyko and Kmodakis (2017); Wang *et al.* (2020c); Ji *et al.* (2021)) has been utilized, which is one of the popular methods for feature-based distillation. Zagoruyko *et al.* (Zagoruyko and Kmodakis (2017)) suggest activation-based attention transfer (AT), which uses a sum of squared attention mapping function computing statistics across the channel dimension. Although the depth of teacher and student is different, knowledge can be transferred by the attention mapping function, which matches the depth size as one. The activation-based spatial attention maps are used as the source of knowledge for distillation with intermediate layers, where the maps are created as: $f^d_{sum}(A) = \sum_{j=1}^{c} |A_j|^d$, where f is a computed attention map, A is an output of a layer, c is the number of channels for the output, j is the number for the channel, and $d > 1$. A higher value of d corresponds to a heavier weight on the most discriminative parts defined by activation level. AT (feature-based distillation method) shows better effectiveness when used with traditional KD (response-based KD) (Zagoruyko and Kmodakis (2017)). The method encourages the student to generate similar normalized maps as the teacher. However, these studies have only focused on mimicking the teacher's activation from a layer (Wang and Yoon (2021)), not considering the teacher's dual ability to accurately distinguish between positive (relevant to the target object) and negative (irrelevant). Teacher not only can generate and transfer its knowledge as an activation map directly, but also can transfer separability to distinguish between positive and negative

features. I refer to this as a dual ability, which I consider for improved distillation. The emphasized positive feature regions that encapsulate regions of the target object are crucial to predicting the correct class. In general, a higher-capacity model shows better performance, producing those regions with more attention and precision compared to the smaller network. This suggests that the transfer of distinct regions of the positive and negative pairs from teacher to student could significantly improve performance. This motivates us to focus on utilizing positive and negative pairs for extracting more attentive features, implying better separability, for distillation.

Spherical Feature Embeddings. The majority of existing methods (Sun *et al.* (2014); Wen *et al.* (2016)) rely on Euclidean distance for feature distinction. These approaches could not solve the problem that classification under open-set protocol shows a meaningful result only when successfully narrowing maximal intra-class distance. To solve this problem, an angular-softmax (A-softmax) function is proposed to distinguish the features by increasing the angular margins between features (Liu *et al.* (2017)). According to its geometric interpretation, using A-softmax function equivalents to the projection of features onto the hypersphere manifold, which intrinsically matches the preliminary condition that features also lie on a manifold. Applying the angular margin penalty corresponds to the geodesic distance margin penalty in the hypersphere (Liu *et al.* (2017)). A-softmax function encourages learned features to be discriminative on hypersphere manifold. For this reason, the A-softmax function shows superior performance to the original softmax function when tested on several classification problems (Liu *et al.* (2017)). On the other hand, Choi *et al.* (Choi *et al.* (2020)) introduced angular margin based contrastive loss (AMC-loss) as an auxiliary loss, employing the discriminative angular distance metric that corresponds to geodesic distance on a hypersphere manifold. AMC-loss increases inter-class separability and intra-class compactness, improving performance in classification. The

method can be combined with other deep techniques, because it easily encodes the angular distributions obtained from many types of deep feature learners (Choi *et al.* (2020)).

The previous methods work with logits only or work with an auxiliary loss, such as a contrastive loss. I focus on features modeled as coming from angular distributions, and focus on their separability. The observations give us an insight that the high quality features for knowledge distillation can be obtained by projecting the feature pairs onto a hypersphere. For better distillation, I construct a derive new type of implicit knowledge with positive and negative pairs from intermediate layers. The details are explained in section 3.4.

3.3 Background

3.3.1 Traditional Knowledge Distillation

In standard knowledge distillation (Hinton *et al.* (2015)), the loss for training a student is:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\mathcal{C}} + \lambda\mathcal{L}_{\mathcal{K}}, \tag{3.1}$$

where, $\mathcal{L}_{\mathcal{C}}$ denotes the standard cross entropy loss, $\mathcal{L}_{\mathcal{K}}$ is KD loss, and λ is a hyperparameter; $0 < \lambda < 1$. The error between the output of the softmax layer of a student network and the ground-truth label is penalized by the cross-entropy loss:

$$\mathcal{L}_{\mathcal{C}} = \mathcal{H}(softmax(a_S), y), \tag{3.2}$$

where $\mathcal{H}(\cdot)$ is a cross entropy loss function, a_S is the logits of a student (inputs to the final softmax), and y is a ground truth label. The outputs of student and teacher are matched by KL-divergence loss:

$$\mathcal{L}_{\mathcal{K}} = \tau^2 KL(z_T, z_S), \tag{3.3}$$

where, $z_T = softmax(a_T/\tau)$ is a softened output of a teacher network, $z_S = softmax(a_S/\tau)$ is a softened output of a student, and τ is a hyperparameter; $\tau > 1$. Feature distillation methods using intermediate layers can be used with the standard knowledge distillation that uses output logits. When they are used together, in general, it is beneficial to guide the student network towards inducing more similar patterns of teachers and getting a better classification performance. Thus, I also utilize the standard knowledge distillation with the proposed method.

3.3.2 Attention Map

Denote an output as $A \in \mathbb{R}^{c \times h \times w}$, where c is the number of output channels, h is the height for the size of output, and w is width for the size of the output. The attention map for the teacher is given as follows:

$$f_T^l = \sum_{j=1}^{c} |A_{T,j}^l|^2. \tag{3.4}$$

Here, A_T is an output of a layer from a teacher model, l is a specific layer, c is the number of channels, j is the number for the output channel, and T denotes a teacher network. The attention map for the student is $f_S^{l'} = \sum_{j'=1}^{c'} |A_{S,j'}^{l'}|^2$, where $A_S^{l'}$ is an output of a layer from a student, l' is the corresponding layer of l, c' is the number of channels for the output, j' is the number for the output channel, and S denotes a student network. If the student and teacher use the same depth for transfer, l' can be the layer at the same depth as l; if not, l' can be the end of the same block for the teacher. From the attention map, I obtain positive and negative maps and I project features onto hypersphere to calculate angular distance for distillation. The details are explained in section 3.4.

3.3.3 *Spherical Feature with Angular Margin*

In order to promote the learned features to have an angular distribution, (Liu *et al.* (2017); Wang *et al.* (2018a)) proposed to introduce the angular distance between features W and weights x. For example, $W^T x = \|W\| \, \|x\| \, cos(\theta)$, where bias is set as 0 for simplicity, and θ is the angle between W and x. Then, the normalization of feature and weight makes the outputs only depend on the angle between weights and features and further, $\|x\|$ is replaced to a constant s such that the features are distributed on a hypersphere with a radius of s. To enhance the discrimination power, angular margin m is applied to the angle of the target. Finally, output logits are used to formulate probability with angular margin m as below (Liu *et al.* (2017); Wang *et al.* (2018a)):

$$G^i = log\left(\frac{e^{s\cdot(cos(m\cdot\theta_{y_i}))}}{e^{s\cdot(cos(m\cdot\theta_{y_i}))} + \sum_{j=1,j\neq y_i}^{J} e^{s\cdot(cos(\theta_j))}}\right), \tag{3.5}$$

where, y_i is a label and θ_{y_i} is a target angle for class i, θ_j is an angle obtained from j-th element of output logits, s is a constant, and J is the class number. Liu *et al.* (Liu *et al.* (2017)) and Wang *et al.* (Wang *et al.* (2018a)) utilized output logits to obtain more discriminative features for classification on a hypersphere manifold, which performs better than using original softmax function. I use Equation (3.5) to create the new type of feature-knowledge in the intermediate layers instead of output logits in the final classifier, thereby more attentive feature maps are transferred to the student model.

3.4 Proposed Method

The proposed method utilizes features from intermediate layers of deep networks for extracting angular-margin based knowledge as illustrated in Figure 3.1. The resultant angular margin loss is computed at various depths of the student and teacher

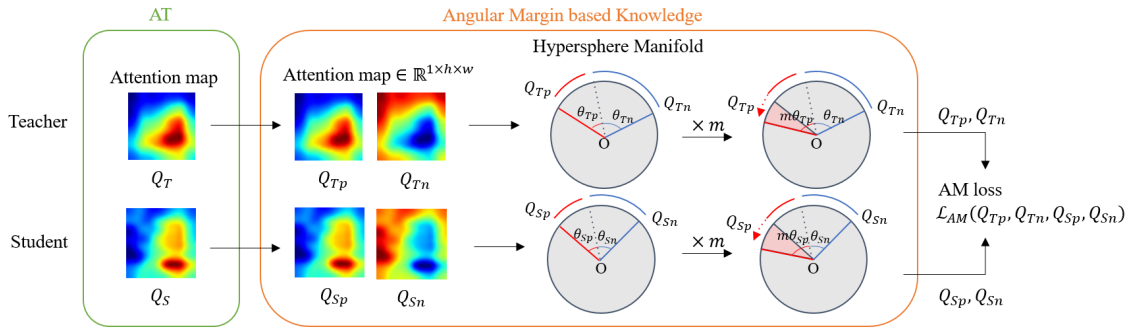


Figure 3.1: The Existing Attention Map-based Method (AT (Zagoruyko and Kmodakis (2017))) Suggested the Direct Use of the Feature Map in the Intermediate Layer as Shown in the Green Box. Instead, I First Decouple the Feature Map into the Positive (q_p) and Negative (q_n) Features and Map Them on the Hypersphere with Angular Margin, m. Then, I Convert Them into the Probability Forms and Compute Loss Based on AM Loss Function. The Details Are Explained in Section 3.4.2.

as illustrated in Figure 3.2. To obtain the angular distance between positive and negative features, I first generate attention maps from the outputs of intermediate layers. I then decouple the maps into positive and negative features. The features are projected onto a hypersphere to extract angularly distributed features. For effective distillation, more attentive features are obtained by introducing angular margin to the positive feature and the probability forms for distillation are computed. Finally, the knowledge of the teacher having better discrimination of positive and negative features is transferred to the student. The details for obtaining the positive and negative maps and the angular margin based knowledge are explained in the following section.

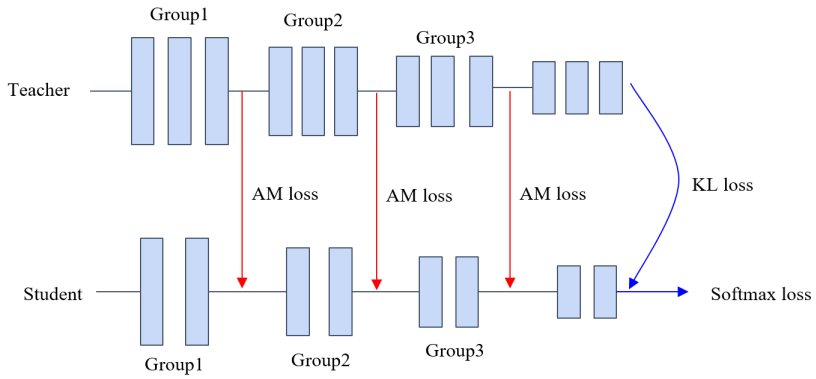


Figure 3.2: Schematics of Teacher-student Knowledge Transfer with the Proposed Method.

3.4.1 Generating Attention Maps

To transfer activated features from teacher to student, the output of intermediate layers are used. To match the dimension size between teacher and student models, I create the normalized attention maps (Zagoruyko and Kmodakis (2017)), which has benefits in generating maps discriminatively between positive and negative features. This reduces the need for any additional training procedure for matching the channel dimension sizes between teacher and student. I use the power value $d = 2$ for generating the attention maps, which shows the best results as reported in previous methods (Zagoruyko and Kmodakis (2017)).

3.4.2 Angular Margin Computation

Although the activation map-based distillation provides additional context information for student model learning, there is still room to craft an attentive activation map that can distill a superior student model in KD. To further refine the original attention map, I propose an angular margin-based distillation (AMD) that encodes

new knowledge using the angular distance between positive (relevant to the target object) and negative features (irrelevant) on the hypersphere.

I denote the normalized positive map as $Q_p = f/\|f\|$ where f is the output map extracted from the intermediate layer in networks. Further, I can obtain the normalized negative map by $Q_n = 1 - Q_p$.

Then, to make the positive map more attentive, I insert an angular margin m into the positive features. In this way, a new feature-knowledge encoding attentive feature can be defined as follows:

$$G^l(Q_p, Q_n) = log\left(\frac{e^{s\cdot(cos(m\cdot\theta_{p_l}))}}{e^{s\cdot(cos(m\cdot\theta_{p_l}))} + e^{s\cdot(cos(\theta_{n_l}))}}\right), \tag{3.6}$$

where, $\theta_{p_l} = cos^{-1}(Q_p)$ and $\theta_{n_l} = cos^{-1}(Q_n)$ for l^{th} layer in the networks, and m is a scalar angular margin. $G^l \in \mathbb{R}^{1\times h\times w}$ reflects the angular distance between positive and negative features in l^{th} layer. For transferring knowledge, I aim to make the student's $G^l(Q_{Sp}, Q_{Sn})$ approximate the teacher's $G^l(Q_{Tp}, Q_{Tn})$ by minimizing the angular distance between feature maps.

3.4.3 Angular Margin Based Distillation Loss

With redesigned knowledge as above, I finally define the angular margin based distillation loss that accounts for the knowledge gap between the teacher and student activations as:

$$\begin{aligned}\mathcal{L}_{AM}(Q_{Tp}, Q_{Tn}, Q_{Sp}, Q_{Sn}) = \frac{1}{3|L|}\sum_{(l,l')\in L} \\ \Bigg(\underbrace{\left\|\hat{G}^l(Q_{Tp}, Q_{Tn}) - \hat{G}^{l'}(Q_{Sp}, Q_{Sn})\right\|_F^2}_{\mathbf{A}} + \\ \underbrace{\left\|\hat{Q}^l_{Tp} - \hat{Q}^{l'}_{Sp}\right\|_F^2}_{\mathbf{P}} + \underbrace{\left\|\hat{Q}^l_{Tn} - \hat{Q}^{l'}_{Sn}\right\|_F^2}_{\mathbf{N}}\Bigg).\end{aligned} \tag{3.7}$$

Here, $\hat{G}$ denotes a function for normalization for output of function G, $\hat{Q}$ is a normalized map. L collects the layer pairs (l and l'), and $\|\cdot\|_F$ is the Frobenius norm (Tung and Mori (2019)). I will verify the performance of each component (A, P, and N) in section 3.5.3.

The final loss ($\mathcal{L}_{AMD}$) of the proposed method combines all the distillation losses, including the conventional logit distillation (Equation (3.3)). Thus, the overall learning objective can be written as:

$$\mathcal{L}_{AMD} = \lambda_1 \mathcal{L}_{\mathcal{C}} + \lambda_2 \mathcal{L}_{\mathcal{K}} + \gamma \mathcal{L}_{\mathcal{A}}, \tag{3.8}$$

where, $\mathcal{L}_{\mathcal{C}}$ is a cross-entropy loss, $\mathcal{L}_{\mathcal{K}}$ is a knowledge distillation loss, $\mathcal{L}_{\mathcal{A}}$ denotes the angular margin based loss from $\mathcal{L}_{AM}$, and λ_1, λ_2, and γ are hyperparameters to control the balance between different losses.

Global and Local Feature Distillation. So far, I only consider the global feature (i.e., preserving its dimension and size). However, I point out that the global feature sometimes does not transfer more informative knowledge and rich spatial information across contexts of an input. Therefore, I also suggest utilizing local features during distillation. Specifically, the global feature is the original feature without a map division. Local features are determined by the division of the global feature. I split the global feature map from each layer by 2 for the width and height sizes of the maps to create four (2×2) local feature maps. That is, one local map has $h/2 \times w/2$ size, where h and w are the height and width sizes of the global map. Similar to before, local features encoding the attentive angle can be extracted for both teacher and student. Then, the losses considering global and local features for the proposed method are:

$$\begin{aligned} \mathcal{L}_{\mathcal{A}_{\text{global}}} &= \mathcal{L}_{AM}(Q_T, Q_S), \\ \mathcal{L}_{\mathcal{A}_{\text{local}}} &= \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{AM}(Q_T^k, Q_S^k), \end{aligned} \tag{3.9}$$

where Q_T and Q_S are global features of the teacher and student for distillation, and Q_T^k and Q_S^k are local features of the teacher and student, respectively, for k-th element of K, where K is the total number of local maps from a map; $K = 4$. When $\mathcal{L}_{\mathcal{A}_{\text{global}}}$ and $\mathcal{L}_{\mathcal{A}_{\text{local}}}$ are used together, I applied weights of 0.2 for local and 0.8 for global features to make a balance for learning.

3.5 Experiments

In this section, I present experimental validation of the proposed method. I evaluate the proposed method, AMD, with various combinations of teacher and student, which have different architectural styles. I run experiments on four public datasets that have different complexities. I examine the sensitivity with several different hyperparameters (γ and m) for the proposed distillation and discuss which setting is the best. To demonstrate the detailed contribution, I report the results with various aspects, using classification accuracy as well as activation maps extracted by Grad-CAM (Selvaraju *et al.* (2017)). Finally, I investigate performance enhancement by combining previous methods including filtered feature based distillation. Each experiment and its corresponding section are described in Table 3.1.

3.5.1 Datasets

CIFAR-10. CIFAR-10 dataset (Krizhevsky and Hinton (2009)) includes 10 classes with 5000 training images per class and 1000 testing images per class. Each image is an RGB image of size 32×32. I use the 50000 images as the training set and 10000 as the testing set. The experiments on CIFAR-10 helps validate the efficacy of the models with less time consumption.

CINIC-10. I extend the experiments on CINIC-10 (Darlow *et al.* (2018)). CINIC-10 comprises of augmented extension in the style of CIFAR-10, but the dataset con-

Table 3.1: Description of Experiments and Their Corresponding Sections.

Description	Section
1. Does AMD work to distill a better student?	
• Comparison with various attention based distillation methods. • Investigating the effect of each component of the proposed method.	3.5.3
2. What is the effect of learning with AMD from various teachers?	
• Exploring with different capacity of teachers.	3.5.4
3. What is the effect of different hyperparameters?	
• Ablation study with γ and m.	3.5.5
4. What are the visualized results for the area of interest?	
• Visualized results of activation maps from intermediate layers with or without local feature distillation.	3.5.6
5. Is AMD able to perform with existing methods?	
• Evaluation with various methods such as fine-grained feature distillation, augmentation, and other distillation methods. • Generalizability analysis with ECE and reliability diagrams.	3.5.7

tains 270,000 images whose scale is closer to that of ImageNet. The images are equally split into each 'train', 'test', and 'validate' sets. The size of the images is 32×32. There are ten classes with 9000 images per class.

Table 3.2: Architecture of WRN Used in Experiments. Downsampling Is Performed in the First Layers of Conv3 and Conv4. 16 and 28 Mean Depth and k Is Width (Channel Multiplication) of the Network.

Group Name	Output Size	WRN16-k	WRN28-k
conv1	32×32	3×3, 16	3×3, 16
conv2	32×32	$\begin{bmatrix} 3\times3,\ 16k \\ 3\times3,\ 16k \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3,\ 16k \\ 3\times3,\ 16k \end{bmatrix} \times 4$
conv3	16×16	$\begin{bmatrix} 3\times3,\ 32k \\ 3\times3,\ 32k \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3,\ 32k \\ 3\times3,\ 32k \end{bmatrix} \times 4$
conv4	8×8	$\begin{bmatrix} 3\times3,\ 64k \\ 3\times3,\ 64k \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3,\ 64k \\ 3\times3,\ 64k \end{bmatrix} \times 4$
	1×1	average pool, 10-d fc, softmax	

Tiny-ImageNet / ImageNet. To extend the experiments on a larger scale dataset having more complexity, I use Tiny-ImageNet (Le and Yang (2015)). The size of the images for Tiny-ImageNet is 64×64. I pad them to 68×68, then they are randomly cropped to 64×64, and horizontally flipped, for augmentation to account for the complexity of the dataset. The training and testing sets are of size 100k and 10k respectively. The dataset includes 200 classes. For ImageNet (Deng *et al.* (2009)), The dataset has 1k categories with 1.2M training images. The images are randomly cropped and then resized to 224×224 and horizontally flipped.

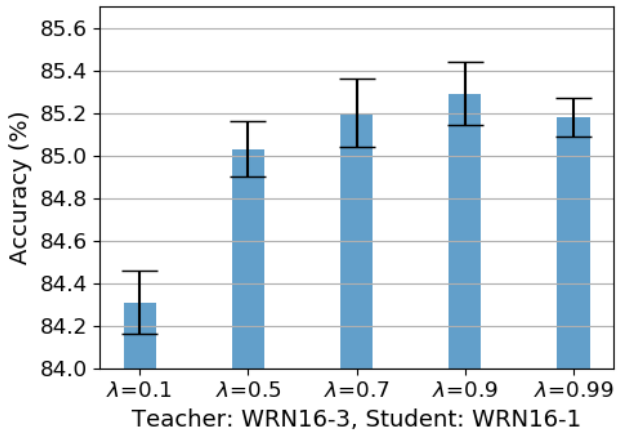


Figure 3.3: Accuracy (%) of Students (WRN16-1) Trained with a Teacher (WRN16-3) on CIFAR-10 for Various λ_2. λ_1 Is Obtained by 1 - λ_2.

3.5.2 Settings for Experiments

For experiments on CIFAR-10, CINIC-10, and Tiny-ImageNet, I set the batch size as 128, the total epochs as 200 using SGD with momentum 0.9, a weight decay of 1×10^{-4}, and the initial learning rate lr as 0.1 which is decayed by a factor of 0.2 at epochs 40, 80, 120, and 160. For ImageNet, I use SGD with momentum of 0.9 and the batch size is set as 256. I run a total epoch of 100. The initial learning rate lr is 0.1 decayed by 0.1 in 30, 60, and 90 epochs.

In experiments, I use the proposed method with WideResNet (WRN) (Zagoruyko and Komodakis (2016)) for teacher and student models to evaluate the classification accuracy, which is popularly used for KD (Cho and Hariharan (2019); Yim *et al.* (2017); Zagoruyko and Kmodakis (2017); Tung and Mori (2019)). Their network architectures are described in Table 3.2.

To determine optimal parameters λ_1 and λ_2 for KD, I tested with different values for λ_1 and λ_2 for training based on KD on CIFAR-10 dataset. As shown in Figure 3.3, when λ_1 is 0.1 and λ_2 is 0.9 ($\tau = 4$) with KD, the accuracy of a student (WRN16-1)

trained with WRN16-3 as a teacher is the best. If λ_1 is small and λ_2 is large, the distillation effect of KD is increased. Since the accuracy depends on λ_1 and λ_2, I referred to previous studies (Cho and Hariharan (2019); Ji *et al.* (2021); Tung and Mori (2019)) to choose the popular parameters for experiments. The parameters of ($\lambda_1 = 0.1$, $\lambda_2 = 0.9$, $\tau = 4$), ($\lambda_1 = 0.4$, $\lambda_2 = 0.6$, $\tau = 16$), ($\lambda_1 = 0.7$, $\lambda_2 = 0.3$, $\tau = 16$), and ($\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $\tau = 4$) are used for KD on CIFAR-10, CINIC-10, Tiny-ImageNet, and ImageNet, respectively.

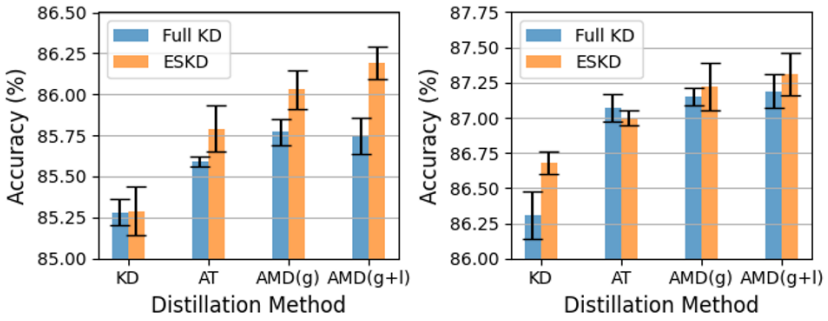
I perform baseline comparisons with traditional KD (Hinton *et al.* (2015)), attention transfer (AT) (Zagoruyko and Kmodakis (2017)), relational knowledge distillation (RKD) (Park *et al.* (2019b)), variational information distillation (VID) (Ahn *et al.* (2019)), similarity-preserving knowledge distillation (SP) (Tung and Mori (2019)), correlation congruence for knowledge distillation (CC) (Peng *et al.* (2019)), contrastive representation distillation (CRD) (Tian *et al.* (2019)), attentive feature distillation and selection (AFDS) (Wang *et al.* (2020c)), and attention-based feature distillation (AFD) (Ji *et al.* (2021)) that is a new feature linking method considering similarities between the teacher and student features, including state-of-the-art approaches. Note that, for fair comparison, the distillation methods are performed with traditional KD to see if they enhance standard KD, keeping the same setting as the proposed method. The hyperparameters of the methods follow their respective papers. For the proposed method, the constant parameter s and margin parameter m are 64 and 1.35, respectively. The loss weight γ of the proposed method is 5000. I determine the hyperparameters empirically, considering the distillation effects by the capacity of models. A more detailed description of parameters appears in section 3.5.5. All experiments were repeated five times, and the averaged best accuracy and the standard deviation of performance are reported.

No augmentation method is applied for CIFAR-10 and CINIC-10. For the proposed method, additional techniques, such as using the other hidden layers for generating better distillation effects from teachers or reshaping the dimension size of the feature maps, are not applied. All of the experiments are run on a 3.50 GHz CPU (Intel® Xeon(R) CPU E5-1650 v3), 48 GB memory, and NVIDIA TITAN Xp (3840 NVIDIA® CUDA® cores and 12 GB memory) graphic card (NVIDIA (2016)).

To obtain the best performance, I adopt early-stopped KD (ESKD) (Cho and Hariharan (2019)) for training teacher and student models, leveraging its effects across the board in improving the efficacy of knowledge distillation. As shown in Figure 3.4, the early stopped model of a teacher tends to train student models better than Full KD that uses a fully trained teacher.

3.5.3 Attention-based Distillation

In this section, I explore the performance of attention based distillation approaches with different types of combinations for teacher and student. I set four types of combinations for teacher and student that consist of the same or different structure of networks. The four types of combinations are described in Table 3.3. Since the proposed method is relevant to using attention maps, I implemented various baselines that are state-of-the-art attention based distillation methods, including AT (Zagoruyko and Kmodakis (2017)), AFDS (Wang *et al.* (2020c)), and AFD (Ji *et al.* (2021)). As described in section 3.2, AT (Zagoruyko and Kmodakis (2017)) uses activation-based spatial attention maps for transferring from teacher to student. AFDS (Wang *et al.* (2020c)) includes attentive feature distillation and accelerates the transfer-learned model by feature selection. Additional layers are used to calculate a transfer importance predictor used to measure the importance of the source activation maps and enforce a different penalty for training a student. AFD (Ji *et al.* (2021)) extracts



(a) T: WRN28-1, S: WRN16-1

(b) T: WRN16-8, S: WRN28-1

(c) T: WRN28-1, S: WRN16-1

(d) T: WRN28-3, S: WRN16-1

Figure 3.4: Accuracy (%) for Full KD and ESKD. (a) and (b) Are on CIFAR-10, and (c) and (d) Are on CINIC-10, Respectively. T and S Denotes Teacher and Student Models, Respectively.

channel and spatial attention maps and identifies similar features between teacher and student, which are used to control the distillation intensities for all possible pairs and compensate for the limitation of learning to transfer (L2T) (Jang *et al.* (2019)) using manually selected links. I implemented AFDS (Wang *et al.* (2020c)) when the dimension size of features for intermediate layers from the student is the same as the one from the teacher to concentrate on the distillation effects. I use four datasets that have varying degrees of difficulty in a classification problem. These baselines are used in the following experiments as well.

Table 3.4 presents the accuracy of various knowledge distillation methods for all setups in Table 3.3 on CIFAR-10 dataset. The proposed method, AMD (global+local),

Table 3.3: Details of Teacher and Student Network Architectures. ResNet (He *et al.* (2016)) and WideResNet (Zagoruyko and Komodakis (2016)) Are Denoted by ResNet (Depth) and WRN (Depth)-(Channel Multiplication), Respectively.

DB	Setup	Compression type	Teacher	Student	FLOPs (teacher)	FLOPs (student)	# of params (teacher)	# of params (student)	Compression ratio
CIFAR-10	(a)	Channel	WRN16-3	WRN16-1	224.63M	27.24M	1.50M	0.18M	11.30%
	(b)	Depth	WRN28-1	WRN16-1	56.07M	27.24M	0.37M	0.18M	47.38%
	(c)	Depth+Channel	WRN16-3	WRN28-1	224.63M	56.07M	1.50M	0.37M	23.85%
	(d)	Different architecture	ResNet44	WRN16-1	99.34M	27.24M	0.66M	0.18M	26.47%
CINIC-10	(a)	Channel	WRN16-3	WRN16-1	224.63M	27.24M	1.50M	0.18M	11.30%
	(b)	Depth	WRN28-1		56.07M		0.37M		47.38%
	(c^a)	Depth+Channel	WRN28-3		480.98M		3.29M		5.31%
	(d)	Different architecture	ResNet44		99.34M		0.66M		26.47%
Tiny-ImageNet	(a)	Channel	WRN16-3	WRN16-1	898.55M	108.98M	1.59M	0.19M	11.82%
	(b^b)	Depth	WRN40-1		339.60M		0.58M		32.52%
	(c^b)	Depth+Channel	WRN40-2		1,323.10M		2.27M		8.26%
	(d)	Different architecture	ResNet44		397.36M		0.67M		27.82%

has the best performing results in all cases. Table 3.5 describes the CINIC-10 results. In most cases, AMD (global+local) achieves the best results. For experiments on Tiny-ImageNet, as illustrated in Table 3.6, AMD outperforms previous methods, and AMD (global) shows better results in (a) and (b^b) setups. For (c^b) and (d) setups, AMD (global+local) provides better results. For experiments on ImageNet, standard KD is not applied to baselines and Full KD is utilized. Teacher and student networks are ResNet34 and ResNet18, respectively. The results of baselines are referred from prior works (Ji *et al.* (2021); Tian *et al.* (2019)). As described in Table 3.7, AMD (global) outperforms other distillation methods, increasing the top-1 and top-5 accuracy by 1.83% and 1.43% over the results of learning from scratch, respectively.

Table 3.4: Accuracy (%) on CIFAR-10 with Various Knowledge Distillation Methods. The Methods Denoted by "*" Are Attention Based Distillation. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively.

Setup	Method										
	Teacher	Student	KD	AT*	SP	RKD	VID	AFDS*	AFD*	AMD (g)	AMD (g+l)
(a)	87.76 ±0.12	84.11 ±0.12	85.29 ±0.15	85.79 ±0.14	85.69 ±0.11	85.45 ±0.09	85.40 ±0.14	–	86.23 ±0.13	86.28 ±0.06	**86.36** ±0.10
(b)	85.59 ±0.13	84.11 ±0.12	85.48 ±0.12	85.79 ±0.12	85.77 ±0.07	85.47 ±0.12	84.92 ±0.13	85.53 ±0.13	85.84 ±0.11	86.04 ±0.12	**86.10** ±0.10
(c)	87.76 ±0.12	85.59 ±0.12	86.57 ±0.16	86.77 ±0.11	86.56 ±0.09	86.38 ±0.22	86.64 ±0.24	–	87.24 ±0.03	87.13 ±0.14	**87.35** ±0.10
(d)	86.41 ±0.20	84.11 ±0.21	85.44 ±0.06	85.95 ±0.05	85.41 ±0.12	85.50 ±0.06	85.17 ±0.11	85.14 ±0.13	85.78 ±0.09	86.22 ±0.07	**86.34** ±0.05

Compared to KD, AT obtains better performance in most cases across datasets. That is, the attention map helps the teacher to transfer its knowledge. Even though there is a case that AT shows lower performance than KD in Table 3.6, AMD outperforms KD in all cases. It verifies that applying the discriminative angular distance metric for knowledge distillation maximizes the attention map's efficacy of transferring the knowledge and performs to complement the traditional KD for various combinations of teacher and student. The accuracies of SP with setup (a) and (d), and AFD with setup (d), are even lower than the accuracy of learning from scratch, while AMD performs better than other methods as shown in Table 3.6. When the classification problem is harder, AMD (global) can perform better than AMD (global+local)

Table 3.5: Accuracy (%) on CINIC-10 with Various Knowledge Distillation Methods. The Methods Denoted by "*" Are Attention Based Distillation. AMD Outperforms RKD (Park *et al.* (2019b)). "g" and "l" Denote Using Global and Local Feature Distillation, Respectively.

Setup	Method									
	Teacher	Student	KD	AT*	SP	VID	AFDS*	AFD*	AMD (g)	AMD (g+l)
(a)	75.40 ±0.12	72.05 ±0.12	74.31 ±0.10	74.63 ±0.13	74.43 ±0.14	74.35 ±0.05	–	74.13 ±0.12	75.04 ±0.11	**75.18** ±0.09
(b)	75.59 ±0.15		74.66 ±0.08	74.73 ±0.02	74.94 ±0.11	73.85 ±0.08	74.54 ±0.08	74.36 ±0.04	75.14 ±0.06	**75.21** ±0.04
(c[a])	76.97 ±0.05		74.26 ±0.06	74.19 ±0.11	75.05 ±0.10	74.06 ±0.15	–	74.20 ±0.12	74.72 ±0.07	**75.17** ±0.07
(d)	74.30 ±0.15		74.47 ±0.09	74.67 ±0.05	74.46 ±0.17	74.43 ±0.10	74.64 ±0.12	73.31 ±0.13	74.93 ±0.07	**75.10** ±0.10

in some cases. When the teacher and student have different channels or architectural styles, AMD (global+local) can generate a better student than AMD (global).

Components of AMD Loss Function. As described in Equation 3.7, angular margin distillation loss function ($\mathcal{L}_{AM}(Q_{Tp}, Q_{Tn}, Q_{Sp}, Q_{Sn})$) includes three components (A, P, N). To verify the performance of each component in AMD loss, I experiment with each component separately. As shown in Figure 3.5, among all components, (A) provides the strongest contribution. Each component in AMD contributes to improvements in performance, which transfers different knowledge. Adding one component to the other one provides richer information, which leads to better performance. The combination of all the components (AMD) show a much higher performance.

Table 3.6: Accuracy (%) on Tiny-ImageNet with Various Knowledge Distillation Methods. The Methods Denoted by "*" Are Attention Based Distillation. AMD Outperforms VID (Ahn *et al.* (2019)) and RKD (Park *et al.* (2019b)). "g" and "l" Denote Using Global and Local Feature Distillation, Respectively.

Setup	Method								
	Teacher	Student	KD	AT*	SP	AFDS*	AFD*	AMD (g)	AMD (g+l)
(a)	58.16 ±0.30	49.45 ±0.20	49.99 ±0.15	49.72 ±0.15	49.27 ±0.19	–	50.00 ±0.23	**50.32** ±0.07	49.92 ±0.04
(b^b)	54.74 ±0.24		49.56 ±0.17	49.79 ±0.22	49.89 ±0.20	49.46 ±0.28	50.04 ±0.27	**50.15** ±0.10	49.97 ±0.18
(c^b)	59.92 ±0.15		49.67 ±0.13	49.62 ±0.16	49.59 ±0.25	–	49.78 ±0.24	49.88 ±0.20	**50.07** ±0.10
(d)	54.66 ±0.14		49.52 ±0.16	49.45 ±0.28	49.13 ±0.20	49.55 ±0.13	49.44 ±0.27	49.92 ±0.09	**50.08** ±0.16

Table 3.7: Top-1 and Top-5 Accuracy (%) on ImageNet with Various Knowledge Distillation Methods. The Methods Denoted by "*" Are Attention Based Distillation. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively.

	Teacher	Student	KD	AT*	RKD	SP	CC	AFD*	CRD(+KD)	AMD (g)	AMD (g+l)
Top-1	73.31	69.75	70.66	70.70	70.59	70.79	69.96	71.38	71.17(71.38)	**71.58**	71.47
Top-5	91.42	89.07	89.88	90.00	89.68	89.80	89.17	–	90.13(90.49)	**90.50**	90.49

This result indicates that all components (AMD) are critical to distilling the best student model.

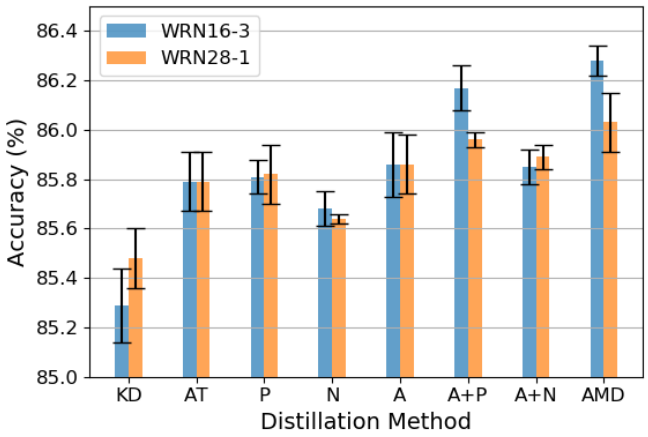


Figure 3.5: Accuracy (%) of Students (WRN16-1) Trained with Teachers (WRN16-3 and WRN28-1) on CIFAR-10 for Various Loss Functions.

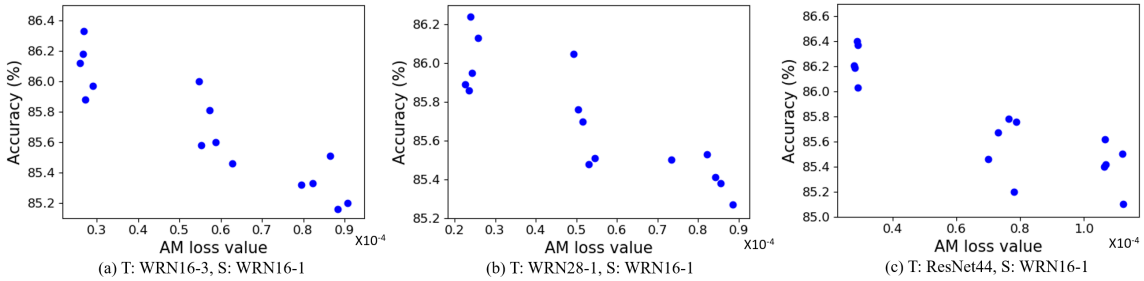


Figure 3.6: $\mathcal{L}_A$ Vs. Accuracy (%) for (from Left to Right) WRN16-1 Students (S) Trained with WRN16-3, WRN28-1, and ResNet44 Teachers (T), on CIFAR-10.

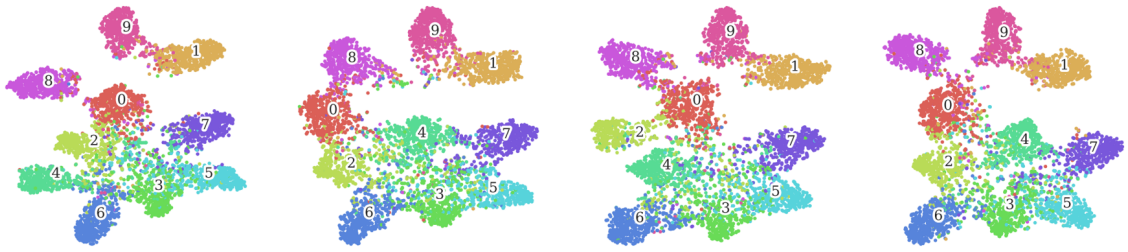


Figure 3.7: t-SNE Plots of Output for Teacher Model (ResNet44) and Students (WRN16-1) Trained with KD and AMD on CIFAR-10.

In Figure 3.6, I show $\mathcal{L}_{\mathcal{A}}$ vs. accuracy, when using KD, SP, and AMD (global), for WRN16-1 students trained with WRN16-3, WRN28-1, and ResNet44 teachers, on CIFAR-10 testing set. As shown in Figure 3.6, when the loss value is smaller, the accuracy is higher. Thus, these plots verify that $\mathcal{L}_{\mathcal{A}}$ and performance are correlated.

t-SNE Visualization and Cluster Metrics. To measure the clustering performance, I plot t-SNE (van der Maaten and Hinton (2008)) and calculate V-Score (Rosenberg and Hirschberg (2007)) of outputs from penultimate layers of KD and the proposed method on CIFAR-10, where V-Score is clustering metrics implying a higher value is better clustering. As shown in Figure 3.7, compared to KD, AMD helps get tighter clusters and better separation between classes as seen in higher V-Score.

3.5.4 Effect of Teacher Capacity

To understand the effect of the capacity of the teacher, I implemented various combinations of teacher and student, where the teacher has a different capacity. I use well-known benchmarks for image classification which are WRN (Zagoruyko and Komodakis (2016)), ResNet (He *et al.* (2016)), and MobileNetV2 (M.NetV2) (Sandler *et al.* (2018)). I applied the same settings as in the experiments of the previous section.

The results in classification accuracy for the student models are described in Table 3.8 across three datasets, trained with attention based and non-attention based methods (Hinton *et al.* (2015); Zagoruyko and Kmodakis (2017); Tung and Mori (2019)). The number of trainable parameters are noted in in brackets. For all cases, the proposed method, AMD, shows the highest accuracy. When the complexity of the dataset is higher and the depth of teacher is largely different from the one of the student, AMD (global) tends to generate a better student than AMD (global+local). When a larger capacity of students is used, the accuracy observed is higher. This is seen in the results from WRN16-1 and ResNet20 students with WRN16-3 and

Table 3.8: Accuracy (%) with Various Knowledge Distillation Methods for Different Combinations of Teachers and Students. "Teacher" and "Student" Denote Results of the Model Used to Train the Distillation Methods and Trained from Scratch, Respectively. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively.

Method	CIFAR-10				CINIC-10									Tiny-ImageNet		
Teacher	WRN 28-1 (0.4M, 85.84)	WRN 40-1 (0.6M, 86.39)	WRN 16-3 (1.5M, 88.15)	WRN 16-8 (11.0M, 89.50)	WRN 16-3 (1.5M, 75.65)	WRN 16-8 (11.0M, 77.97)	WRN 28-1 (0.4M, 73.91)	WRN 40-1 (0.6M, 74.49)	WRN 28-3 (3.3M, 77.14)	WRN 40-2 (2.2M, 76.66)	WRN 16-3 (1.5M, 75.65)	WRN 28-3 (3.3M, 77.14)	M.Net V2 (0.6M, 80.98)	WRN 40-1 (0.6M, 55.28)	WRN 40-2 (2.3M, 60.18)	WRN 16-3 (1.6M, 58.78)
Student	WRN16-1 (0.2M, 84.11±0.21)		WRN28-1 (0.4M, 85.59±0.13)		WRN16-1 (0.2M, 72.05±0.12)						ResNet20 (0.3M, 72.74±0.09)			WRN16-1 (0.2M, 49.45±0.20)		ResNet20 (0.3M, 51.75±0.19
KD	85.48 ±0.12	85.42 ±0.11	86.57 ±0.16	86.68 ±0.08	74.31 ±0.10	74.17 ±0.16	74.66 ±0.08	74.45 ±0.03	74.26 ±0.06	74.29 ±0.09	75.12 ±0.11	74.97 ±0.07	76.69 ±0.06	49.56 ±0.17	49.67 ±0.13	51.72 ±0.13
AT	85.79 ±0.12	85.79 ±0.11	86.77 ±0.11	87.00 ±0.05	74.63 ±0.13	74.23 ±0.14	74.73 ±0.02	74.55 ±0.06	74.19 ±0.11	74.48 ±0.08	75.33 ±0.11	75.18 ±0.09	77.34 ±0.10	49.79 ±0.22	49.62 ±0.16	51.65 ±0.05
SP	85.77 ±0.07	85.90 ±0.11	86.56 ±0.09	86.94 ±0.08	74.43 ±0.11	74.34 ±0.13	74.94 ±0.11	74.86 ±0.07	75.04 ±0.10	74.81 ±0.09	75.29 ±0.10	75.50 ±0.09	73.71 ±0.10	49.89 ±0.20	49.59 ±0.25	51.87 ±0.09
AMD (g)	86.04 ±0.12	86.03 ±0.09	87.13 ±0.14	87.22 ±0.17	75.04 ±0.11	74.93 ±0.09	75.14 ±0.06	**75.12** ±0.07	74.72 ±0.07	74.95 ±0.20	75.66 ±0.08	75.61 ±0.06	78.45 ±0.03	**50.15** ±0.11	49.88 ±0.20	51.89 ±0.25
AMD (g+l)	**86.10** ±0.10	**86.15** ±0.06	**87.35** ±0.10	**87.31** ±0.15	**75.18** ±0.09	**75.20** ±0.05	**75.21** ±0.04	75.10 ±0.04	**75.22** ±0.07	**75.04** ±0.06	**75.75** ±0.08	**75.76** ±0.11	**78.62** ±0.04	49.97 ±0.18	**50.07** ±0.10	**52.12** ±0.15

WRN28-3 teachers on CINIC-10 dataset. For the combinations, ResNet20 students having a larger capacity than WRN16-1 generate better results. Furthermore, on CIFAR-10, when a WRN16-3 teacher is used, a WRN28-1 student achieves 87.35% for AMD (global+local), whereas a WRN16-1 student achieves 86.36% for AMD (global+local). On Tiny-ImageNet, when AMD (global+local) is used, the accuracy of a ResNet20 student is 52.12%, which is higher than the accuracy of a WRN16-1 student, which is 49.92%.

Compared to KD, in most cases, AT achieves better performance. However, when the classification problem is difficult, such as when using Tiny-ImageNet, and when WRN40-2 teacher and WRN16-1 student are used, both AT and SP show worse performance than KD. When the WRN16-3 teacher and ResNet20 student are used, KD and AT perform worse than the model trained from scratch. The result of AT is even lower than that of KD. So, there are cases where AT and SP cannot complement the performance of the traditional KD. On the other hand, for the proposed method, the results are better than the baselines in all the cases. Interestingly, on CIFAR-10 and CINIC-10, the result of a WRN16-1 student trained by AMD with a WRN28-1 teacher is even better than the result of the teacher. Therefore, I conclude that the proposed method maximizes the attention map's efficacy of transferring the knowledge and complements traditional KD.

Also, when applying the larger teacher model and the smaller student model, the performance degradation of AMD can occur. For example, on CINIC-10, WRN16-1 student trained with WRN40-1 (0.6M) teacher outperforms the one trained with WRN40-2 (2.3M) teacher. Both AMD and other methods produce some cases with lower performance when a better (usually larger) teacher is used. This is consistent with prior findings (Cho and Hariharan (2019); Wang and Yoon (2021); Stanton *et al.* (2021)) that a better teacher does not always guarantee a better student.

Heterogeneous Teacher-student. In Table 3.8, I present the results of the teacher-student combinations from similar architecture styles. Tian *et al.* (Tian *et al.* (2019)) found that feature distillation methods such as SP sometimes struggled to find the optimal solution in different architecture styles. In this regard, I implemented heterogeneous teacher-student combination, where the teacher and student have very different structure of networks. I use vgg (Simonyan and Zisserman (2014)) network to compose heterogeneous combinations.

As describe in Table 3.9, I observe similar findings, showing degraded performance in using SP when vgg13 teacher and ResNet20 student are used, while AMD consistently outperforms all baselines I explored. Also, in most cases, WRN16-8 teacher distills a better student (vgg8) than WRN28-1 teacher. However, KD and SP shows better performance with WRN28-1 teacher, which corroborates a better teacher does not always distill a better student.

Table 3.9: Accuracy (%) with Various Knowledge Distillation Methods for Different Structure of Teachers and Students on CIFAR-10. "Teacher" and "Student" Denote Results of the Model Used to Train the Distillation Methods and Trained from Scratch, Respectively. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively.

Teacher	WRN28-1 (0.4M, 85.84)	WRN16-8 (11.0M, 89.50)	vgg13 (9.4M, 88.56)	M.NetV2 (0.6M, 89.61)
Student	vgg8 (3.9M, 85.41±0.06)		ResNet20 (0.3M, 85.20±0.17)	ResNet26 (0.4M, 85.65±0.20)
KD	86.93±0.11	86.74±0.13	85.39±0.07	87.74±0.08
AT	87.16±0.09	87.29±0.10	85.63±0.20	88.61±0.04
SP	87.29±0.02	86.82±0.07	85.00±0.07	85.78±0.10
AMD (g)	87.43±0.04	87.61±0.11	86.18±0.14	**88.70**±0.03
AMD (g+l)	**87.56**±0.03	**87.63**±0.07	**86.41**±0.04	88.42±0.08

3.5.5 Ablations and Sensitivity Analysis

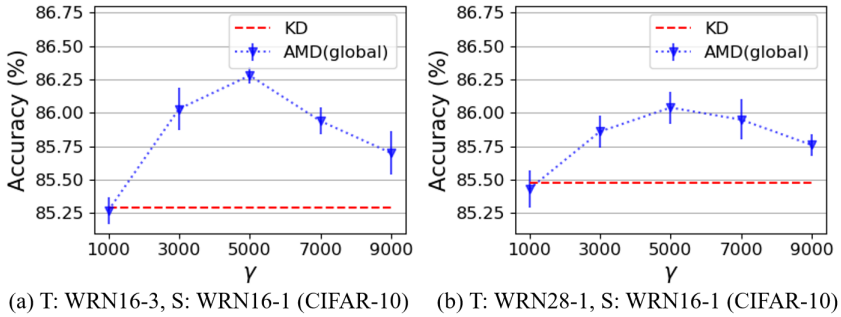
In this section, I investigate sensitivity for hyperparameters (γ and m) used for the angular margin based attention distillation.

Effect of Angular Distillation Hyperparameter γ

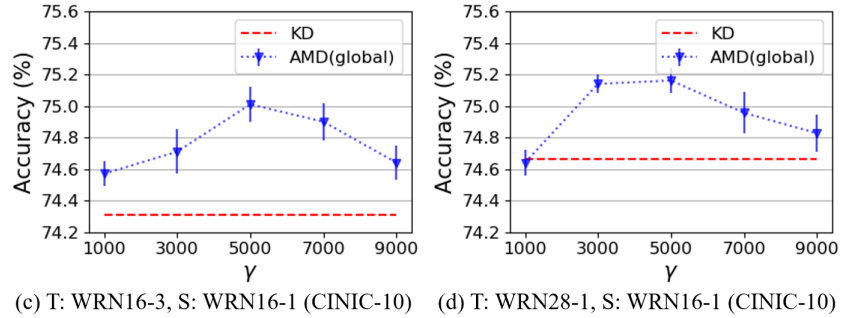
The results of a student model (WRN16-1) for AMD (global) trained with teachers (WRN16-3 and WRN28-1) by using various γ on CIFAR-10 (the first row) and CINIC-10 (the second row) are depicted in Figure 3.8 ($m = 1.35$). When γ is 5000, all results show the best accuracy. For CIFAR-10, when WRN16-3 is used as a teacher, the accuracy of $\gamma = 3000$ is higher than that of $\gamma = 7000$. However, for WRN28-1 as a teacher, the accuracy of $\gamma = 7000$ is higher than that of $\gamma = 3000$. When γ is 1000, the accuracy is lower than KD, implying that it does not complement KD and adversely affects the performance. On the other hand, for CINIC-10, when the WRN16-3 teacher is used, the result of $\gamma = 7000$ is better than that of $\gamma = 3000$. But, for the WRN28-1 teacher, $\gamma = 3000$ is higher than that of $\gamma = 7000$. Therefore, γ values between 3000 and 7000 achieve good performance, while too small or large γ values do not help much with improvement. Therefore, setting the proper γ value is important for performance. I recommend using γ as 5000, which produces the best results across datasets and combinations of teacher and student.

Effect of Angular Margin m

The results of a student model (WRN16-1) for AMD (global) trained with teachers (WRN16-3 and WRN28-1) by various angular margin m on CIFAR-10 (the first row) and CINIC-10 (the second row) are illustrated in Figure 3.9 ($\gamma = 5000$). As described in section 3.4.2, using the large value of m corresponds to producing more distinct positive features in the attention map and making a large gap between positive and negative features for distillation. When m is 1.35 for the WRN16-3 teacher, the WRN16-1 student shows the best performance of 86.28% on CIFAR-10. When $m = 1.5$ for CINIC-10, the student's accuracy is 75.13%, which is higher than when $m =$



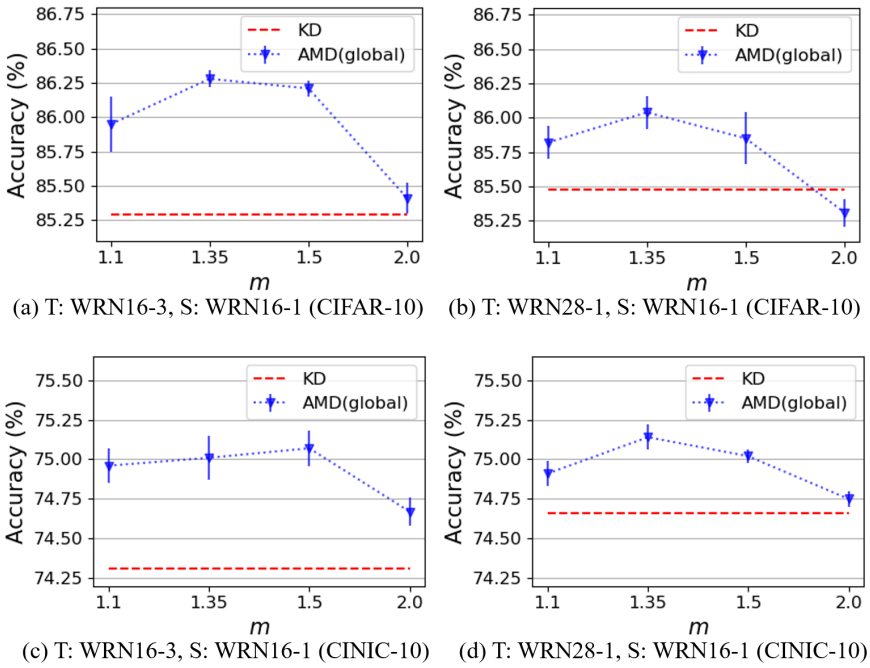
(a) T: WRN16-3, S: WRN16-1 (CIFAR-10) (b) T: WRN28-1, S: WRN16-1 (CIFAR-10)



(c) T: WRN16-3, S: WRN16-1 (CINIC-10) (d) T: WRN28-1, S: WRN16-1 (CINIC-10)

Figure 3.8: Accuracy (%) of Students (WRN16-1) for AMD (global) with Various γ, Trained with Teachers (WRN16-3 and WRN28-1) on CIFAR-10 and CINIC-10. "T" and "S" Denote Teacher and Student, Respectively.

1.35. When the teacher is WRN28-1, the student produces the best accuracy with m = 1.35 on both datasets. The student model with $m = 1.35$ performs better than the one with $m = 1.1$ and 2.0. When the complexity of the dataset is higher, using m (1.5) which is larger than 1.35 can produce a good performance. When $m = 1.0$ (no additional margin applied to the positive feature) for CIFAR-10 and CINIC-10 with setup (b), the results are 85.81% and 74.83%, which are better than those of 85.31% and 74.75% from $m = 2.0$, respectively. This result indicates that it is important to set an appropriate m value for the proposed method. I believe that angular margin plays a key role in determining the gap between positive and negative features. As angular margin increases, the positive features are further emphasized, and in this case of over-emphasis by a much larger m, the performance is worse than that of the



(a) T: WRN16-3, S: WRN16-1 (CIFAR-10) (b) T: WRN28-1, S: WRN16-1 (CIFAR-10)

(c) T: WRN16-3, S: WRN16-1 (CINIC-10) (d) T: WRN28-1, S: WRN16-1 (CINIC-10)

Figure 3.9: Accuracy (%) of Students (WRN16-1) for AMD (global) with Various Angular Margin m, Trained with Teachers (WRN16-3 and WRN28-1) on CIFAR-10 and CINIC-10. "T" and "S" Denote Teacher and Student, Respectively.

smaller m. I recommend using a margin m of around 1.35 ($m > 1.0$), which generates the best results in most cases.

3.5.6 Analysis with Activation Maps

To analyze results with intermediate layers, I adopt Grad-CAM (Selvaraju *et al.* (2017)) which uses class-specific gradient information to visualize the coarse localization map of the important regions in the image. In this section, I present the activation maps from intermediate layers and the high level of the layer with various methods. The red region is more crucial for the model prediction than the blue one.

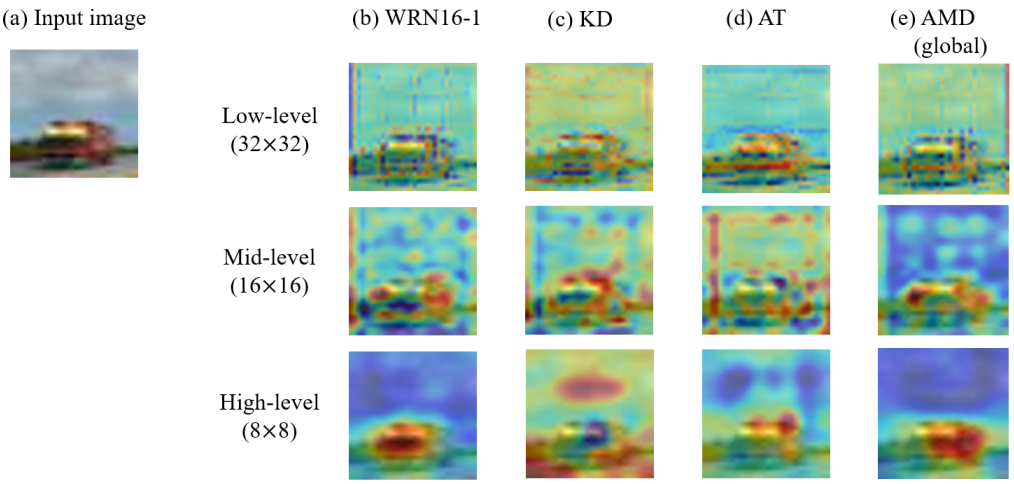


Figure 3.10: Activation Maps for Different Levels of Students (WRN16-1) Trained with a Teacher (WRN16-3) on CIFAR-10.

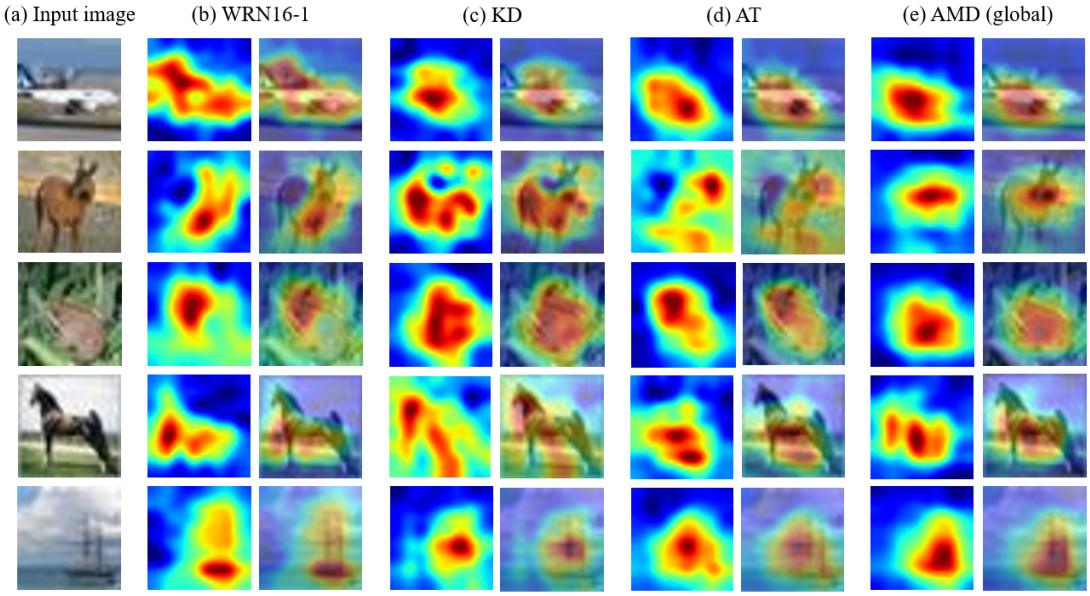


Figure 3.11: Activation Maps of High-level from Students (WRN16-1) Trained with a Teacher (WRN16-3) for Different Input Images on CIFAR-10.

Activation Maps for the Different Levels of Layers

The activation maps from intermediate layers with various methods are shown in Figure 3.10. The proposed method, AMD, shows intuitively similar activated regions

to the traditional KD (Hinton *et al.* (2015)) in the low-level. However, at mid-level and high-level, the proposed method represents the higher activations around the region of a target object, which is different from the previous methods (Hinton *et al.* (2015); Zagoruyko and Kmodakis (2017)). Thus, the proposed method can classify positive and negative areas more discriminatively, compared to the previous methods (Hinton *et al.* (2015); Zagoruyko and Kmodakis (2017)). The high-level activation maps with various input images are described in Figure 3.11. The activation from proposed method is seen to be more centered on the target. The result shows that the proposed method performs better in focusing on the foreground object distinctly with high weight, while being less distracted by the background compared to other methods (Hinton *et al.* (2015); Zagoruyko and Kmodakis (2017)). With higher weight over regions of interest, the student from the proposed method has a stronger discrimination ability. Therefore, the proposed method guides student models to increase class separability.

Activation Maps for Global and Local Distillation of AMD

To investigate the impact of using global and local features for AMD, I illustrate relevant results in Figure 3.12. When both global and local features are used for distillation, the activated area is located and shaped more similar to the teacher, than using the global feature only. Also, AMD (global+local) focuses more on the foreground object with higher weights than AMD (global). AMD (global+local) guides the student to focus more on the target regions and finds discriminative regions. Thus, using global and local features is better than using global features alone for the proposed method.

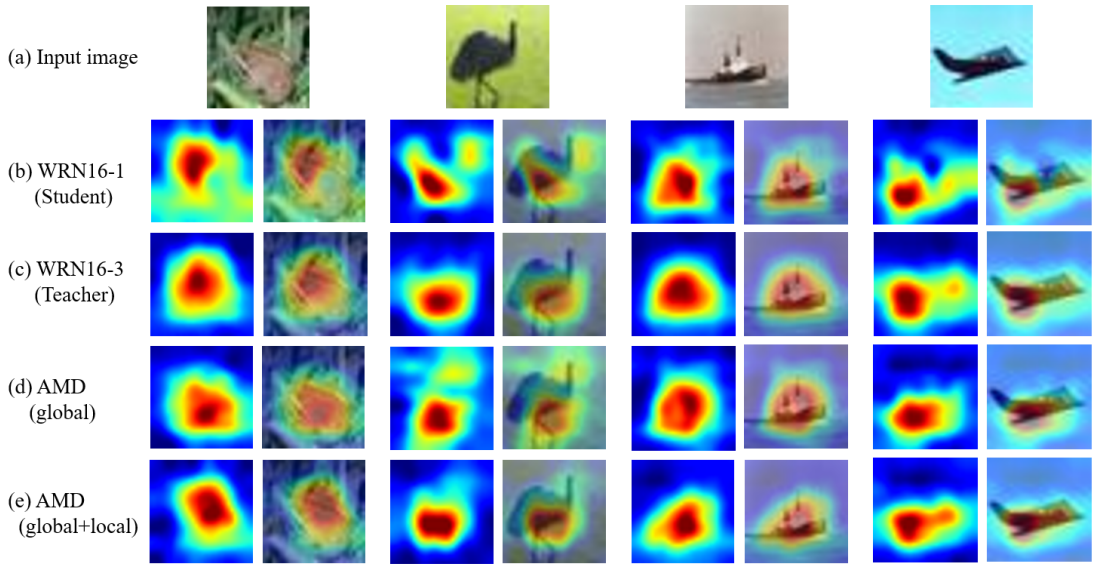


Figure 3.12: Activation Maps of High-level from Students (WRN16-1) for AMD Trained with a Teacher (WRN16-3) for Different Input Images on CIFAR-10.

3.5.7 Combinations with Existing Methods

Even if a model shows good performance in classification, it may have miscalibration problems (Guo *et al.* (2017)) and may not always obtain improved results from combining with other robust methods. In this section, to evaluate the generalizability of models trained by each method and to explore if the method can complement other methods, I implement experiments with various existing methods. I use the method in various ways to demonstrate how easily it can be combined with any previous learning tasks. I trained students with fine-grained features (Wang *et al.* (2019, 2020a)), augmentation methods, and one of the baselines such as SP (Tung and Mori (2019)) that is not based on the attention feature based KD. WRN16-1 students were trained with WRN16-3 and WRN28-1 teachers. I examine whether the proposed method can be combined with other techniques and compare the results to baselines.

Fine-grained Feature-based Distillation

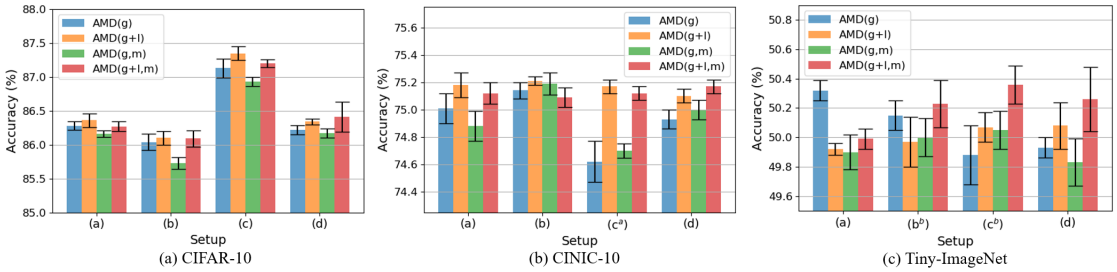


Figure 3.13: Accuracy (%) from Students (WRN16-1) for AMD Trained with a Teacher (WRN16-3) With/Without Masked Features. "g", "l", and "m" Denote Global, Local, and Masked Feature, Respectively.

If the features of teacher and student are compatible, it results in a student achieving 'minor gains' (Wang *et al.* (2019)). To perform better distillation and to overcome the problem of learning minor gains, a technique for generating a fine-grained feature has been used (Wang *et al.* (2019, 2020a)). For distillation with AMD and creating the fine-grained (masked) feature, a binary mask is adopted when the negative feature is created. For example, if the probability of the point for the negative map is higher than 0.5, the point is multiplied by 1, otherwise by 0. Then, compared to non-masking, it boosts the difference between teacher and student, where the difference can be more focused on loss function for training. The results for AMD with or without using masked feature-based distillation are presented in Figure 3.13. The parameter γ for training a student based on AMD without masked features is 5000 for all setups across datasets. When masked features are used for AMD, to generate the best results, γ of 3000 is applied to setup (b) on CIFAR-10, setup (c^a) on CINIC-10, and all setups on Tiny-ImageNet. For CIFAR-10, AMD (global+local) without masked features has the best performing result in most cases. AMD (global+local) with masked features shows the best with setup (d). For CINIC-10, the results of

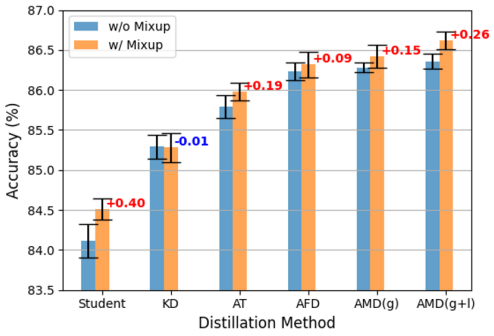
AMD with masked features for setup (d) show the best. For Tiny-ImageNet, in most cases, AMD with masked features performs the best. Therefore, when the complexity of a dataset is high, fine-grained features can help more effectively improve the performance, and the smaller parameter of γ, 3000, generates better accuracy. Also, AMD (global+local) with masked features produces better performance than AMD (global) with the one. For setup (d) – different architectures for teacher and student – with/without masked features, AMD (global+local) outperforms AMD (global). This could be due to the fact that the teacher's features differ from the student's because the two networks have different architectures, resulting in different distributions. So, masked features with both global and local distillation influence more on setup (d) than other setups. The difference between AMD (global) and AMD (global+local) with masked features is also discriminatively shown with the harder problem in classification. If the student's and teacher's architectural styles are similar, the student is more likely to achieve plausible results (Wang and Yoon (2021)).

Applying Augmentation Methods

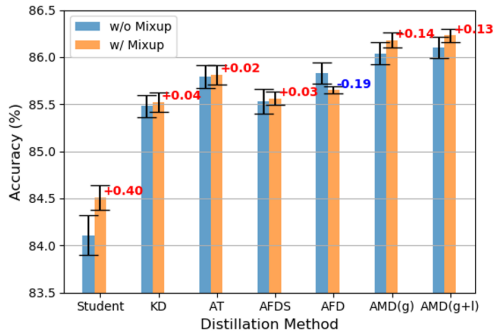
In this section, I investigate of the compatibility with different types of augmentation methods.

Mixup. Mixup (Zhang *et al.* (2018a)) is one of the most commonly used augmentation methods. I demonstrate here that AMD complements Mixup. Mixup's parameter is set to $\alpha_{\text{Mixup}} = 0.2$. A teacher is trained with the original training set and learns from scratch. A student is trained with Mixup and the teacher model is implemented as a pre-trained model.

As described in Figure 3.14, with Mixup, most of the methods generate better results. However, KD shows slight degradation when a WRN16-3 teacher is used. This degradation might be related to the artificially blended labels by Mixup. Conventional



(a) T: WRN16-3, S: WRN16-1



(b) T: WRN28-1, S: WRN16-1

Figure 3.14: Accuracy (%) of Students (WRN16-1) for Knowledge Distillation Methods, Trained with Mixup and a Teacher (WRN16-3) on CIFAR-10. "T" and "S" Denote Teacher and Student, Respectively. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively. "Student" Is a Result of WRN16-1 Trained from Scratch.

KD achieves the success by transferring concise logit knowledge. However, with Mixup in KD, the knowledge from a teacher is affected by the mixed labels and is not concise logits, which can hurt distillation quality (Das *et al.* (2020)). So, the knowledge for separating different classes can be better encoded by traditional KD (without Mixup) (Das *et al.* (2020)). Even though the KD performs degradation with Mixup, all other baselines and proposed methods transferring features with intermediate layers show improvement. Thus, the feature based distillation methods help to reduce the negative effects from noisy logits. When a WRN28-1 teacher is used, the performance of the student from AFD is degraded. AFD utilizes similarity of features for all possible pairs of the teacher and student. For this combination, Mixup produces noisy features, which can affect to mismatch the pair for distillation to perform degradation. Compared to the baselines, AMD obtains more gains from Mixup. To study the generalizability and regularization effects of Mixup, I measured expected calibration error (ECE) (Guo *et al.* (2017); Naeini *et al.* (2015)) and negative log likelihood

Table 3.10: ECE (%) and NLL (%) for Various Knowledge Distillation Methods with Mixup on CIFAR-10. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively. The Results (ECE, NLL) for WRN16-3 and WRN28-1 Teachers Are (1.469%, 44.42%) and (2.108%, 64.38%), Respectively.

Setup	Method	w/o Mixup		w/ Mixup	
		ECE	NLL	ECE	NLL
	Student	2.273	70.49	7.374 (+5.101)	90.58 (+20.09)
(a)	KD (Hinton *et al.* (2015))	2.065	63.34	1.818 (-0.247)	55.62 (-7.71)
	AT (Zagoruyko and Kmodakis (2017))	1.978	60.48	1.652 (-0.326)	50.84 (-9.64)
	AFD (Ji *et al.* (2021))	**1.890**	**56.71**	1.651 (-0.240)	50.22 (-6.49)
	AMD (g)	1.933	59.67	1.645 (-0.288)	50.33 (-9.34)
	AMD (g+l)	1.895	57.60	**1.592** (-0.304)	**49.68** (-7.92)
(b)	KD (Hinton *et al.* (2015))	2.201	68.75	1.953 (-0.249)	58.81 (-9.93)
	AT (Zagoruyko and Kmodakis (2017))	2.156	67.14	1.895 (-0.261)	56.51 (-10.62)
	AFDS (Wang *et al.* (2020c))	2.197	68.53	1.978 (-0.219)	58.86 (-9.68)
	AFD (Ji *et al.* (2021))	2.143	66.05	1.900 (-0.243)	57.68 (-8.37)
	AMD (g)	**2.117**	**66.47**	1.869 (-0.248)	56.05 (-10.42)
	AMD (g+l)	2.123	67.51	**1.853** (-0.270)	**55.15** (-12.36)

(NLL) (Guo *et al.* (2017)) for each method. ECE is a metric to measure calibration, representing the reliability of the model (Guo *et al.* (2017)). A probabilistic model's quality can be measured by using NLL (Guo *et al.* (2017)). The results of training from scratch with Mixup show a higher ECE and NLL than the results of training without Mixup, as seen in Table 3.10. However, the methods, including knowledge distillation, generate lower ECE and NLL. This implies that knowledge distillation from teacher to student influences the generation of a better model not only for accuracy but also for reliability. In both (a) and (b), with Mixup, AMD (global+local)

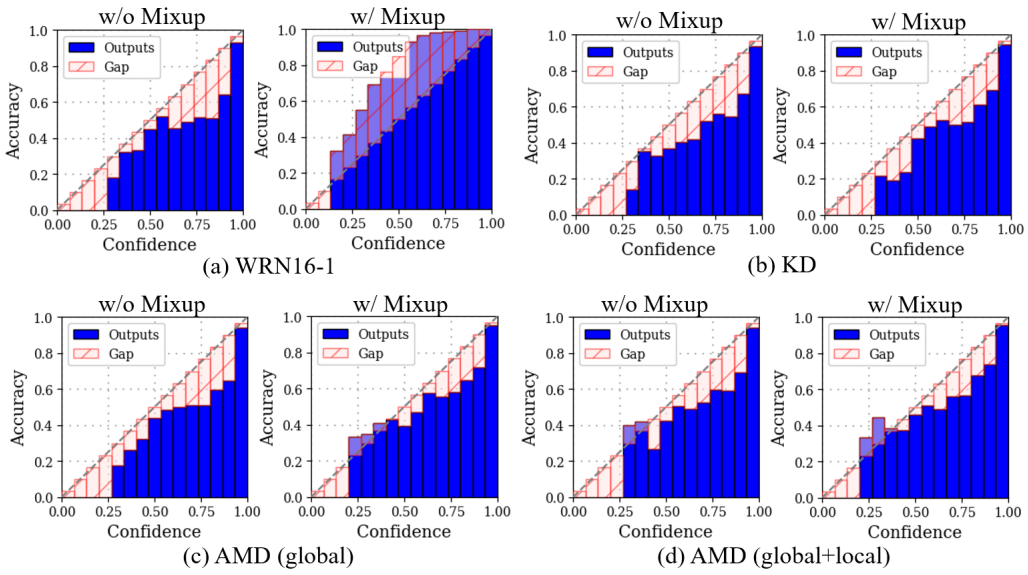


Figure 3.15: Reliability Diagrams of Students (WRN16-1) for Knowledge Distillation Methods, Trained with Mixup and a Teacher (WRN16-3) on CIFAR-10. For the Results of Each Method, the Left Is the Result Without Mixup, and the Right Is with Mixup.

shows robust calibration performance. Therefore, I confirm that an augmentation method such as Mixup gets the benefits from AMD in generating better calibrated performance. As can be seen in Figure 3.15, WRN16-1 trained from scratch with Mixup produces underconfident predictions (Zhang *et al.* (2018a)), compared to KD (Hinton *et al.* (2015)) with Mixup. AMD (global+local) with Mixup achieves the best calibration performance. These results support the advantage of AMD, that it can be easily combined with common augmentation methods to improve the performance in classification with good calibration.

CutMix. CutMix (Yun *et al.* (2019)) one of the most popular augmentation methods, which is more advanced method to Mixup. I evaluate AMD with CutMix. I referred to the previous study to set the parameters for CutMix (Yun *et al.* (2019)). As illustrated in Figure 3.16, all methods are improved by CutMix. Compared to other

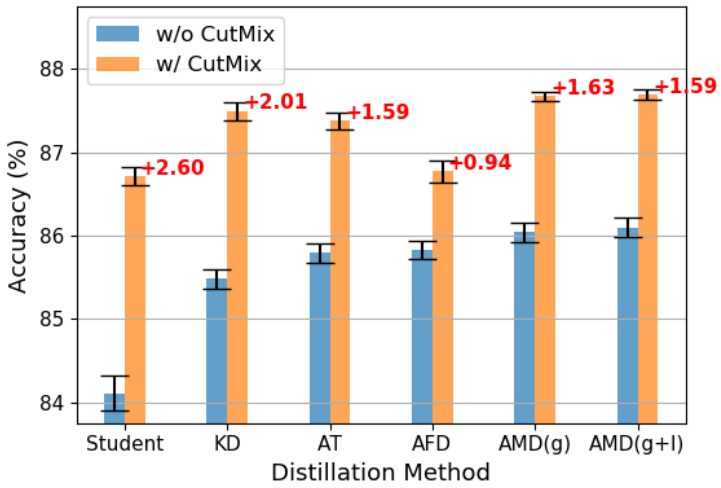


Figure 3.16: Accuracy (%) of Students (WRN16-1) for Knowledge Distillation Methods, Trained with Cutmix and a Teacher (WRN28-1) on CIFAR-10. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively. "Student" Is a Result of WRN16-1 Trained from Scratch.

baselines, AFD gains less improvement. Both AMD (global) and AMD (global+local) perform better with CutMix and these results also show that the proposed method can be easily combined with the advanced augmentation methods.

MoEx. To test with a latent space augmentation method, MoEx (Li *et al.* (2021a)) is adopted to train with AMD, which is one of the state-of-the-art technique for augmentation. I applied the same parameter by referring to the prior study (Li *et al.* (2021a)). I apply MoEx to a layer before stage 3 in the student network (WRN16-1), which achieves the best with KD.

As shown in Figure 3.17, most of KD based methods with MoEx perform better than the one without MoEx. AFD shows degradation. Since AFD transfers the knowledge considering all pair of features from teacher and student, MoEx in AFD hinders the pair matching and transferring the high quality knowledge. Both AMD

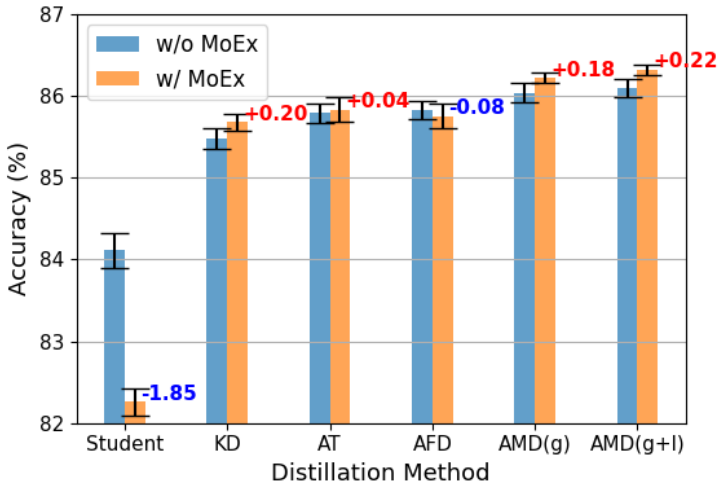


Figure 3.17: Accuracy (%) of Students (WRN16-1) for Knowledge Distillation Methods, Trained with MoEx and a Teacher (WRN28-1) on CIFAR-10. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively. "Student" Is a Result of WRN16-1 Trained from Scratch.

(global) and AMD (global+local) outperform baselines. This results verify that latent space augmentation based methods can be combined with the proposed method. Therefore, the proposed method can implement with various augmentation methods to improve the performance.

Additionally, I explore the work of MoEx at different layers. As described in Figure 3.18, when MoEx is applied the layer before stage 3 of the student model, AMD shows the best performance. KD also shows its best when MoEx is applied to a layer before stage 3. This aspect is different from the result of learning from scratch, which shows the best when MoEx is applied to a layer before stage 1 (Li *et al.* (2021a)). Thus, when latent space augmentation is combined with KD based method including baselines and the proposed method, a layer to apply augmentation

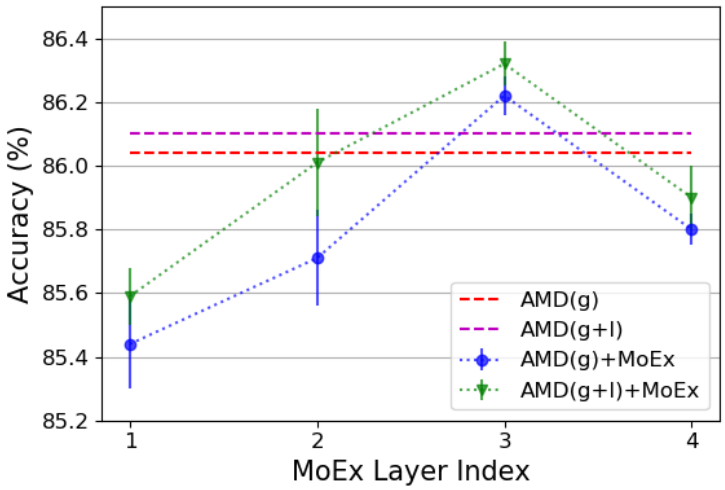
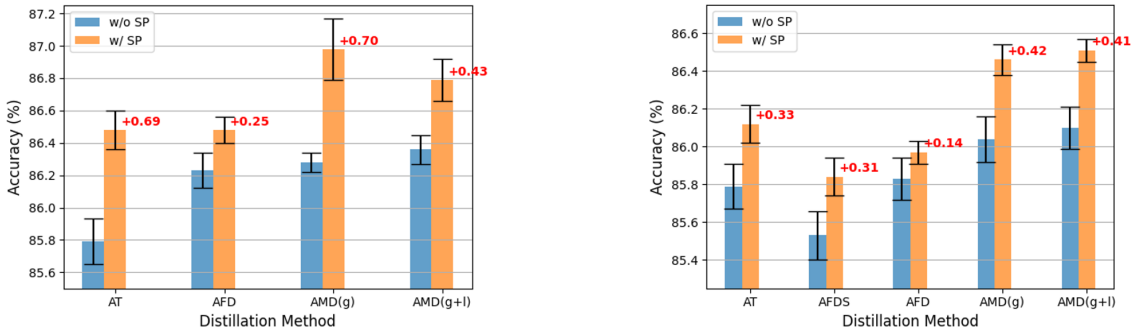


Figure 3.18: Accuracy (%) of Students (WRN16-1) for Knowledge Distillation Methods, Trained with MoEx and a Teacher (WRN28-1) on CIFAR-10. I Denote the Layer Index to Apply MoEx as (1=before Stage 1, 2=before Stage 2, 3=before Stage 3, 4=after Stage 3). "g" and "l" Denote Using Global and Local Feature Distillation, Respectively.

method has to be chosen considerably. And, these results imply that a layer before stage 3 plays a key role for knowledge distillation.

Combination with Other Distillation Methods

To demonstrate how AMD can perform with the other distillation methods, I adopt SP (Tung and Mori (2019)) which is not an attention based distillation method. A teacher is trained with the original training set and learns from scratch. SP (Tung and Mori (2019)) is applied while a student is being trained. I compare with baselines, depicted in Figure 3.19. In all cases, with SP, the accuracy is increased. Compared to the other attention based methods, AMD gets more gains by SP. Therefore, AMD can be enhanced and can perform well with the other distillation methods such as SP.



(a) T: WRN16-3, S: WRN16-1

(b) T: WRN28-1, S: WRN16-1

Figure 3.19: Accuracy (%) of Students (WRN16-1) for Knowledge Distillation Methods, Trained with SP and a Teacher (WRN16-3) on CIFAR-10. "T" and "S" Denote Teacher and Student, Respectively. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively. "Student" Is a Result of WRN16-1 Trained from Scratch.

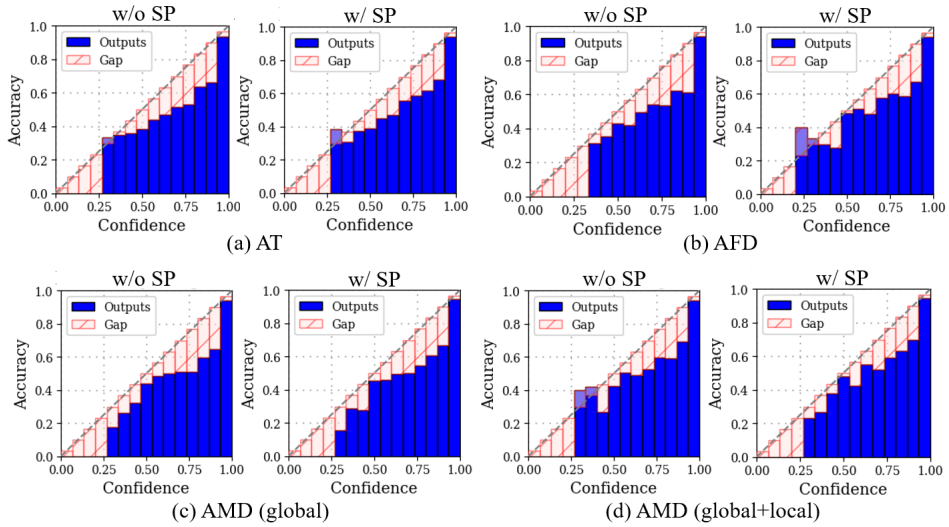


Figure 3.20: Reliability Diagrams of Students (WRN16-1) for Knowledge Distillation Methods, Trained with SP and a Teacher (WRN16-3) on CIFAR-10. For the Results of Each Method, the Left Is the Result Without SP, and the Right Is with SP.

I additionally analyzed the reliability described in Table 3.11. AMD (global+local) with SP shows the lowest ECE and NLL values. It verifies that AMD with SP can

Table 3.11: ECE (%) and NLL (%) for Various Knowledge Distillation Methods with SP on CIFAR-10. "g" and "l" Denote Using Global and Local Feature Distillation, Respectively. The Results (ECE, NLL) for WRN16-3 and WRN28-1 Teachers Are (1.469%, 44.42%) and (2.108%, 64.38%), Respectively.

Setup	Method	w/o SP		w/ SP	
		ECE	NLL	ECE	NLL
(a)	AT (Zagoruyko and Kmodakis (2017))	1.978	60.48	1.861 (-0.118)	56.22 (-4.26)
	AFD (Ji *et al.* (2021))	**1.890**	**56.71**	1.881 (-0.010)	56.73 (-0.02)
	AMD (g)	1.933	59.67	1.808 (-0.125)	54.74 (-4.93)
	AMD (g+l)	1.895	57.60	**1.803** (-0.092)	**53.80** (-3.80)
(b)	AT (Zagoruyko and Kmodakis (2017))	2.156	67.14	2.095 (-0.060)	65.38 (-1.75)
	AFDS (Wang *et al.* (2020c))	2.197	68.53	2.128 (-0.069)	66.61 (-1.92)
	AFD (Ji *et al.* (2021))	2.143	66.05	2.118 (-0.024)	65.39 (-0.66)
	AMD (g)	**2.117**	**66.47**	2.058 (-0.059)	63.37 (-3.10)
	AMD (g+l)	2.123	67.51	**2.043** (-0.080)	**63.23** (-4.28)

generate a model having higher reliability with better accuracy. Thus, the proposed method can be used with an additional distillation method. Also, the proposed method with SP can perform with different combinations of teacher and student with well-calibrated results. As illustrated in Figure 3.20, with SP (Tung and Mori (2019)), AT (Zagoruyko and Kmodakis (2017)) and AFD (Ji *et al.* (2021)) produce more overconfident predictions, compared to AMD (global+local) with SP (Tung and Mori (2019)) that gives the best calibration performance. Conclusively, these empirical findings reveal that AMD can perform with other distillation methods such as SP (Tung and Mori (2019)) to generate more informative features for distillation from teacher to student.

3.6 Conclusion

In this chapter, I proposed a new type of distillation loss function, AMD loss, which uses the angular distribution of features. I validated the effectiveness of distillation with this loss, under the setting of multiple teacher-student architecture combinations of KD in image classification. Furthermore, I have confirmed that the proposed method can be combined with previous methods such as fine-grained feature, various augmentation methods, and other types of distillation methods.

In future work, I aim to extend the proposed method to explore the distillation effects with different hypersphere feature embedding methods (Wang *et al.* (2018b); Deng *et al.* (2019)). Also, I plan to extend AMD to different approaches in image classification, such as vision transformer (Dosovitskiy *et al.* (2020)) and MLP-mixer (Tolstikhin *et al.* (2021)) that are not based on convolutional neural network. In addition, the proposed approach could provide insights for further advancement in other applications such as object detection and semantic segmentation.

Chapter 4

TOPOLOGICAL PERSISTENCE GUIDED KNOWLEDGE DISTILLATION FOR WEARABLE SENSOR DATA

4.1 Introduction

Wearable sensor data, used with deep learning methods, has achieved great successes in various fields such as smart homes, health-care services, and intelligent surveillance (Nweke *et al.* (2018)). However, analysis of wearable sensor data suffers from particular challenges because of inter- and intra-person variability and noisy signal problems (Seversky *et al.* (2016); Edelsbrunner and Harer (2022)). To mitigate these problems, utilizing invariant features obtained by topological data analysis (TDA) has been proposed as a solution and has proven beneficial (Seversky *et al.* (2016)). TDA in fusion with machine learning methods has achieved significant results in stock market analysis (Gholizadeh and Zadrozny (2018); Yen and Cheong (2021)), time-series forecasting (Zeng *et al.* (2021)), disease classification (Pachauri *et al.* (2011); Nawar *et al.* (2020)), and texture classification (Edelsbrunner and Harer (2022)).

TDA has been used to characterize the shape of complex data with the persistence of connected components and high-dimensional holes which are decoded by the persistent homology (PH) algorithm (Edelsbrunner and Harer (2022)). The persistence information can be represented by features such as persistence image (PI) (Adams *et al.* (2017)). However, I found two key challenges in utilizing topological features: (1) because of the large computational memory and time consumption to extract persistence features from large-scale data (Som *et al.* (2020)), it is challenging to

implement on small devices with limited computational power. Utilizing two modalities also requires an increase in the computational power to store and interpret the data. (2) because of a significant modality gap between the one-dimensional signal and two-dimensional TDA feature representation, it is difficult to integrate them in a unified framework. These differences in feature representations make conventional models difficult to use for fusing these different representations.

Based on these observations, in this chapter, I propose a new framework in knowledge distillation, which enables the student to acquire benefits from both teachers trained with different modality – time series and persistence image. Knowledge distillation (KD) has been utilized to generate a smaller model (student) from the learned knowledge of a larger model (teacher) (Hinton *et al.* (2015)). It has been demonstrated to have outstanding performance in the analysis of wearable sensor data (Jeon *et al.* (2022b); Som *et al.* (2020); Gou *et al.* (2021)). Also, using multi-teachers in KD has been studied to provide richer information, which is generally implemented with uni-modal data (Reich *et al.* (2020); Liu *et al.* (2020); Gou *et al.* (2021)). Given this insight and to resolve the mentioned issues, I develop a framework preforming with multi-modal data in KD using different teachers to distill a small model, named Topological Persistence Guided Knowledge Distillation (TPKD). An overview of the TPKD is presented in Figure 4.1. As shown in the figure, firstly, I extract PIs from persistence diagrams with TDA. I then train two models with time series data and PIs, respectively. Secondly, I use the two pre-trained models as teachers separately in KD. The features from intermediate layers are transformed to a correlation map, reflecting the similarities of samples for a mini-batch in the activations of the network, and the maps from teachers are merged for integrating the features for distillation. However, since features are from different modalities, it is hard to guarantee that the simply fused activation map from teachers is properly correlated with the one from

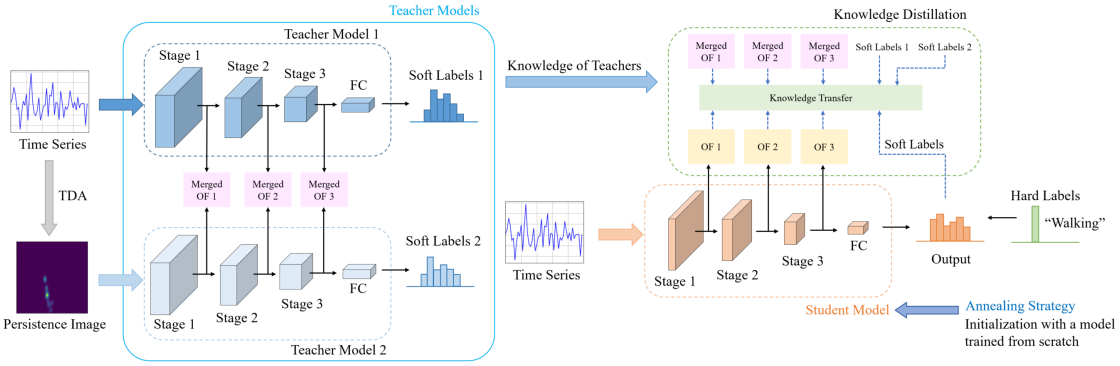


Figure 4.1: An Overview of Topological Persistence Guided Knowledge Distillation (TPKD). Two Teachers, Learned with Different Representations of the Same Raw Time Series Data, Are Utilized to Train a Compact Student Model.

the student, which is a hindrance for training the student (Gou *et al.* (2021)). To better accommodate information from different modalities, I construct a new form of knowledge utilizing orthogonal features (OF), representing prominent relationships between features by deformation of activation maps, which provides disentangled representation to incorporate different desirable properties involved in multimodality (Wang *et al.* (2020b)). Based on the better expressive knowledge implying feature relationships by OF, the student can easily learn from the teachers. In the third step, to reduce the knowledge gap and consider the properties inherent in the model using time series as an input, I apply an annealing strategy in KD. The annealing strategy guides the student model to initialize its weights from a model learned from scratch, instead of random initialization. Finally, a robust and small model is distilled by the proposed method, which uses the raw time-series data only as its input.

The contributions of this chapter are as follows:

- I propose a new framework based on knowledge distillation that transfers topological features to the student using time-series data only as an input.

- I develop a technique for leveraging orthogonal features from intermediate layers and an annealing strategy in KD with multiple teachers, which reduces the statistical gap in features between teachers and student for better knowledge transfer.

- I show strong empirical results demonstrating the strength of our approach with various teacher-student combinations on wearable sensor data for human activity recognition.

The rest of this chapter is organized as follows. In section 4.2, I provide a brief overview of generating PIs, KD techniques, and an annealing strategy. In section 4.3, I introduce the proposed method, a new framework in KD. In section 4.4, I describe our experimental results and analysis. In section 4.5, I discuss our findings and conclusions.

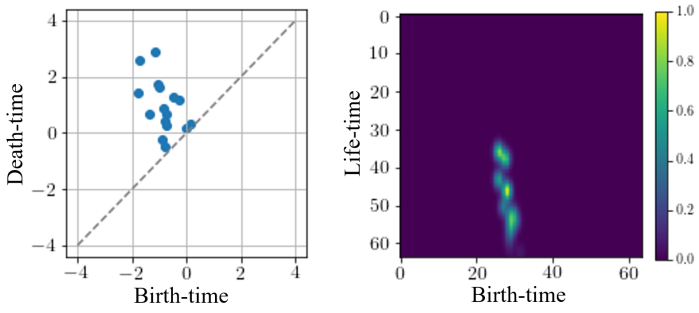
4.2 Background

4.2.1 Topological Feature Extraction

TDA has been utilized in various fields (Adams *et al.* (2017); Wang *et al.* (2021); Gholizadeh and Zadrozny (2018)). It has achieved great successes with providing novel insight on the shape of complex data, particularly in machine learning for different applications (Gholizadeh and Zadrozny (2018); Zeng *et al.* (2021); Som *et al.* (2020)). As a key algorithm of TDA, persistent homology tracks the variations in n-dimensional holes present in data, characterized by points, edges, and triangles by a dynamic thresholding process, which is called a filtration (Edelsbrunner *et al.* (2002)). The persistence of these topological cavities during a filtration is described in persistence feature, such as persistence diagram (PD) which encodes the birth and death times as x and y coordinates of planar scatter points (Adams *et al.* (2017); Edels-

brunner and Harer (2022)). Utilizing PDs directly in machine learning is challenging because of their heterogeneous nature, implying that the number and locations of the scatter points are not fixed and can be different at the presence of slight perturbations on the underlying data. Organizing the scatter points based on their persistence (life time) provides a way to vectorize the PDs.

Persistence image (PI) is a vector representation of PD, which represents the life-time of homological structures in data. Firstly, to construct the PI, PD is projected into a persistence surface (PS) $\rho : \mathbb{R} \to \mathbb{R}^2$, defined by a weighted sum of Gaussian functions centered at the scatter points in the PD. The PS is discretized and results in a grid. By integrating the PS over the grid, PI is obtained and represented as a matrix of pixel values. The higher values of a PI imply high-persistence points of the corresponding PD. The example of a PD and its PI are depicted in Figure 4.2. Even if TDA can provide complementary information to improve the performance, since extracting PIs by TDA requires large memory and time consumption (Som *et al.* (2020)), it is difficult to implement the method on small devices with limited computational resources. To solve this issue, I propose a method in knowledge distillation. I utilize features of topological knowledge and time series data to distill a smaller model that generates good performance as a larger model.



(a) Persistence Diagram (b) Persistence Image

Figure 4.2: PD and Its Corresponding PI. In PD, Higher Life-time Appears Brighter.

4.2.2 Knowledge Distillation

Knowledge distillation is one of promising techniques to train a small model in supervision of a large model. KD was firstly explored by Buciluǎ *et al.* (Buciluǎ *et al.* (2006)) and more developed by Hinton *et al.* (Hinton *et al.* (2015)). Soft labels having richer information than hard labels (labeled data), outputs of a teacher network, are used in KD. Soft label enables a student network to easily mimic the softened class scores of the teacher trained with hard labels alone. For traditional KD, a student is trained with the loss function as follows:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\mathcal{CE}} + \lambda\mathcal{L}_{\mathcal{KD}}, \tag{4.1}$$

where $\mathcal{L}_{\mathcal{CE}}$ is the standard cross entropy loss, $\mathcal{L}_{\mathcal{KD}}$ is KD loss, and λ is a hyperparameter; $0 < \lambda < 1$. The error between the output of the softmax layer for a student network and the ground-truth label is penalized by the cross entropy loss:

$$\mathcal{L}_{\mathcal{CE}} = \mathcal{Q}(\sigma(l_S), y), \tag{4.2}$$

where $\mathcal{Q}(\cdot)$ is a cross entropy loss function, $\sigma(\cdot)$ is a softmax function, l_S is the logits of a student, and y is a ground truth label. The outputs of student and teacher are matched by KL-divergence loss:

$$\mathcal{L}_{\mathcal{KD}} = \tau^2 KL(p_T, p_S), \tag{4.3}$$

where, $p_T = \sigma(l_T/\tau)$ is a softened output of a teacher network, $p_S = \sigma(l_S/\tau)$ is a softened output of a student, and τ is a hyperparameter; $\tau > 1$. The standard KD is to use a fully trained teacher and student networks. Recent studies show the effectiveness of early stopping for KD (ESKD), which utilizes early stopped model of a teacher to produce a better student than the standard knowledge distillation (Full KD) (Cho and Hariharan (2019)). For the best performance, ESKD is adopted to this chapter, improving the efficacy of KD (Cho and Hariharan (2019)).

To transfer better knowledge from a teacher network, feature-based distillation using intermediate layers has been proposed (Gou *et al.* (2021); Zagoruyko and Kmodakis (2017); Tung and Mori (2019)). Zagoruyko *et al.* (Zagoruyko and Kmodakis (2017)) suggest attention transfer (AT), which uses intermediate layers to extract a map by a sum of squared attention mapping function. Tung *et al.* (Tung and Mori (2019)) utilizes similarity between a mini-batch of samples from a teacher, which must be matched to those from a student. The activation maps of the teacher and student have the same dimension size, which is determined by size of the mini-batch. In details, the activation map $G' \in \mathbb{R}^{b \times b}$ is produced as follows:

$$G' = A \cdot A^{\top}; A \in \mathbb{R}^{b \times chw}, \tag{4.4}$$

where A is reshaped features from an intermediate layer of a model, b is the size of a mini-batch, c is the number of output channels, and h and w are the height and width of the output, respectively. These methods are popularly used to improve the performance, however, they generally deal with uni-modal problems with a single teacher. On the other hand, using of multiple teachers to transfer more information has been investigated (Gou *et al.* (2021); Liu *et al.* (2020); Zhang *et al.* (2022)). Multiple teachers can provide more useful knowledge to generate a better student. Because different teachers can provide diverse knowledge, more richer information can be transferred to a student. Knowledge from teachers can be utilized individually or integrated to train a student. However, a data sample or label utilized for training a teacher cannot always be used to train/test a student (Gou *et al.* (2021)). Also, leveraging different modalities in KD increases the knowledge gap between a teacher and student, which is a factor in performance degradation (Gou *et al.* (2021)). To resolve the problem and capture the superior knowledge, I develop a framework in KD to use topological features and two teachers for providing richer information and

training a student model that does not use PIs from TDA as an input. The details of the proposed method is explained in section 4.3.

4.2.3 Simulated Annealing

Simulated annealing was first introduced by Kirkpatrick *et al.* (Kirkpatrick *et al.* (1983)) and has been used to solve optimization problems in various applications (Yang (2020)). Recently, it was applied to solve KD related problems. Born-again multitask network (BAM) (Clark *et al.* (2019)) uses a few single-task teachers to generate a multi-task student. A dynamic weighted loss for the outputs of a teacher and ground truth are used to train a student. In the early epochs of training, the student model is mostly trained by the teacher, but later, it is mostly trained by hard labels. Annealing KD (Jafari *et al.* (2021)) presented two stages to reduce the capacity gap between the outputs of a teacher and student. In the first stage, a temperature of KD decreases as the epoch grows while the logits of a teacher and student are matched in a regression task. In the second stage, the student is fine-tuned with hard labels by cross entropy loss. Different from existing annealing methods (Clark *et al.* (2019); Jafari *et al.* (2021)), I propose a strategy of using two teachers with KD to facilitate fast saturation and reduce the knowledge gap. For the proposed method, two teachers are trained with different types of data – time series and persistence image data – and their student is trained with the raw time series data only. So, the statistical features of two teachers are different, and their distillation effects on a student are not the same. To consider the different properties of teachers and the student in distillation, I apply an annealing strategy in KD, which reduces the search space for fast saturation and helps to mitigate a knowledge gap issue by leveraging the weights of a model trained from scratch. Our method is described in the next section.

4.3 Proposed Method

For the proposed method, two teachers learned with different data are used to train a student. Firstly, to leverage topological features, I extract PIs from PDs of time series data using TDA. Two teacher models are trained with time series data and extracted PIs, respectively. Secondly, orthogonal features from fused correlation maps of teachers are used for distillation, considering differently activated features from teachers. In the third step, I apply an annealing strategy for knowledge distillation to optimize the weight of the student model, taking into account the time series properties inherent in the model. Finally, a student model preserving topological features is distilled. The details of the proposed method are explained in the following section.

4.3.1 Extracting Persistence Image

Topological features provide complementary information to improve the performance in machine learning (Gholizadeh and Zadrozny (2018); Zeng *et al.* (2021); Som *et al.* (2020)). To leverage topological features, I first extract PIs to train a model. I use Scikit-TDA python library (Saul and Tralie (2019)) and the Ripser package for producing PDs, referring to a previous study (Som *et al.* (2020)). PDs of level-set filtration for time series signals are calculated by the library. Scalar field topology presents a summary for different peaks in the signal. The PD for each channel of a sample is computed. And then, PIs are extracted from PDs based on the birth-time vs. lifetime information. I set the matrix size of the PIs as $b \times b$. The dimension size of one PI is $b \times b \times c$, where c is the number of channels for a sample. Secondly, I train a model on the extracted PIs in supervised learning. The model is used as the pre-trained model as a teacher, transferring topological features to a student model.

4.3.2 KD with Multiple Teachers

In test-time, generating PIs requires a large computational burden. To this end, I adopt KD to distill a student model using only time series data as an input and learning topological features from a teacher.

Distillation with Logits of Different Teachers

For the proposed method, since the knowledge from two teachers is transferred separately, additional processing for concatenation and hidden layers is not necessarily required. To utilize features from two teachers, KD loss can be written as:

$$\mathcal{L}_{\mathcal{KD}_m} = \tau^2 \left(\alpha KL(p_{T_1}, p_S) + (1-\alpha) KL(p_{T_2}, p_S) \right), \tag{4.5}$$

where α is a hyperparameter to balance the losses from different teachers, and p_{T_1} and p_{T_2} are softened outputs of teachers trained with time series data and PIs, respectively.

Similarity of Different Teachers

For better distillation, I use features from intermediate layers of teachers. However, the architectures of the teachers and student are different, and their data used for training are also different modalities. Using methods similar those proposed in Tung *et al.* (Tung and Mori (2019)), I extract activation similarity matrices $G' \in \mathbb{R}^{b \times b}$ to use activated features with the same dimension size from the two teachers and student, as explained in Equation (4.4). The pattern of the activation map is highly related to the same or a different class. In details, two inputs in the same category generate the similar activation maps from a teacher, which is a beneficial to guide a student to acquire the knowledge of the teacher.

However, because of the information gap from different modalities, difficulties still exist to transferring the each different knowledge (Gou *et al.* (2021)). As shown in

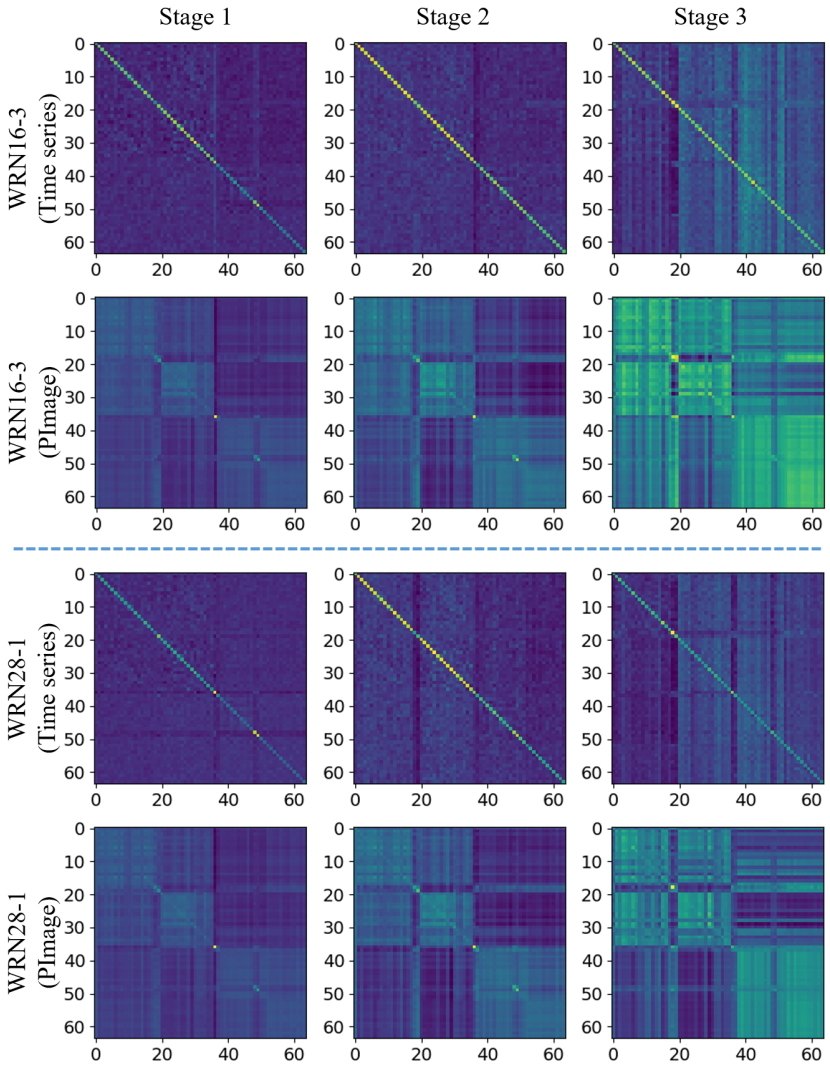


Figure 4.3: Examples of Activation Similarity Maps G' Produced by a Layer for the Indicated Stage of the Network for a Batch on GENEActiv. High Similarities for Samples of the Batch Are Represented with High Values. The Blockwise Pattern Is More Distinctive for WRN16-3 Networks, Implying the Higher Capacity of This Network Can Well Capture the Semantics of the Dataset.

Figure 4.3, two models trained with different data generate dissimilar activations. These differences from multimodality make difficulties in interpreting and fusing the content, which may mislead the student (Kwon *et al.* (2020); Gou *et al.* (2021)). To

solve this issue, I create a map from two teachers by merging the activation maps with the weight value of α parameter as follows:

$$G_T^{(l)} = \alpha G_{T_1}^{\prime(l^{T_1})} + (1-\alpha) G_{T_2}^{\prime(l^{T_2})}, \tag{4.6}$$

where $G_T^{(l)} \in \mathbb{R}^{b \times b}$ is the generated map from the activation maps of a layer pair (l^{T_1} and l^{T_2}) of two teachers G'_{T_1} and G'_{T_2}. By merging the maps, the similarities between two teachers are more highlighted.

Extracting and Transferring Orthogonal Features

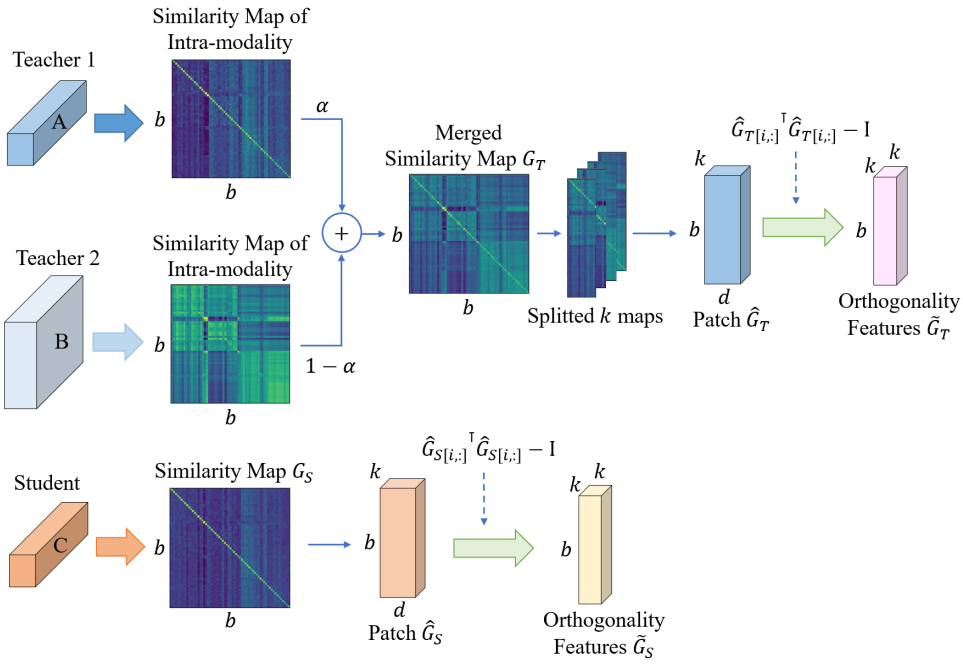


Figure 4.4: Framework of Extracting Orthogonal Features. A and B Denote Mini-batch Features at a Layer of Teacher1 and Teacher2, Respectively. C Denotes Mini-batch Features at a Layer of Student.

If features from multiple teachers are ideally correlated, the errors of one teacher would not essentially affect the other one (Park *et al.* (2020)). Since features from

different teachers are merged and the data used for training a student is different from that of the teachers, it is difficult to guarantee that the teachers and student are perfectly correlated, so the merged map from teachers may not always be good for distillation. In previous studies (Wang *et al.* (2020b)), orthogonality properties improve better feature explanation and lead to provide various desirable features, which enables a model to easily learn more diverse and expressive features. Given the insight, to capture the better explanative information accounting for modality gap, I design new knowledge reflecting orthogonal properties by transforming the merged map into several patches to produce more attentive feature relationship. The overview of extracting orthogonal features is described in Figure 4.4. An input-patch-matrix $\widehat{G} \in \mathbb{R}^{bd \times k}$ can be constructed by unrolling the $G/\|G\|_2$, the normalized G, into k columns of the matrix, where k is the number of partitions and d is the size of each partition for b. By using the computed patch-matrix, new knowledge encoding feature relationships based on orthogonal properties is defined as follows:

$$\widetilde{G}_{[i,:]} = \widehat{G}_{[i,:]}^{\top} \widehat{G}_{[i,:]} - \mathbf{I}, \tag{4.7}$$

where $\widetilde{G}_{[i,:]} \in \mathbb{R}^{k \times k}$ represents $k \times k$ knowledge patches for ith element of b, involving orthogonality properties, and $\mathbf{I} \in k \times k$ is identity matrices. From the merged map G_T and a map for the student G_S, $\widetilde{G_T}$ and $\widetilde{G_S}$ can be generated, respectively. Finally, the knowledge reflecting feature relationships from teachers are transferred to the student by minimizing the difference between two maps of each corresponding layer:

$$\mathcal{L}_{Oth} = \frac{1}{|L|} \sum_{(l,l^S) \in L} \left\| \widetilde{G_T^{(l)}} - \widetilde{G_S^{(l^S)}} \right\|_F^2, \tag{4.8}$$

where L collects the layer pairs (l and l^S), and $\|\cdot\|_F$ is the Frobenius norm (Tung and Mori (2019)). In this way, the student is encouraged to get the similar features to the merged teacher. Therefore, the student can preserve topological as well as time series

features, which uses the raw time series data only as an input. The overall learning objective of the proposed method can be written as:

$$\mathcal{L}_{TP} = (1 - \lambda)\mathcal{L}_{\mathcal{CE}} + \lambda\mathcal{L}_{\mathcal{KD}_m} + \beta\mathcal{L}_{Oth}, \tag{4.9}$$

where β is a hyperparameter to control the effect of loss $\mathcal{L}_{Oth}$.

4.3.3 Annealing Strategy for Multiple Teachers

Because teachers and student are trained with different data, models generate different statistical properties of features. Their architectures are even different, which produces more statistical gaps between features from models and difficulties in training a student (Gou *et al.* (2021); Jafari *et al.* (2021)). To reduce the effects of the knowledge gap, I apply an annealing strategy in KD for the proposed method. Before training a student, I train a small model from scratch with time series data, where the model has the same architecture as the student. When weight values are initialized to train the student, the values are determined by the pre-trained model, instead of randomly chosen values. In this way, the knowledge gap between teachers and student is mitigated and the search space for optimization is reduced. Also, this initialization enforces the student to get features that can perform well with time series data while teachers provide their own features.

4.4 Experiments

In this section, I describe datasets used for evaluation and experimental settings. I evaluate the proposed method with various teacher-student combinations on wearable sensor data. I investigate the sensitivity of the proposed distillation with various hyperparameters (α, β, and k). And, I explore the effectiveness of TPKD with

visualization of feature maps, feature similarity analysis, and generalizability analysis. Also, I measure computational time with different methods.

4.4.1 Data Description and Experimental Settings

Data Description

I evaluate the proposed method with wearable sensor data on GENEActiv and PAMAP2 datasets.

GENEActiv. GENEActiv (Wang *et al.* (2016)) is wearable sensor based activity dataset, collected with GENEActiv sensor which is a light-weight, waterproof, and wrist-worn tri-axial accelerometer with sampling frequency of 100 Hz. In this experiment, referring to the previous study (Jeon *et al.* (2022b)), I use 14 daily activities such as walking, sitting, and standing. Each class has over 900 data samples. The number of subjects for training and testing are over 130 and 43, respectively. I use full-non-overlapping window size of 500 time-steps (5 seconds) data. The number of samples for training and testing are approximately 16k and 6k, respectively.

PAMAP2. PAMAP2 dataset (Reiss and Stricker (2012)) consists of 18 physical activities (12 daily and 6 optional activities) for 9 subjects, obtained by measurements of heart rate, temperature, accelerometers, gyroscopes, and magnetometers with 100Hz of sampling frequency. The sensors were placed on hands, chest, and ankles of the subject. In experiments on this dataset, I use 12 daily activities with 40 channels recorded from the heart rate and 4 IMUs, where activities are lying, sitting, standing, walking, etc. To compare with previous methods, the recordings are downsampled to 33.3Hz. I evaluate methods with leave-one-subject-out combination. There is missing data for some subjects and the dataset has non-uniform distribution. I use 100 time-steps (3 seconds) of a sliding window for a sample and 22 time-steps or

660 ms of step size for segmenting the sequences, which allows semi-non-overlapping sliding windows with 78% overlapping (Reiss and Stricker (2012)).

Experimental Settings

In extracting PIs, for GENEActiv, the parameter for the Gaussian function in PD is 0.25 and the values for birth-time range of PI are set as [-10, 10], as do the previous study (Som *et al.* (2020)). For PAMAP2, Gaussian parameter and the birth-time range are 0.015 and [-1, 1], respectively. Each calculated PI is normalized by its maximum value. To train network models, I set the total epochs as 200 using SGD with momentum of 0.9, the batch size as 64, and a weight decay as 1×10^{-4}. To train a model with time series data on both datasets, the initial learning rate lr is 0.05 which decreases by 0.2 at 10 epochs and drops down by 0.1 every $[\frac{t}{3}]$ where t is the total number of epochs. For training a model with image data on GENEActiv, the initial learning rate lr is set to 0.1 and decreases by 0.5 at 10 epochs and drops down by 0.2 at 40, 80, 120, and 160 epochs. For PAMAP2 with image data, the initial learning rate lr is set as 0.1 that drops down by 0.2 at 40, 80, 120, and 160 epochs. For constructing teacher and student models, I use WideResNet (WRN) (Zagoruyko and Komodakis (2016)) to evaluate the performance of the proposed method, which is popularly used to validate in KD (Cho and Hariharan (2019); Jeon *et al.* (2022b)). The model for training with time series data consists of 1D convolutional layers, on the other hand, the one with image data consists of 2D convolutional layers. I determine τ and λ for GENEActiv as 4 and 0.7, and for PAMAP2 as 4 and 0.99, respectively, as the previous works do (Jeon *et al.* (2022b)). To obtain the best results, I set optimal α as 0.7 for GENEActiv and 0.3 for PAMAP2, respectively. I run 3 times and report with the best averaged accuracy and standard deviation for the following experiments. I perform baseline comparisons with traditional KD

(Hinton *et al.* (2015)), attention transfer (AT) (Zagoruyko and Kmodakis (2017)), and similarity-preserving knowledge distillation (SP) (Tung and Mori (2019)), which are popularly used for distillation. α_{AT} and γ_{SP} are set as 1500 and 1000 for GENEActiv, and 3500 and 700 for PAMAP2, respectively. Also, I compare with multi-teacher based approaches such as AVER (You *et al.* (2017)), EBKD (Kwon *et al.* (2020)), and CA-MKD (Zhang *et al.* (2022)). Since I use different dimensional input data and structured teachers, only the outputs from the last layer (logits) are used for baselines in distillation.

4.4.2 Various Capacity of Teachers

In this section, I explore the proposed method with various capacity of teachers which are trained with time series data and PIs, respectively.

The experimental results on GENEActiv with various teachers are described in Table 4.1. Note, "Time series" and "PImage" denote results of the model trained by KD with Teacher1 trained with time series data and Teacher2 trained with PIs, respectively. "TS", "Base", and "Ann." denote using a teacher trained with time series data, a model trained by two teachers in KD using logits balanced with α, and applying annealing strategy, respectively. "Orth." denotes using orthogonal features in distillation. When TPKD is implemented without orthogonal features, the merged map of teachers and the one of student are matched directly by mean squared error in distillation. The numbers in brackets imply trainable parameters of the model and accuracy, respectively. From the left to right combinations of teachers in the table, (β, k) of TPKD are defined as (900, 4), (700, 2), (700, 4), and (900, 4), respectively. TPKD (TS+PImage with Ann.+orthogonal feature distillation), as shown in the table, achieves the best performing results in all cases. Base models trained with the annealing strategy (Ann.) outperform the results of baselines and the

Table 4.1: Accuracy (%) with Various Knowledge Distillation Methods for Different Capacity of Teachers on GENEActiv.

	Teacher1 (1D CNNs)	WRN16-1 (0.06M, 67.66)	WRN16-3 (0.5M, 68.89)	WRN28-1 (0.1M, 68.63)	WRN28-3 (1.1M, 69.23)
	Teacher2 (2D CNNs)	WRN16-1 (0.2M, 58.64)	WRN16-3 (1.6M, 59.80)	WRN28-1 (0.4M, 59.45)	WRN28-3 (3.3M, 69.69)
	Student (1D CNNs)	WRN16-1 (0.06M, 67.66$\pm$0.45)			
PI	KD	67.83 $\pm$0.17	68.76 $\pm$0.73	68.51 $\pm$0.01	68.46 $\pm$0.28
Time series	KD	69.71 $\pm$0.38	69.50 $\pm$0.10	68.32 $\pm$0.63	68.58 $\pm$0.66
	AT	68.21 $\pm$0.64	69.79 $\pm$0.36	68.09 $\pm$0.24	67.73 $\pm$0.27
	SP	67.20 $\pm$0.36	67.85 $\pm$0.24	68.71 $\pm$0.46	67.39 $\pm$0.49
TS+PImage	AVER	68.99 $\pm$0.76	68.74 $\pm$0.35	68.77 $\pm$0.70	69.02 $\pm$0.50
	EBKD	68.43 $\pm$0.25	69.24 $\pm$0.25	68.45 $\pm$0.73	67.50 $\pm$0.40
	CA-MKD	69.33 $\pm$0.61	69.80 $\pm$0.16	69.61 $\pm$0.57	68.81 $\pm$0.79
	Base	69.09 $\pm$0.37	69.24 $\pm$0.62	69.55 $\pm$0.41	69.42 $\pm$0.58
	Ann.	70.15 $\pm$0.03	70.71 $\pm$0.12	70.44 $\pm$0.10	69.97 $\pm$0.06
	TPKD (w/o Orth.)	70.71 $\pm$0.20	70.93 $\pm$0.26	70.71 $\pm$0.14	70.12 $\pm$0.21
	TPKD (w/ Orth.)	**71.05** $\pm$0.13	**71.10** $\pm$0.11	**70.97** $\pm$0.12	**70.50** $\pm$0.15

Table 4.2: Accuracy (%) for Related Methods on GENEActiv with 7 Classes.

	Method	Window length	
		1000	500
Time series	SVM (Cortes and Vapnik (1995))	86.29	85.86
	Choi *et al.* (Choi *et al.* (2018))	89.43	87.86
	WRN16-1	89.29±0.32	86.83±0.15
	WRN16-3	89.53±0.15	87.95±0.25
	WRN16-8	89.31±0.21	87.29±0.17
	ESKD (WRN16-3)	89.88±0.07	88.16±0.15
	ESKD (WRN16-8)	89.58±0.13	87.47±0.11
	Full KD (WRN16-3)	89.84±0.21	87.05±0.19
	Full KD (WRN16-8)	89.36±0.06	86.38±0.06
	AT (WRN16-1)	90.10±0.49	87.25±0.22
	AT (WRN16-3)	90.32±0.09	87.60±0.22
	SP (WRN16-1)	87.08±0.56	87.65±0.11
	SP (WRN16-3)	88.47±0.19	87.69±0.18
TS+PImage	AVER (WRN16-1)	90.01±0.46	87.53±0.16
	AVER (WRN16-3)	90.06±0.33	87.05±0.37
	EBKD (WRN16-1)	90.35±0.12	87.51±0.41
	EBKD (WRN16-3)	89.82±0.14	87.66±0.28
	CA-MKD (WRN16-1)	90.01±0.28	87.14±0.25
	CA-MKD (WRN16-3)	90.13±0.34	88.04±0.26
	Ann. (WRN16-1)	90.44±0.16	88.18±0.12
	Ann. (WRN16-3)	90.71±0.15	88.26±0.24
	TPKD (w/ Orth.) (WRN16-1)	**90.93**±0.11	**88.83**±0.22
	TPKD (w/ Orth.) (WRN16-3)	90.83±0.09	88.60±0.25

basic model (Base), indicating that the strategy aids in performance improvement. Also, larger model does not guarantee to generate a better student, corroborating previous observations (Cho and Hariharan (2019)). To compare with different sample

Table 4.3: Accuracy (%) with Various Knowledge Distillation Methods for Different Capacity of Teachers on PAMAP2.

Teacher1 (1D CNNs)		WRN16-1 (0.06M, 85.27)	WRN16-3 (0.5M, 85.80)	WRN28-1 (0.1M, 84.81)	WRN28-3 (1.1M, 84.46)
Teacher2 (2D CNNs)		WRN16-1 (0.2M, 86.93)	WRN16-3 (1.6M, 87.23)	WRN28-1 (0.4M, 87.45)	WRN28-3 (3.3M, 87.88)
Student (1D CNNs)		WRN16-1 (0.06M, 82.99±2.50)			
PI	KD	85.04 ±2.58	86.68 ±2.19	85.08 ±2.44	85.39 ±2.35
TS	KD	85.96 ±2.19	86.50 ±2.21	84.92 ±2.45	86.26 ±2.40
TS+PImage	Base	85.91 ±2.32	86.18 ±2.37	85.54 ±2.26	86.04 ±2.24
	Ann.	86.09 ±2.33	87.12 ±2.26	85.89 ±2.26	86.33 ±2.30
	TPKD (w/o Orth.)	87.26 ±2.09	88.00 ±2.21	86.47 ±2.26	86.92 ±2.27
	TPKD (w/ Orth.)	**87.67** ±2.01	**88.45** ±2.10	**86.86** ±2.07	**87.40** ±2.13

window lengths and more previous studies, I evaluate the methods with 7 classes of GENEActiv dataset, as do the previous study (Jeon *et al.* (2022b); Choi *et al.* (2018)). WRN16-1 (1D CNNs) student is used. The brackets denote the teacher models. (β, k) of TPKD are set as (1100, 4) for WRN16-1 teachers with both window lengths and WRN16-3 teachers with window length of 500, and (500, 8) for WRN16-3 teachers with window length of 1000, respectively. As summarized in Table 4.2, results of TPKD (w/ Orth.) with WRN16-1 teachers show the best in both cases.

Table 4.4: Accuracy (%) for Related Methods on PAMAP2.

	Method	Accuracy (%)
Time series	Chen and Xue (2015)	83.06
	Ha *et al.* (2015)	73.79
	Ha and Choi (2016)	74.21
	Kwapisz *et al.* (2011)	71.27
	Catal *et al.* (2015)	85.25
	Kim *et al.* (2012)	81.57
	WRN16-1	82.81±2.51
	WRN16-3	84.18±2.28
	WRN16-8	83.39±2.26
	ESKD (WRN16-3)	86.38±2.25
	ESKD (WRN16-8)	85.11±2.46
	Full KD (WRN16-3)	84.31±2.24
	Full KD (WRN16-8)	83.70±2.52
	AT (WRN16-1)	83.79±2.40
	AT (WRN16-3)	84.44±2.22
	SP (WRN16-1)	84.31±2.38
	SP (WRN16-3)	84.89±2.10
TS+PImage	AVER (WRN16-1)	85.82±2.16
	AVER (WRN16-3)	86.00±2.45
	EBKD (WRN16-1)	85.58±2.31
	EBKD (WRN16-3)	85.62±2.37
	CA-MKD (WRN16-1)	84.06±2.50
	CA-MKD (WRN16-3)	85.02±2.64
	Ann. (WRN16-1)	86.09±2.33
	Ann. (WRN16-3)	87.12±2.26
	TPKD (w/ Orth.) (WRN16-1)	87.67±2.01
	TPKD (w/ Orth.) (WRN16-3)	**88.45**±2.10

When an annealing strategy is applied, smaller teachers distill better students. Since one of teachers (WRN16-1) has the same structure of the student (WRN16-1), the knowledge gap is not much different than the larger teachers (WRN16-3). The results with various capacity of teachers on PAMAP2 are described in Table 4.3. β and k of TPKD are defined as 200 and 4, respectively. TPKD (w/ Orth.) shows the best in all cases. As described in Table 4.4, TPKD outperforms the previous methods. Therefore, TPKD allows model compression and improves accuracy across datasets.

4.4.3 Various Combinations of Teachers

To understand the effect from different teacher architectures, various combinations of two teachers are used, considering different channel and depth of WRN. Results on GENEActiv and PAMAP2 are described in Table 4.5 and 4.6, respectively. (β, k) on GENEActiv for each combination is indicated in Table 4.5. (β, k) on PAMAP2 is set as (200, 4). β for TPKD without using orthogonal features is set as 700 and 200 on GENEActiv and PAMAP2, respectively.

As shown in Table 4.5, in most cases, TPKD (w/ Orth.) shows the best performance. When the capacity of Teacher1 is high, the result gap between baselines and TPKD tends to be small, where TPKD still performs better. When both teachers are small (e.g. WRN28-1 Teacher1 and WRN16-1 Teacher2), the student by TPKD performs better than the one from the other combinations of teachers. Also, when width of teachers is the same as the student, the proposed method shows better performance than other combinations of teachers.

As described in Table 4.6, TPKD shows better performance than Ann. (applying annealing strategy only) in all cases. This result also shows when the capacity of Teacher1 is high, the result gap between baselines and TPKD tends to be small. This is because a large teacher creates more knowledge gap which makes challenges in

Table 4.5: Accuracy (%) with Various Knowledge Distillation Methods for Different Structure of Teachers on GENEActiv.

Method	Architecture Difference											
	Depth				Width				Depth+Width			
Teacher1 (1D CNNs)	WRN 16-1 (0.06M, 67.66)	WRN 16-1 (0.06M, 67.66)	WRN 28-1 (0.1M, 68.63)	WRN 40-1 (0.2M, 69.05)	WRN 16-1 (0.06M, 67.66)	WRN 16-3 (0.5M, 68.89)	WRN 28-1 (0.1M, 68.63)	WRN 28-3 (1.1M, 69.23)	WRN 28-1 (0.1M, 68.63)	WRN 28-3 (1.1M, 69.23)	WRN 40-1 (0.2M, 69.05)	WRN 16-1 (0.06M, 67.66)
Teacher2 (2D CNNs)	WRN 28-1 (0.1M, 59.45)	WRN 40-1 (0.6M, 59.67)	WRN 16-1 (0.2M, 58.64)	WRN 16-1 (0.2M, 58.64)	WRN 16-3 (1.6M, 59.80)	WRN 16-1 (0.2M, 58.64)	WRN 28-3 (3.3M, 59.69)	WRN 28-1 (0.4M, 59.45)	WRN 16-3 (1.6M, 59.80)	WRN 40-1 (0.6M, 59.67)	WRN 28-3 (3.3M, 59.69)	WRN 28-3 (3.3M, 59.69)
Student (1D CNNs)	WRN16-1 (0.06M, 67.66$_{\pm 0.45}$)											
Base	68.71 ± 0.36	68.41 ± 0.27	67.89 ± 0.27	68.33 ± 0.17	68.77 ± 0.43	68.92 ± 0.79	68.26 ± 0.13	69.09 ± 0.59	68.04 ± 0.24	68.29 ± 0.27	68.90 ± 0.50	68.15 ± 0.23
Ann.	69.95 ± 0.05	69.86 ± 0.07	70.34 ± 0.14	70.56 ± 0.04	69.68 ± 0.14	71.06 ± 0.02	70.28 ± 0.08	69.95 ± 0.07	70.28 ± 0.13	69.87 ± 0.23	70.49 ± 0.05	69.65 ± 0.04
TPKD (w/o Orth.)	70.39 ± 0.12	70.47 ± 0.40	71.01 ± 0.04	71.36 ± 0.06	69.82 ± 0.23	71.11 ± 0.18	70.53 ± 0.26	70.31 ± 0.15	**70.55** ± 0.28	70.57 ± 0.18	70.55 ± 0.22	70.68 ± 0.10
TPKD (w/ Orth.)	**70.67** ± 0.33	**70.76** ± 0.22	**71.74** ± 0.07	**71.40** ± 0.05	**70.03** ± 0.14	**71.25** ± 0.18	**71.08** ± 0.21	**70.35** ± 0.09	70.42 ± 0.21	**70.65** ± 0.24	**71.04** ± 0.29	**71.00** ± 0.33
(β, k)	(900, 4)	(900, 4)	(700, 4)	(900, 4)	(700, 4)	(700, 2)	(900, 4)	(700, 4)	(1100, 4)	(900, 4)	(700, 4)	(900, 4)

distillation. There is some cases that baselines produce less improvement with large Teacher1, compared to using small one. Even if the performance is affected from the knowledge gap, TPKD alleviates the negative effects in distillation, which outperforms the all baselines, and even generates a better student than its teachers. Also, the results corroborate that large teachers does not always distill a better student (Cho and Hariharan (2019)).

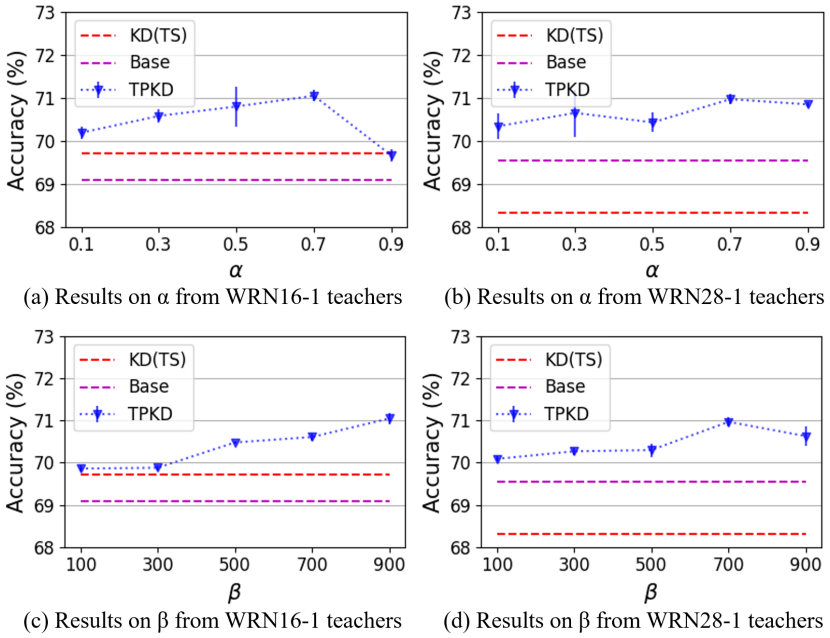
Table 4.6: Accuracy (%) with Various Knowledge Distillation Methods for Different Structure of Teachers on PAMAP2.

Method	Architecture Difference					
	Depth		Width	Depth+Width		
Teacher1 (1D CNNs)	WRN 28-1 (0.1M, 84.81)	WRN 16-1 (0.06M, 85.27)	WRN 28-3 (1.1M, 84.46)	WRN 16-3 (0.5M, 85.80)	WRN 16-1 (0.06M, 85.27)	WRN 28-3 (1.1M, 84.46)
Teacher2 (2D CNNs)	WRN 16-1 (0.2M, 86.93)	WRN 28-1 (0.4M, 87.45)	WRN 28-1 (0.4M, 87.45)	WRN 28-1 (0.4M, 87.45)	WRN 28-3 (3.3M, 87.88)	WRN 16-1 (0.2M, 86.93)
Student (1D CNNs)	WRN16-1 (0.06M, 82.99$_{\pm 2.50}$)					
Ann.	85.97 $\pm$2.33	85.33 $\pm$2.22	85.59 $\pm$2.28	85.82 $\pm$2.26	85.94 $\pm$2.31	85.86 $\pm$2.42
TPKD (w/ Orth.)	**86.10** $\pm$2.30	**87.26** $\pm$1.96	**87.94** $\pm$2.08	**87.82** $\pm$2.07	**87.02** $\pm$1.98	**86.97** $\pm$2.26

4.4.4 Ablations and Sensitivity Analysis

In this section, I investigate the effects of hyperparameters (α, β, and k) on TPKD (with orthogonal features). And, feature maps from intermediate layers of trained students are visualized to better understand the performance of TPKD. Also, I analyze feature similarities and generalizability of models. Additionally, to figure out the robustness of TPKD, I explore the proposed method under noisy testing data.

Effect of Distillation Hyperparameters on TPKD



(a) Results on α from WRN16-1 teachers (b) Results on α from WRN28-1 teachers

(c) Results on β from WRN16-1 teachers (d) Results on β from WRN28-1 teachers

Figure 4.5: Sensitivity to α and β of the Proposed Method for WRN16-1 Students on GENEActiv.

The results of students (WRN16-1), trained with two different teachers by using various α and β ($k = 4$), are illustrated in Figure 4.5. For (a) and (b), β is set as the previous section. KD is the result of a student trained with time series data. Most results from TPKD outperform baselines. The results show their best when α is 0.7. On the other hand, for PAMAP2, their best are shown with $\alpha = 0.3$. Since GENEActiv has a larger window size and much lower number of channels than PAMAP2, utilizing features from time series data may help improvements more than PIs. On the other hand, because PAMAP2 has a much smaller window size but more channels, using projected image data from PIs may provide more useful information than raw time series data. The results with various β are shown in (c) and (d) of Figure 4.5. α is set as 0.7. All results from TPKD outperform baselines. The best

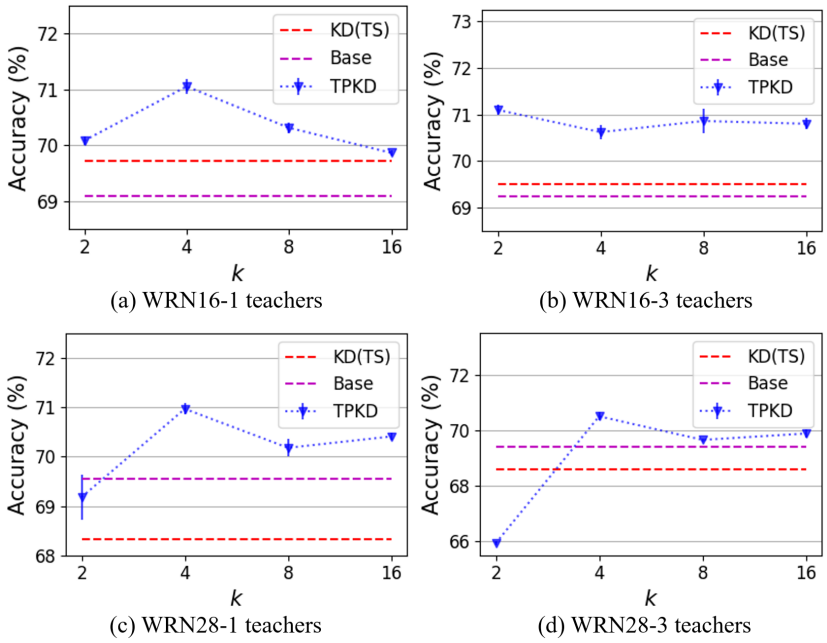


Figure 4.6: Sensitivity to k of the Proposed Method for WRN16-1 Students on GENEAtiv.

results are presented with $\beta = 900$ for (c) and $\beta = 700$ for (b). The majority of the best results in the previous section had beta values of 700 or higher. For PAMAP2, with the same structured teachers, smaller number of β (200) shows the best. When the window size is large and the number of channels is small, orthogonal features can have more influence on classification with $\beta \geq 700$. The results of WRN16-1 students with various k are illustrated in Figure 4.6. α is 0.3 and β is set as the same for each combination in section 4.4.2. Most k cases outperform baselines and best result is yielded with $k = 4$. When teacher models have different width of networks to their student, $k = 2$ shows lower accuracy than baselines, whereas $k \geq 4$ shows higher one. And, as described in section 4.4.2 and 4.4.3, most cases on GENEActiv and PAMAP2 perform best when $k = 4$. Based on these results, setting appropriate hyperparameters has to be considered to generate the best performance.

Visualization of Feature Maps

To see more details of activations, I visualize the maps of the teachers (WRN16-3) and student (WRN16-1), representing similarity with high values for inputs. "Teacher1" and "Teacher2" denote teachers trained with time series data and PIs, respectively. KD is the result of a student trained with time series data. Student is the result of a model trained from scratch. As illustrated in Figure 4.7, in all cases. the produced maps in stage 3 have more distinctive patterns than the ones from stage 1 and 2. The maps of two teachers are very different, and the merged one and student are dissimilar, indicating the knowledge gap between them. Some columns of the map from models trained with time series data are highlighted. The blockwise patterns are more shown from models trained with PIs. Intuitively, the pattern of the map from Teacher1 is more monotonous than the one from Teacher2. And, the diagonal points of the map trained with time series only are more highlighted prominently. The merged map contains characteristics of both Teacher1 and Teacher2. A student trained with TPKD generates maps closer to those of the merged maps from teachers. Also, the maps from TPKD represent blockwise highlighted features, which verifies that the student preserves topological features by the proposed method.

More results from different combinations of teachers with a layer for stage 3 of the network are illustrated in Figure 4.8. Compared to baselines, maps of students by TPKD are more similar to the merged ones which contain both topological and time series features. Therefore, TPKD encourages a student to well obtain both features of topological and time series data while reducing the knowledge gap.

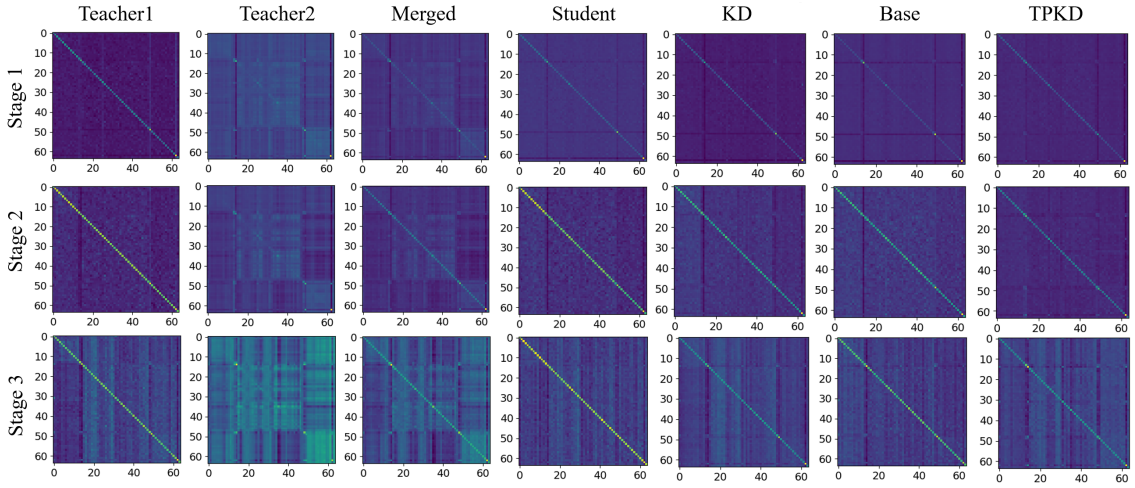


Figure 4.7: Activation Similarity Maps Produced by a Layer for the Indicated Stage of the Network for a Batch on GENEActiv. High Similarities for Samples of the Batch Are Represented with High Values.

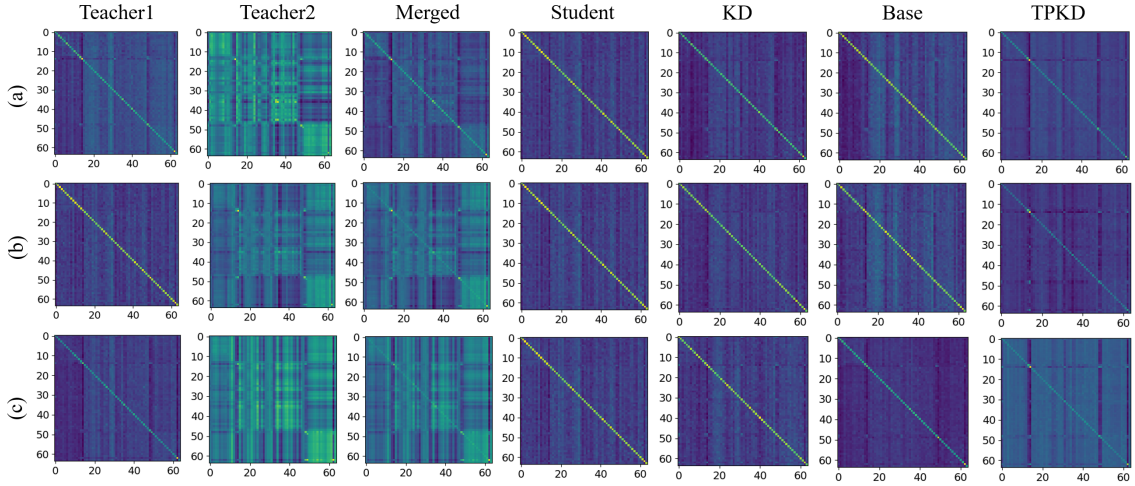


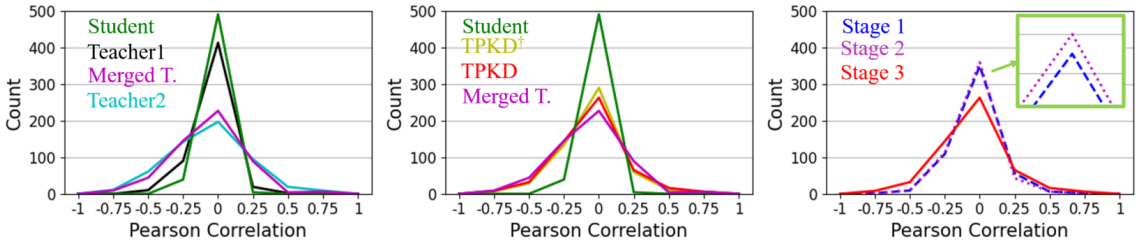
Figure 4.8: Activation Similarity Maps Produced by a Layer for the Stage 3 of the Network for a Batch on GENEActiv. From (a) to (c), (Teacher1, Teacher2) Are (WRN28-1, WRN16-1), (WRN16-1, WRN16-3), and (WRN40-1, WRN28-3), Respectively. High Similarities for Samples of the Batch Are Represented with High Values.

Analysis of Orthogonality in Distillation

To analyze the effects of leveraging orthogonal features, I measure feature similarity quantitatively with Pearson correlation coefficient on activation maps of models from various knowledge distillation methods. Also, I analyze the generalizability of student models for the different methods.

Feature Similarity. I calculate Pearson correlation coefficient on activation similarity maps from intermediate layers. Four patches $\widehat{G} \in \mathbb{R}^{bd \times k} = [\widehat{G}_1, \widehat{G}_2, \cdots, \widehat{G}_k]$ $(k = 4)$ from students trained with WRN28-3 teachers are used to generate feature similarity plots. All pair combinations of the patches $[(\widehat{G}_1, \widehat{G}_2), (\widehat{G}_1, \widehat{G}_3), \cdots, (\widehat{G}_{k-1}, \widehat{G}_k)]$ are considered for the coefficient. As depicted in Figure 4.9 (a), the similarities between the two teachers are very different. The model trained from scratch with time series alone shows high values in 0 of the correlation coefficient. This implies that most of the patches from the models are decorrelated. On the other hand, the patches of Teacher2 are more correlated and much different from Student and Teacher1. The result from the merged patches (Merged T.) for teachers shows intermediate results between two teachers, but closer to Teacher2. These show there is a statistical gap between the teachers and student. In the figure (b), TPKD (with orthogonal features) shows a more similar result to Merged T. than the one without orthogonal features which is a direct map matching method. By orthogonal features in distillation, the student can learn more attentive features and perform more teacher-like tasks. Also, the student trained by TPKD is implemented with time series data only as an input, but it produces similar features to Merged T. Thus, TPKD distills a student preserving topological features while reducing the knowledge difference between teachers and a student. As described in (c) of the figure, the patches from 3rd stage of the network are more correlated with each other

than the other stages. And, the features from each stage have different statistical characteristics. So, transferring features with different stages can help to improve performance.



(a) Feature Similarities for Teachers (b) Feature Similarities for methods (c) Feature Similarities for TPKD

Figure 4.9: Feature Similarities for Various Knowledge Distillation Methods on GENEActiv. Teachers Are WRN28-3 and Students Are WRN16-1 (1D CNNs). Merged T. Denote the Merged Features from Teachers. (a) and (b) Are Results from 3rd Stage of the Networks. † Denotes Without Orthogonal Features.

Model Reliability. To study the generalizability and regularization effects, I measured expected calibration error (ECE) (Guo *et al.* (2017)) and negative log likelihood (NLL) (Guo *et al.* (2017)). ECE is to measure calibration, representing the reliability of the model. The probabilistic quality of a model can be measured by NLL. I used students trained by teachers of WRN16-3 and WRN28-1. In Table 4.7, ECE and NLL with various methods on GENEActiv are described. The results of Base outperform KD and Student (learning from scratch). This implies that leveraging topological features improves performance in reliability. TPKD (with orthogonal features) generates the lowest ECE and NLL in both cases. The results on PAMAP2 are shown in Table 4.8. In both cases, Base performs better than KD and the model learned from scratch. TPKD outperforms all baselines, and using orthogonal features shows the best results. This implies that utilizing orthogonal features in distillation aids in generating a better model, not only for accuracy but also for reliability.

Table 4.7: ECE (%) and NLL for Various Knowledge Distillation Methods on GENEActiv. Teachers are WRN16-3 and WRN28-1. Students are WRN16-1 (1D CNNs).

Method	WRN16-3		WRN28-1	
	ECE	NLL	ECE	NLL
Student	3.548	2.067	3.548	2.067
KD	3.200	1.520	3.064	1.512
Base	2.998	1.142	3.009	1.271
TPKD (w/o Orth.)	2.728	1.128	2.634	1.114
TPKD (w/ Orth.)	**2.637**	**1.103**	**2.616**	**1.068**

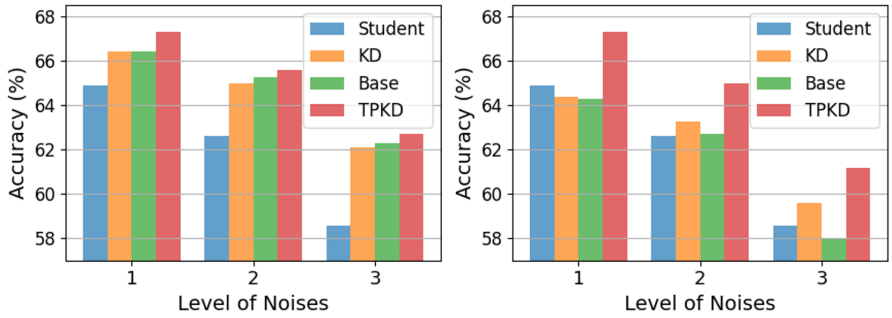
Table 4.8: ECE (%) and NLL for Various Knowledge Distillation Methods on PAMAP2. Teachers are WRN16-3 and WRN28-1. Students are WRN16-1 (1D CNNs).

Method	WRN16-3		WRN28-1	
	ECE	NLL	ECE	NLL
Student	2.299	1.287	2.299	1.287
KD	2.183	1.061	2.323	1.329
Base	2.039	0.815	2.130	0.955
TPKD (w/o Orth.)	1.897	0.754	2.075	0.896
TPKD (w/ Orth.)	**1.692**	**0.708**	**1.818**	**0.856**

Analysis of Invariance from Noises

To explore the model ability of invariances from noises, I evaluate the models on a noisy testing set, including continuous missing and Gaussian noises, where the noises reflect the errors commonly encountered in time series (Jeon *et al.* (2022b); Wen *et al.*

(2021a); Wang and Wang (2019)). To consider the unknown noises in nature, I set randomly chosen parameters; (κ_R, σ_G) denotes (the percent of the window size to be removed, the standard deviation for Gaussian noise). The exact value for the noise is chosen randomly, which is less than the defined parameter. Both noises are applied together and I define variations of noises as three levels; Level 1 (0.15, 0.06), Level 2 (0.22, 0.09), and Level 3 (0.30, 0.12). Note, the classification models were trained with the original training set.



(a) Teacher: WRN28-1 (b) Teacher: WRN28-3

Figure 4.10: Accuracy (%) with Various Knowledge Distillation Methods for Various Noise Severity Levels on GENEActiv. Students Are WRN16-1 (1D CNNs).

As shown in Figure 4.10, TPKD (with orthogonal features) outperforms others in all cases. The results for WRN28-1 teachers show that KD and Base perform better than learning from scratch. The accuracy of Base is higher than KD, which implies that topological features complement the performance. However, when teachers are WRN28-3, there is a case that results of both KD and Base are lower than the model trained from scratch. Even if both models show better performance when testing set is not corrupted, they are impatient with noises. Because the capacity of the teacher is much higher than the one of the student, the knowledge difference is larger and it is more difficult to get benefits from distillation. In this case, only TPKD outperforms

learning from scratch in all cases. Thus, TPKD helps reducing the knowledge gap to distill a better student.

4.4.5 Computational Time

Table 4.9: Processing Time of Various Models on GENEActiv.

Model	Learning from scratch		KD		TPKD (w/ Orth.)
	TS (1D)	PImage (2D)	TS	PImage	TS+PImage
	WRN28-3	WRN16-3	WRN16-1 (1D CNNs)		
Accuracy (%)	69.23	59.8	69.71	68.76	**71.74**
GPU (sec)	29.94	356.92 (PIs on CPU) +13.63 (model)	**15.23**		
CPU (sec)	1977.89	356.92 (PIs on CPU) +11191.45 (model)	**16.66**		

I compare the computational time of various methods for testing set on GENEActiv. I implemented the evaluation on a desktop with a 3.50 GHz CPU (Intel® Xeon(R) CPU E5-1650 v3), 48 GB memory, and NVIDIA TITAN Xp (3840 NVIDIA® CUDA® cores and 12 GB memory) graphic card (NVIDIA (2016)). I tested approximately 6k samples with a batch size of 1. In Table 4.9, the considered accuracies are the best ones from Table 4.1 and 4.5. Since the time is required to generate PIs on the CPU, a model learned from scratch with PIs takes the largest amount of time in the table. A WRN16-1 (1D CNNs) student from TPKD takes the lowest time with the best accuracy. The result on the CPU strongly presents that a model compression method such as KD is required to run on small devices having limited power and computational resources.

4.5 Conclusion

In this chapter, I propose a framework in knowledge distillation utilizing topological representations on wearable sensor data, reducing the statistical gap between the teacher and student by orthogonal features and an annealing strategy. I evaluated the effectiveness of the proposed method, TPKD, under a variety of combinations of KD in classification. TPKD showed more accurate and efficient performance than baselines, which is significant in various applications running on edge devices. In future work, I aim to extend the proposed method by leveraging more various teachers trained with different representations (e.g. Gramian Angular Field based images) of time series data. Also, I would like to explore the effects of augmentation methods on the representations for using multiple teachers in knowledge distillation.

Chapter 5

CONSTRAINED ADAPTIVE DISTILLATION BASED ON TOPOLOGICAL PERSISTENCE FOR WEARABLE SENSOR DATA

5.1 Introduction

Converting wearable sensor data to impactful health applications continues to be challenging. The sources of variability in the raw sensor data include a) sensor-level noise characteristics, b) drifts in sampling rates, c) gaps in recorded sensor data, d) intrinsic variability in physiological signals, e) and variability due to sensor placement and particular human movements. These issues make training robust machine learning models with small datasets that much harder, calling for new approaches to describe and account for such variabilities. In this context, topological data analysis (TDA) has been used for representing time-series data with robustness to many types of signal perturbation (Adams *et al.* (2017); Turkeš *et al.* (2021)). These methods have achieved great success in various fields such as human activity recognition (Rieck *et al.* (2020); Som *et al.* (2020)), disease classification (Rieck *et al.* (2020); Nawar *et al.* (2020)), and shape and texture classification (Guo *et al.* (2018)). Particularly, persistence images (PIs) have been widely used to representations that are stable to signal perturbations. However, extracting PIs by TDA requires large computational and time resources, which are particularly difficult for small devices with limited computational power and real time systems on CPU (Hensel *et al.* (2021)).

Beyond just the computational load of TDA, it has also been found that the TDA features have many different data structures like barcodes, persistence diagrams, which can be featurized in many ways, but their integration with contempo-

rary machine-learning techniques has required independently computing the features and fusing with deep-features later (Adams *et al.* (2017); Edelsbrunner and Harer (2022)). Also, TDA features are computationally difficult to integrate with time-series features to create a unified model because of their heterogeneous dimension sizes and statistical characteristics (Som *et al.* (2020)). However, a careful use of knowledge distillation can address both of these issues by creating an integrated student model that blends the benefits of both TDA features and deep-features without requiring separate computation at test-time.

In this chapter, I address these issues by employing knowledge distillation (KD) which is a promising solution to produce a compact model (student) from a larger model (teacher). KD has been demonstrated to be effective in activity recognition and wearable sensor data analysis (Chen *et al.* (2018); Zhang *et al.* (2021); Qi *et al.* (2023); Cheng *et al.* (2023); Jeon *et al.* (2022b); Gou *et al.* (2021)). Also, KD has been broadly used to design a real-time system (Thai *et al.* (2022); Baghersalimi *et al.* (2022); Angarano *et al.* (2023); Remigereau *et al.* (2022)). Incorporating multiple teachers in KD has been shown to improve performance by leveraging various features (Gou *et al.* (2021); Liu *et al.* (2020); Zhang *et al.* (2022)), which are generally implemented in a unimodal manner. I utilize multiple teacher networks trained with the raw time-series and persistence images generated by TDA, respectively. Importantly, a single student is implemented with only time-series data as an input. However, I found two significant challenges in utilizing different teachers in KD: (1) The large discrepancy between the one-dimensional time-series and two-dimensional TDA feature representations makes it difficult to effectively fuse multimodal features and run models on a unified framework. (2) Due to the different architectural designs of teachers and student (e.g., 1D CNNs vs. 2D CNNs), it is challenging to extract similar structural features that allow the student to benefit from distillation.

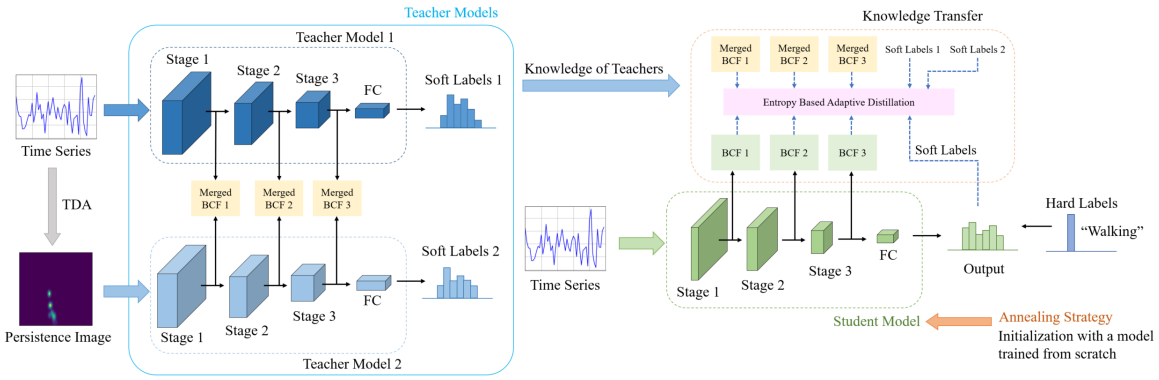


Figure 5.1: The Overview of Constrained Adaptive Distillation Based on Topological Persistence (CADTP). A Compact Student Model Is Trained by Using Two Teachers, Which Are Learned with Different Representations of the Same Raw Time-series Data. BCF Denotes Batch and Channel Similarity Features.

To address these problems, I propose a new framework, named Constrained Adaptive Distillation Based on Topological Persistence (CADTP), which uses multimodal inputs in KD using two different teachers and a single student. An overview of the proposed method is presented in Figure 5.1. Firstly, to obtain topological features, PIs are extracted from persistence diagrams. I train two models with time-series data and PIs, respectively. In the second step, the pre-trained models serve as teacher models in KD to distill a single student. Logits from two teachers are used independently for distillation. To address the knowledge discrepancy between two teachers, an entropy based adaptive weighting mechanism is employed to measure the confidence of knowledge and give more weight to the teacher with lower entropy values for each sample. To preserve desirable effects from both teachers, I propose a novel adaptive weighting mechanism with constraints to balance the contribution of teachers. The weights are initialized but gradually increase or decrease as the epoch number grows. This enables a student to learn to be more confident and keep beneficial knowledge from different teachers by placing more weight on the confident knowledge between

the two. In the third step, to integrate different structural information from different models and to provide strong supervision, I utilize the batch and channel correlation maps of intermediate representations within a mini-batch, which aids in matching different dimensional sizes of knowledge. Batch and channel similarity features capture distinct activations, providing complementary information to each other.

The contributions of this chapter are as follows:

- I propose a new framework with knowledge distillation, which transfers time-series and topological features to a student using time-series data only as an input.
- I propose a technique for adaptive distillation that balances the influence of different teachers based on entropy to effectively transfer knowledge despite the statistical difference in their features.
- I utilize batch and channel similarities from intermediate layers and an annealing strategy to integrate diverse knowledge from multiple teachers, allowing a single student to effectively learn desirable features.
- I rigorously evaluate the effectiveness of the proposed method in various aspects using different teacher-student combinations and feature visualization on wearable sensor data for human activity recognition.

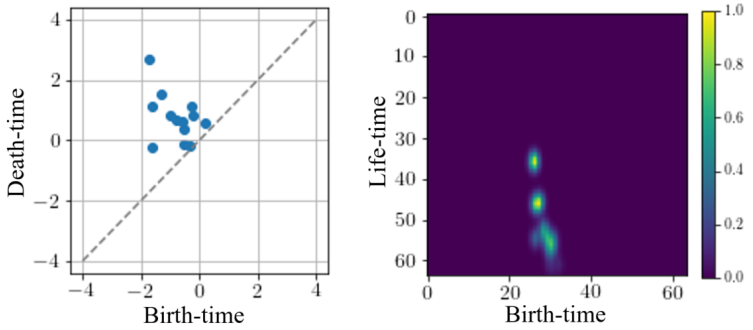
The rest of the chapter is organized as follows. In section 5.2, I provide a brief overview of creating PIs, KD techniques, and an annealing strategy. In section 5.3, I introduce the proposed new framework for KD. In section 5.4, I describe our experimental results and analysis. In section 5.5, I discuss our findings and conclusions.

5.2 Background

5.2.1 Topological Feature Extraction

The integration of TDA with machine learning has shown robust performance in many applications (Gholizadeh and Zadrozny (2018); Zeng *et al.* (2021); Edelsbrunner and Harer (2022)). TDA aims to capture the intricate shape of complex data – persistent homology is one of the popular algorithms which is able to capture variations in topologically meaningful structures over multiple scales of the data, formed by the interlinking of points, edges, and triangles, and in general simplicial complexes, by a dynamic thresholding process called filtration (Edelsbrunner *et al.* (2002)). From this filtration, the birth and death of these topological cavities can be described as a point (x, y) in the persistence diagram (PD), where x and y are coordinates of planar scatter points (Adams *et al.* (2017); Edelsbrunner and Harer (2022)). Applying PDs directly to complex machine learning tasks is challenging because they have intrinsically heterogeneous statistical characteristics. PDs are multi-sets on $\mathbb{R}^2$ implying the number and locations of the scatter points that can be different in the presence of perturbations on the underlying data, which require more expressive representations. Ordering the scatter points based on their persistence (lifetime) is a common way to vectorize PDs, which makes it suitable for machine learning tasks.

Persistence image (PI) is a different type of vector representation of a PD. To construct the PI, PD is first projected onto a persistence surface (PS) $\rho : \mathbb{R} \to \mathbb{R}^2$, which is defined by a normalized symmetric Gaussian function as well as a weighting function (Adams *et al.* (2017); Hensel *et al.* (2021)). The PS is discretized over a standard grid. PI is generated by incorporating the PS over the grid and is represented as a matrix of pixel values. Higher values of a PI indicate high-persistence points in the PD. Figure 5.2 depicts an example of a PD and its PI. However, due



(a) Persistence Diagram (b) Persistence Image

Figure 5.2: PD and Its Corresponding PI. In PD, Based on Weighting Function, Points with Higher Life-time Appears Brighter.

to the high computational complexity required to extract PIs by TDA (Som *et al.* (2020)), this method is difficult to use on small devices with limited power and computational resources. To solve this issue, in this chapter, I propose a framework based on knowledge distillation that trains a smaller single student model with topological knowledge to generate good performance as a larger model.

5.2.2 Application of TDA for Activity Recognition

There are lots of works utilizing topological knowledge in applications for activity recognition (Pachauri *et al.* (2011); Nawar *et al.* (2020); Som *et al.* (2020)). These methods use vectorized topological features from PI as inputs to machine learning methods, generally resulting in robustness to signal perturbation. Nawar *et al.* (Nawar *et al.* (2020)) encoded values in PI with forces and moments of data and utilized SVM for classification, which showed better performance than using time-series data. However, the method requires various pre-processing steps for training as well as testing to extract topological features by TDA and transform knowledge into manually defined terms. PI-Net (Som *et al.* (2020)) is to generate PI through CNNs to utilize topological features efficiently instead of using conventional protocols running

on the CPU. To adopt topological features and improve performance, the method combines both time-series and topological features simultaneously to train and test a model. However, running separate models and concatenating features increase the complexity of the model and time consumption. Based on these insights, in this chapter, I propose a framework to generate a single small model using time-series data only, which does not require pre-processing to generate PI, nor needing to run different models separately at test-time.

5.2.3 Knowledge Distillation

Knowledge distillation is the process of training a smaller model from the knowledge of a larger model. KD was first introduced by Buciluǎ *et al.* (Buciluǎ *et al.* (2006)) and further developed by Hinton *et al.* (Hinton *et al.* (2015)). During the KD process, soft labels from the outputs of a teacher network are utilized, which have more useful information than just a hard label and enable the student network to easily encode the knowledge of the teacher (Hinton *et al.* (2015)). For traditional KD, the loss function for training a student is:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\mathcal{CE}} + \lambda\mathcal{L}_{\mathcal{KD}}, \tag{5.1}$$

where $\mathcal{L}_{\mathcal{CE}}$ is the cross entropy loss, $\mathcal{L}_{\mathcal{KD}}$ is KD loss, and λ is a hyperparameter; $0 < \lambda < 1$. By the cross entropy loss, the difference between the output of the softmax layer for a student network and the ground-truth label is penalized:

$$\mathcal{L}_{\mathcal{CE}} = \mathcal{Q}(\sigma(t_S), y), \tag{5.2}$$

where $\mathcal{Q}(\cdot)$ is a cross entropy loss function, $\sigma(\cdot)$ is a softmax function, t_S is the logits of a student, and y is a ground truth label. The outputs of student and teacher are matched by KL-divergence loss:

$$\mathcal{L}_{\mathcal{KD}} = \tau^2 KL(p_T, p_S), \tag{5.3}$$

where, $p_T = \sigma(t_T/\tau)$ is a softened output of a teacher network, $p_S = \sigma(t_S/\tau)$ is a softened output of a student, and τ is a hyperparameter; $\tau > 1$. Vanilla KD utilizes a fully trained teacher. Cho *et al.* (Cho and Hariharan (2019)) investigated the effects of early stopping for KD (ESKD) to distill a better student. To obtain the best performance, I use ESKD that improves the efficacy of KD (Cho and Hariharan (2019)).

As an extension of response based knowledge using logits, feature based knowledge distillation has been used to improve performance (Romero *et al.* (2015); Gou *et al.* (2021); Zagoruyko and Kmodakis (2017); Tung and Mori (2019)). Firstly, the intermediate representations were introduced in Fitnets (Romero *et al.* (2015)). The key idea behind feature matching in KD is to directly match the features of the teacher and student. Many different variants have been proposed to achieve this indirectly, such as the approach of Tung *et al.* (Tung and Mori (2019)), which utilizes similarity between a mini-batch of samples to transfer knowledge. The dimensions of the teacher and student are the same, which is defined by the size of the mini-batch. To calculate the batch similarity, the activation map $A \in \mathbb{R}^{b \times b}$ is produced as follows:

$$A = F_b \cdot F_b^{\top}; F_b \in \mathbb{R}^{b \times chw}, \tag{5.4}$$

where F_b is reshaped features from an intermediate layer of a model, b is the size of a mini-batch, c is the number of output channels, and h and w are the height and width of the output, respectively. These methods using intermediate representations have been popularly used in KD, however, they generally focus on utilizing a single teacher in a unimodal manner.

To transfer more useful information, using multiple teachers has been proposed (Gou *et al.* (2021); Liu *et al.* (2020); Zhang *et al.* (2022)). Since different teachers can produce diverse knowledge, richer knowledge can be leveraged to improve the perfor-

mance of a student (Gou *et al.* (2021)). Despite initial attempts (You *et al.* (2017); Kwon *et al.* (2020)), the problem remains difficult to solve. Combining knowledge from various teachers in KD poses a challenge as it can result in loss of characteristics of each and having them affect each other as noise components. Also, a data sample or label for training a teacher cannot always be used to train or test a student. Furthermore, different modalities in KD increase the knowledge difference between a teacher and student, which results in performance degradation (Gou *et al.* (2021)).

To resolve these problems, I develop a framework in KD using constrained adaptive weighting mechanism, based on entropy, to control the effects of two teachers trained with time-series and topological features. This allows for the transfer of richer information effectively to a single student, which uses the raw time-series data only as an input. The details of the proposed method is described in section 5.3.

5.2.4 Simulated Annealing in KD

Kirkpatrick *et al.* (Kirkpatrick *et al.* (1983)) introduced simulated annealing, which has been applied to various fields with machine learning for solving optimization problems (Yang (2020)). Jafari *et al.* (Jafari *et al.* (2021)) introduced an annealing KD to use two stages to address the capacity gap problem between the outputs of teacher and student networks. In the first stage, while the difference in logits between teacher and student is reduced in a regression task, a temperature parameter decreases as the epoch number increases. In the second stage, the student is fine-tuned with the hard labels by cross entropy loss. Dong *et al.* (Dong *et al.* (2022)) also used two stages in KD. A student learns from a teacher when the teacher model outperforms, otherwise, the student is trained by hard labels. To avoid the teacher's limited accuracy issue, a dynamic annealing weight is used, which increases linearly as finetuning epochs increase. An annealing strategy of the proposed method has

different aspect, compared to prior studies (Clark *et al.* (2019); Jafari *et al.* (2021); Dong *et al.* (2022)). For the proposed method, multiple teachers are trained with different modalities – time-series and persistence image data – but only one type of data is used to train and test a single student. Since the features from teachers and their contributions are different, I apply an annealing strategy that reduces the search space and forces the student to learn enjoyable features for better performance by using the weights of a model trained from scratch. In detail, the strategy is to initialize the student model with weight values from a model learned from scratch, instead of randomly chosen values. This allows the student to preserve desirable features for improved performance – the final model operates only on raw time-series data as input. In this way, the knowledge gap between the teachers and student is also mitigated.

5.3 Proposed Approach

The proposed method utilizes two teachers trained with different data to train a student. Firstly, PIs are extracted from time-series data through TDA to incorporate topological features. The two teachers are trained with the raw time-series data and the extracted PIs, respectively. Secondly, logits of teacher and student networks are used to calculate entropy for balancing the effects of two teachers, considering statistical differences in multimodalities. In the third, correlation maps for batch and channel similarities within a mini-batch are utilized for distillation to provide plentiful information, which allows for the use of differently designed teachers and student. Additionally, an annealing strategy for knowledge distillation is applied to optimize the weight of the student model. Finally, a robust single student is distilled. The details of the proposed method are explained in the following section.

5.3.1 Persistence Image Extraction

To compute PIs, firstly, I utilize Scikit-TDA python library (Saul and Tralie (2019)) and the Ripser package for generating PDs, as described in Som *et. al.* (Som *et al.* (2020)). Level-set filtration PDs for time-series data are computed, which creates a summary representation of different peaks in the signal. PIs are generated in the form of a grid representing birth-time vs. lifetime information. The dimension size of one PI is $m \times m \times c$, where m and c are a constant value and the number of channels for a sample. Secondly, I train a model with the extracted PIs with supervised learning, where the model is used as a teacher model, transferring topological features to a student model.

5.3.2 KD with Multiple Teachers

To generate PIs, TDA requires a large amount of computational resources, which is one of the critical burdens at test-time. To this end, I adopt KD to distill a small model using time-series data alone as an input, to acquire beneficial topological features from a teacher.

Distillation with Logits of Different Teachers

Since the proposed method uses two teachers transferring knowledge of logits separately, no additional function such as concatenation or hidden layers is necessarily needed. KD loss to utilize logit features of two teachers is:

$$\mathcal{L}_{\mathcal{KD}m} = \tau^2 \left(\alpha KL(p_{T_1}, p_S) + (1-\alpha) KL(p_{T_2}, p_S) \right), \tag{5.5}$$

where α is a hyperparameter to control the losses from different teachers, and p_{T_1} and p_{T_2} are softened outputs of teachers learned with time-series data and PIs, respectively.

Entropy-based Constrained Adaptive Distillation

The proposed method uses two teachers trained with different data and designs, which generate statistically heterogeneous features that may interfere with each other. To transfer effective knowledge from multiple teachers, I use the entropy of teachers, which can be utilized as an uncertainty indicator (Kwon *et al.* (2020)). However, since teachers are implemented with multimodalities, two models generate statistically dissimilar features and the entropy values between them are significantly different. This can produce a large discrepancy between the two entropy values, resulting in biased balancing and poor adjustment of losses from the two teachers. To this end, I propose constrained adaptive distillation based on entropy. If the entropy value of labels is smaller, the effect of the KD loss is more important (Long *et al.* (2018); Kwon *et al.* (2020)). Based on this factor, the weight of a teacher is made larger if the model produces smaller entropy. To make a function to adjust the weights gradually, I adopt a part of sigmoid curve whose input is over 0. The weight value α for teacher losses begins at 0.5 and is adjusted dynamically as the epoch number increases. α is defined within the specified range. Since different teachers perform differently at each input data, I set α at each sample. The weight α is determined according to the following rule:

$$\alpha_i = \begin{cases} 0.5 + \left(1/(1+e^{-epoch/\beta}) - 0.5\right)/\kappa & \text{if } \mathcal{H}(t^i_{T_1}) < \mathcal{H}(t^i_{T_2}) \\ 0.5 - \left(1/(1+e^{-epoch/\beta}) - 0.5\right)/\kappa & \text{otherwise} \end{cases}, \tag{5.6}$$

where, $\mathcal{H}(t^i_{T_1})$ and $\mathcal{H}(t^i_{T_2})$ denote the entropy of $t^i_{T_1}$ and $t^i_{T_2}$ for a sample i, respectively. β and κ are constant values to manage the saturation point by the epoch number. KD loss with constrained adaptive weights based on entropy of two teachers can be written as:

$$\mathcal{L}_{\mathcal{KD}ent} = \frac{1}{n}\sum_{i=1}^{n} \tau^2 \left(\alpha_i KL(p^i_{T_1}, p^i_S) + (1-\alpha_i) KL(p^i_{T_2}, p^i_S)\right), \tag{5.7}$$

where n is the number of samples. Therefore, more knowledge is transferred to the student from teachers that have lower entropy values.

Extracting Features of Different Teachers

To provide more comprehensive knowledge from the teachers, I use intermediate features also in distillation. However, since two teachers are trained with different modalities, and teachers and the student have different architectures, it is difficult to transfer the information directly. To accommodate heterogeneous features from networks with different structures, I use the similar method proposed in Tung *et al.* (Tung and Mori (2019)), which can easily make features match the dimensions of activation maps from different models, as explained in equation (5.4). The batch similarity matrices $A \in \mathbb{R}^{b\times b}$ have the same size for teachers and the student. The pattern of the activation map is determined according to the same or different classes. Specifically, if two samples are in the same category, a model generates similar activation maps, which enables a student to acquire beneficial knowledge from a teacher.

Although the batch similarity provides considerable information, more diverse contexts can still be transferred to distill a superior student model in KD. To leverage different contexts, I extract channel similarity that highlights the channel relationship within a mini-batch, which can be simply obtained by reshaping the features of the intermediate layer. To calculate the channel similarity, the activation map $G \in \mathbb{R}^{c\times c}$ is produced as follows:

$$G = F_c \cdot F_c^{\top}; F_c \in \mathbb{R}^{c\times bhw}, \tag{5.8}$$

where F_c is reshaped features from an intermediate layer of a model. G can have different sizes for different layers.

Figure 5.3 shows the batch and channel similarity maps from two teachers. The similarity maps highlight differently and show dissimilar patterns. Thus, these maps

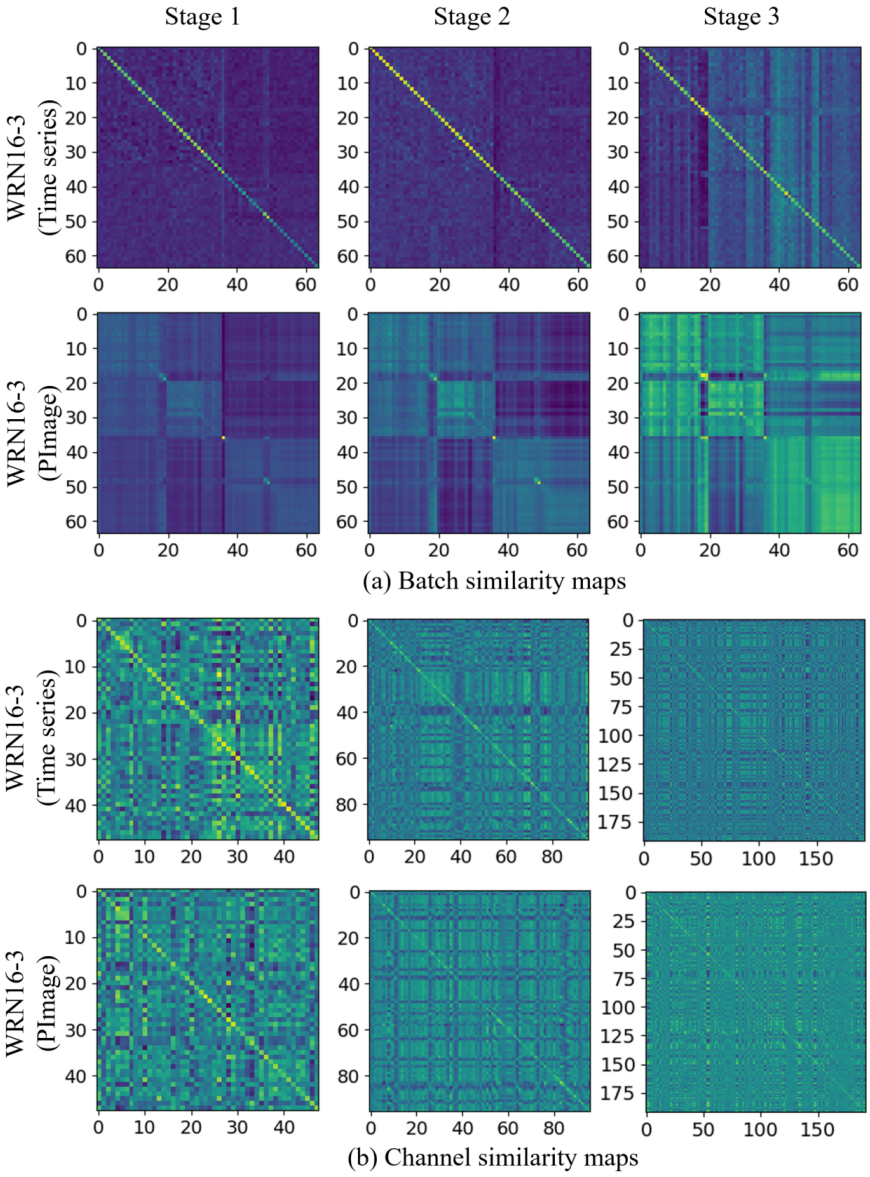


Figure 5.3: Examples of Activation Similarity Maps A and G Produced by a Layer for the Indicated Stage of the Network for a Batch on GENEActiv. High Similarities for Samples Within the Batch Are Shown with High Values. The Blockwise Pattern Is More Prominent for Batch Similarity Maps Using Persistence Image. The Maps with Different Modalities and Similarities Represent Dissimilar Patterns, Which Implies That These Maps Can Capture Diverse Semantics of the Dataset.

can transfer complementary information to each other. Also, two teachers generate very different patterns for both activation maps. This is due to the fact that the two models are trained with different modalities and produce dissimilar features, which can provide misinformation to the student (Kwon *et al.* (2020); Gou *et al.* (2021)). By using fused knowledge, the effects of noise from the teachers can be reduced and the student can better interpret context. To integrate the information, I utilize the calculated weight α. These maps are generated within a mini-batch, and the average of their α is used. The merged map of batch similarity from teachers with the averaged weight value α_{avg} is as follows:

$$A_T^{(l)} = \alpha_{avg} A_{T_1}^{(l^{T_1})} + (1 - \alpha_{avg}) A_{T_2}^{(l^{T_2})}, \tag{5.9}$$

where $A_T^{(l)} \in \mathbb{R}^{b \times b}$ is the generated map from the activation maps of a layer pair (l^{T_1} and l^{T_2}) of two teachers A_{T_1} and A_{T_2}. The merged map of channel similarity from teachers is as follows:

$$G_T^{(l)} = \alpha_{avg} G_{T_1}^{(l^{T_1})} + (1 - \alpha_{avg}) G_{T_2}^{(l^{T_2})}, \tag{5.10}$$

where $G_T^{(l)} \in \mathbb{R}^{c^{(l)} \times c^{(l)}}$ is the generated map from the activation maps of a layer pair (l^{T_1} and l^{T_2}) of two teachers G_{T_1} and G_{T_2}. If G_{T_1} and G_{T_2} have different size, larger one is resized to match the smaller one. By merging the maps, the similarities between two teachers are more highlighted.

Transferring Features from Multiple Teachers

$\widetilde{A_T}$ and $\widetilde{G_T}$ are obtained by normalization as: $A_T / \|A_T\|_2$ and $G_T / \|G_T\|_2$, respectively. $\widetilde{A_S}$ and $\widetilde{G_S}$ are normalized maps from the student A_S and G_S, respectively. If G_T and G_S have different size, the larger one is resized to meet the size of the smaller one. The overview of transferring knowledge with similarity maps is described in Figure

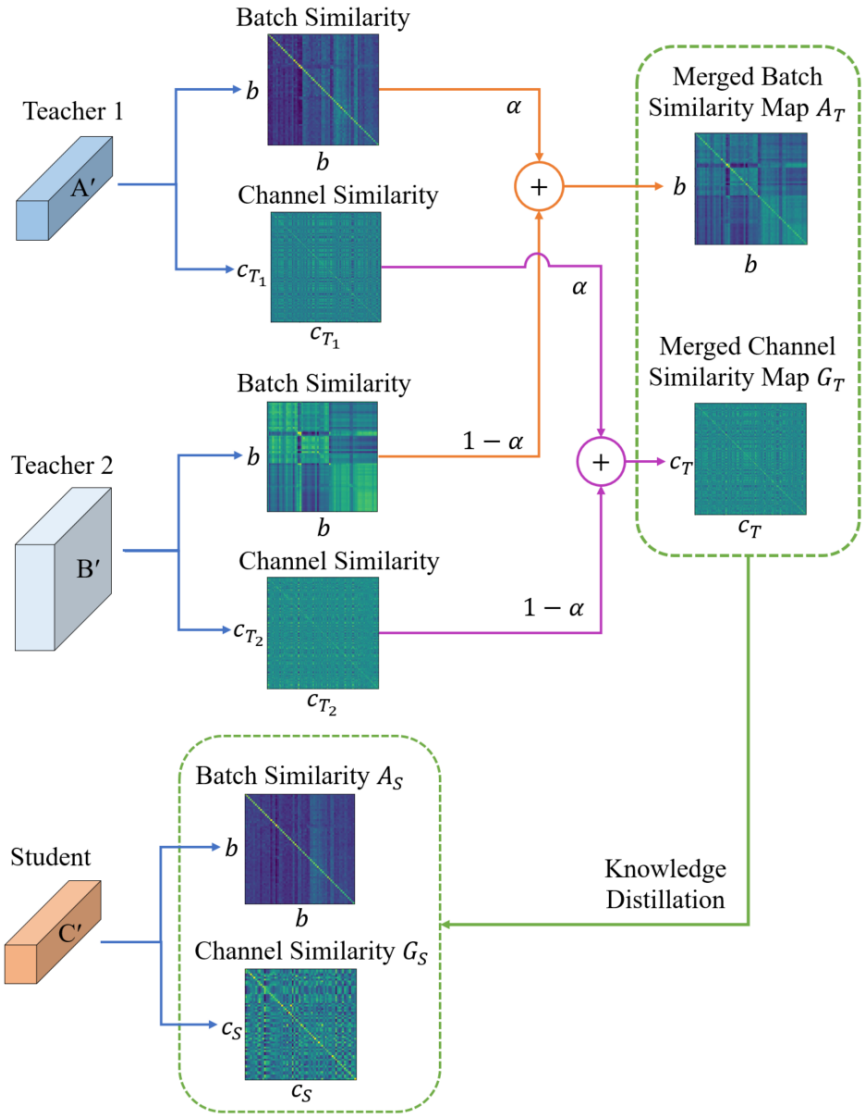


Figure 5.4: Framework of Extracting and Transferring Similarity Features from Different Teachers. A′ and B′ Denote Mini-batch Features at a Layer of Teacher1 and Teacher2, Respectively. C′ Denotes Mini-batch Features at a Layer of Student.

5.4. By minimizing the difference between the teachers and student, the information from similarity maps are transferred as follows:

$$\mathcal{L}_{sim} = \frac{1}{|L|} \sum_{(l,l^S)\in L} \left(\frac{\gamma_b}{b^2} \left\| \widetilde{A_T^{(l)}} - \widetilde{A_S^{(l^S)}} \right\|_F^2 + \frac{\gamma_c}{c_{(l)}^2} \left\| \widetilde{G_T^{(l)}} - \widetilde{G_S^{(l^S)}} \right\|_F^2 \right), \tag{5.11}$$

where L collects the layer pairs (l and l^S), γ_b and γ_c are hyperparameters to balance the effects of batch and channel similarities, $c_{(l)}$ is the size of $G_T^{(l)}$, and $\|\cdot\|_F$ is the Frobenius norm (Tung and Mori (2019)). In this way, the student can get the beneficial diverse knowledge from multiple teachers with the raw time-series and topological representations. The overall learning objective of the proposed method can be written as:

$$\mathcal{L}_{CADTP} = (1-\lambda)\mathcal{L}_{\mathcal{CE}} + \lambda\mathcal{L}_{\mathcal{KD}ent} + \eta\mathcal{L}_{sim}, \tag{5.12}$$

where η is a hyperparameter to control the effect of loss $\mathcal{L}_{sim}$.

5.3.3 Annealing Strategy for KD

Since teachers and student have different architectures and are trained with different data, they generate dissimilar features, which produce statistical gaps and cause degradation in KD (Gou *et al.* (2021); Jafari *et al.* (2021)). To mitigate the effects of the knowledge gap, I use an annealing strategy in KD for the proposed framework. Firstly, a small model that has the same architecture of a student is learned from scratch. Secondly, when the weight values of a student model are initialized for the training process in KD, the values are determined by the pre-trained model, instead of randomly chosen values. Then, the knowledge difference between teachers and student is reduced, and the search space for optimization is decreased. Also, the strategy enables the student to preserve more desirable features for implementing with time-series data while teachers transfer their own features.

5.4 Experiments

In this section, I describe datasets used for evaluation and experimental settings. I demonstrate the proposed method with various teacher-student combinations on

wearable sensor data. I analyze the proposed method under different noise levels and various hyperparameters. Further, I investigate the effectiveness of CADTP with visualization of feature maps and generalizability analysis. Finally, I compare and contrast the computational time with different methods.

5.4.1 Data Description and Experimental Settings

Data Description

I evaluate the proposed method with wearable sensor data on GENEActiv and PAMAP2 datasets.

GENEActiv. GENEActiv (Wang *et al.* (2016)) is an experimental device calibration dataset collected with GENEActiv sensor which is a light-weight, waterproof, and wrist-worn tri-axial accelerometer with sampling frequency of 100 Hz. The dataset was comprised of over 150 generally healthy adults roughly balanced by sex, age (18-64 years of age), and body mass index. All participants provided consent prior to participation. I use 14 daily activities used as in (Jeon *et al.* (2022b)). Each activity class has over 900 samples. I use full non-overlapping window size of 500 time-steps (5 seconds). The number of subjects for training and testing are 131 and 43, respectively. The number of samples for training and testing are approximately 16k and 6k, respectively.

PAMAP2. PAMAP2 (Reiss and Stricker (2012)) is a publicly accessible dataset collected by measurements of heart rate, temperature, accelerometers, gyroscopes, and magnetometers with 100Hz of sampling frequency for 9 subjects (24-32 years of age). The sensors were placed on hands, chest, and ankles of the subject. I use 12 daily activities with 40 channels, which were recorded from the heart rate and 4 IMUs, where activities are lying, sitting, standing, walking, etc. To compare with previous

methods, the recordings are downsampled to 33.3Hz. The evaluation protocol on this dataset follows leave-one-subject-out. Data dropping and connection loss occurred because data was collected using wireless sensors, so missing data is included. The dataset has non-uniform distribution. I utilize 100 time-steps (3 seconds) of a sliding window for a sample with 22 time-steps (660 ms) of step size for segmenting the sequences, which allows semi-non-overlapping sliding windows with 78% overlapping (Reiss and Stricker (2012)).

Experimental Settings

To extract PIs, for GENEActiv, the Gaussian function parameter in PD is 0.25 and the birth-time range for PI is determined between -10 to 10, which are the same as in the previous study (Som *et al.* (2020)). For PAMAP2, the Gaussian function parameter and birth-time range are set as 0.015 and [-1, 1], respectively. Each PI is normalized by its maximum intensity value. m is set to 64 for both datasets. To train network models in experiments, I set the total number of epochs as 200, using SGD with momentum of 0.9, 64 as the batch size, and a weight decay as 1×10^{-4}. I have different strategies for training models with time-series and image representations. The model trained with time-series data is incorporated with 1D convolutional layers, on the other hand, the one trained with image data is designed with 2D convolutional layers. To train a model with time-series data, the initial learning rate is 0.05 which decreases by 0.2 at 10 epochs and drops down by 0.1 every $[\frac{t}{3}]$ where t is the total number of epochs. For image data, a model is trained with 0.1 of the initial learning rate, which decreases by 0.5 at 10 epochs and drops down by 0.2 at 40, 80, 120, and 160 epochs. To evaluate the performance of the proposed method, I use WideResNet (WRN) (Zagoruyko and Komodakis (2016)) to construct different combinations of teachers and student, which is popularly used in validation of KD (Cho and Hariharan

(2019); Jeon *et al.* (2022b)). Also, WRN has been used to design real-time system (Lee *et al.* (2017); Song *et al.* (2021); Kania and Markowska-Kaczmar (2018)). As the previous works do (Jeon *et al.* (2022b)), τ and λ are set as 4 and 0.7 for GENEActiv, and as 4 and 0.99 for PAMAP2, respectively. I run 3 times and the best averaged accuracy and standard deviation are reported for the following experiments. I perform baseline comparisons with traditional KD (Hinton *et al.* (2015)), attention transfer (AT) (Zagoruyko and Kmodakis (2017)), similarity-preserving knowledge distillation (SP) Tung and Mori (2019), and simple knowledge distillation (SimKD) (Chen *et al.* (2022)), which are popularly used for distillation. α_{AT} and γ_{SP} are set as 1500 and 1000 for GENEActiv, and 3500 and 700 for PAMAP2, respectively. Also, I compare with DIST (Huang *et al.* (2022)), which considers intra- and inter-class relationship for knowledge transfer. Additionally, I compare with multi-teacher based approaches such as AVER (You *et al.* (2017)), EBKD (Kwon *et al.* (2020)), CA-MKD (Zhang *et al.* (2022)), Base (Jeon *et al.* (2022a)), and AdTemp (Jeon *et al.* (2022a)). Since I use different dimensional input data and structured teachers, only the outputs from the last layer (logits) are used for baselines in distillation. α for baselines is set as 0.5. For Base, α is 0.7 and 0.3 for GENEActiv and PAMAP2, respectively.

5.4.2 Various Capacity of Teachers

In this section, I evaluate the proposed method with various capacities of teachers that are trained with time-series data and PIs, respectively. WRN16-1 (1D CNNs) is used as a student model. γ_b is 1. Details of models for teachers and a student, used for experiments, are summarized in Table 5.1, representing model complexity and the number of trainable parameters.

The results with various teachers on GENEActiv are described in Table 5.2. Note, "time-series" and "PImage" denote results of KD methods with Teacher1 trained with

Table 5.1: Details of Teacher and Student Network Architectures. Compression Ratio Is Calculated with Two Teachers.

DB	Teacher1 (1D CNNs) & Teacher2 (2D CNNs)	Student	FLOPs			# of params			Compression ratio
			(Teacher1)	(Teacher2)	(Student)	(Teacher1)	(Teacher2)	(Student)	
GENEActiv	WRN16-1	WRN16-1	11.03M	108.97M	11.03M	0.06M	0.18M	0.06M	25.93%
	WRN16-3		93.95M	898.52M		0.54M	1.55M		2.94%
	WRN28-1		22.22M	224.28M		0.13M	0.37M		12.36%
	WRN28-3		192.01M	1923.93M		1.12M	3.29M		1.39%
PAMAP2	WRN16-1	WRN16-1	2.39M	131.02M	2.39M	0.06M	0.18M	0.06M	25.88%
	WRN16-3		19.00M	921.03M		0.54M	1.56M		3.01%
	WRN28-1		4.64M	246.56M		0.13M	0.37M		12.52%
	WRN28-3		38.64M	1947.13M		1.12M	3.30M		1.43%

time-series data and Teacher2 trained with PIs, respectively. "TS", "Ann.", "Ent." denote using a teacher trained with time-series data, applying an annealing strategy, and using entropy based constrained adaptive distillation, respectively. "Ba." and "Ch." denote using batch and channel similarity features in distillation. The numbers in brackets for Teacher1, Teacher2, and Student are their accuracy. η is 700. β and κ are 1.5 and 2.5, respectively. The γ_c values of the teachers in the table are 0.2, 0.01, 0.01, and 0.2, from left to right. As shown in the table, CADTP (with entropy based constrained adaptive distillation) shows the best results in all cases. Ann. performs better than AVER, indicating that the annealing strategy is useful to improve the performance. In most of the cases, CADTP (w/o Ent.) also performs better than other baselines (Ann., Ann.+Ba., and Ann.+Ch.). That is, as more information is provided, the more improvement is seen. Next, using larger teachers does not guarantee a better student, which corroborates the previous observations (Cho and Hariharan (2019)).

Table 5.2: Accuracy (%) with Various Knowledge Distillation Methods for Different Capacity of Teachers on GENEActiv.

Teacher1 (1D CNNs)		WRN16-1 (67.66)	WRN16-3 (68.89)	WRN28-1 (68.63)	WRN28-3 (69.23)
Teacher2 (2D CNNs)		WRN16-1 (58.64)	WRN16-3 (59.80)	WRN28-1 (59.45)	WRN28-3 (59.69)
Student (1D CNNs)		WRN16-1 (67.66±0.45)			
P1	KD	67.83±0.17	68.76±0.73	68.51±0.01	68.46±0.28
Time-series	KD	69.71±0.38	69.50±0.10	68.32±0.63	68.58±0.66
	AT	68.21±0.64	69.79±0.36	68.09±0.24	67.73±0.27
	SP	67.20±0.36	67.85±0.24	68.71±0.46	67.39±0.49
	SimKD	69.39±0.18	69.89±0.11	68.92±0.40	68.80±0.38
	DIST	68.20±0.28	69.71±0.15	69.23±0.19	68.18±0.60
TS+PImage	AVER	68.99±0.76	68.74±0.35	68.77±0.70	69.02±0.50
	EBKD	68.43±0.25	69.24±0.25	68.45±0.73	67.50±0.40
	CA-MKD	69.33±0.61	69.80±0.16	69.61±0.57	68.81±0.79
	Base	69.09±0.37	69.24±0.62	69.55±0.41	69.42±0.58
	AdTemp	69.80±0.68	70.10±0.39	70.01±0.83	69.55±0.51
	Ann.	70.04±0.22	70.27±0.06	70.15±0.24	69.83±0.24
	Ann.+Ba.	70.43±0.15	70.48±0.37	70.40±0.16	69.98±0.31
	Ann.+Ch.	69.16±0.24	69.99±0.28	68.79±0.29	68.51±0.57
	CADTP (w/o Ent.)	70.90±0.59	70.39±0.20	70.53±0.26	71.18±0.59
	CADTP (w/ Ent.)	**71.91**±0.39	**71.68**±0.25	**71.40**±0.27	**71.74**±0.23

To investigate with different size of window lengths and more previous methods, I test the methods with 7 classes of GENEActiv dataset, as do the previous study (Jeon *et al.* (2022b); Choi *et al.* (2018)). β and κ are 1.0 and 1.5, respectively. η parameters are 900 for window size of 500 and 100 for window size of 1000, respectively. γ_c are

Table 5.3: Accuracy (%) for Related Methods on GENEActiv with 7 Classes.

	Method	Window length	
		1000	500
Time-series	SVM (Cortes and Vapnik (1995))	86.29	85.86
	Choi *et al.* (Choi *et al.* (2018))	89.43	87.86
	WRN16-1	89.29±0.32	86.83±0.15
	WRN16-3	89.53±0.15	87.95±0.25
	WRN16-8	89.31±0.21	87.29±0.17
	ESKD (WRN16-3)	89.88±0.07	88.16±0.15
	ESKD (WRN16-8)	89.58±0.13	87.47±0.11
	Full KD (WRN16-3)	89.84±0.21	87.05±0.19
	Full KD (WRN16-8)	89.36±0.06	86.38±0.06
	AT (WRN16-1)	90.10±0.49	87.25±0.22
	AT (WRN16-3)	90.32±0.09	87.60±0.22
	SP (WRN16-1)	87.08±0.56	87.65±0.11
	SP (WRN16-3)	88.47±0.19	87.69±0.18
	SimKD (WRN16-1)	90.25±0.22	87.24±0.09
	SimKD (WRN16-3)	90.47±0.32	88.16±0.37
	DIST (WRN16-1)	90.18±0.31	87.62±0.02
	DIST (WRN16-3)	90.20±0.39	87.05±0.31
TS+PImage	AVER (WRN16-1)	90.01±0.46	87.53±0.16
	AVER (WRN16-3)	90.06±0.33	87.05±0.37
	EBKD (WRN16-1)	90.35±0.12	87.51±0.41
	EBKD (WRN16-3)	89.82±0.14	87.66±0.28
	CA-MKD (WRN16-1)	90.01±0.28	87.14±0.25
	CA-MKD (WRN16-3)	90.13±0.34	88.04±0.26
	Ann. (WRN16-1)	90.64±0.15	87.68±0.15
	Ann. (WRN16-3)	90.78±0.08	88.02±0.21
	CADTP (w/ Ent.) (WRN16-1)	90.85±0.31	**88.89**±0.29
	CADTP (w/ Ent.) (WRN16-3)	**91.48**±0.27	88.45±0.11

Table 5.4: Accuracy (%) with Various Knowledge Distillation Methods for Different Capacity of Teachers on PAMAP2.

Teacher1 (1D CNNs)		WRN16-1 (85.27)	WRN16-3 (85.80)	WRN28-1 (84.81)	WRN28-3 (84.46)
Teacher2 (2D CNNs)		WRN16-1 (86.93)	WRN16-3 (87.23)	WRN28-1 (87.45)	WRN28-3 (87.88)
Student (1D CNNs)		WRN16-1 (82.99±2.50)			
PI	KD	85.04±2.58	86.68±2.19	85.08±2.44	85.39±2.35
TS	KD	85.96±2.19	86.50±2.21	84.92±2.45	86.26±2.40
TS+PImage	AVER	85.82±2.16	86.00±2.45	85.17±2.38	86.64±2.24
	Ann.	86.05±2.23	86.74±2.25	85.89±2.25	86.72±2.26
	Ann.+Ba.	86.53±2.19	86.94±2.32	85.81±2.34	86.84±2.38
	Ann.+Ch.	86.81±2.04	87.25±2.18	86.13±2.16	86.99±2.17
	CADTP (w/o Ent.)	86.68±2.21	87.63±2.30	87.39±2.07	87.22±2.33
	CADTP (w/ Ent.)	**87.11**±2.04	**88.14**±2.07	**87.47**±2.06	**87.55**±2.27

0.2, 0.003, 0.2, and 0.02 for teachers of WRN16-1 for 500 window length, WRN16-3 for 500 window length, WRN16-1 for 1000 window length, and WRN16-3 for 1000 window length, respectively. In Table 5.3, CADTP achieves the best performing results, indicating that the proposed method aids in performance improvement. The results on PAMAP2 are described in Table 5.4. β and κ are 0.3 and 2.5, respectively. η is 200. The γ_c values of the teachers in the table are 0.02, 0.02, 0.2, and 0.2, from left to right. In all cases, CADTP (with Ent.) produces the best results. For this dataset, in most of the cases, CADTP (w/o Ent.) performs better than other baselines (Ann., Ann.+Ba., and Ann.+Ch.). Further, as shown in Table 5.5, CADTP

Table 5.5: Accuracy (%) for Related Methods on PAMAP2.

	Method	Accuracy (%)
Time-series	Chen and Xue (2015)	83.06
	Ha *et al.* (2015)	73.79
	Ha and Choi (2016)	74.21
	Kwapisz *et al.* (2011)	71.27
	Catal *et al.* (2015)	85.25
	Kim *et al.* (2012)	81.57
	WRN16-1	82.81±2.51
	WRN16-3	84.18±2.28
	WRN16-8	83.39±2.26
	ESKD (WRN16-3)	86.38±2.25
	ESKD (WRN16-8)	85.11±2.46
	Full KD (WRN16-3)	84.31±2.24
	Full KD (WRN16-8)	83.70±2.52
	AT (WRN16-1)	83.79±2.40
	AT (WRN16-3)	84.44±2.22
	SP (WRN16-1)	84.31±2.38
	SP (WRN16-3)	84.89±2.10
TS+PImage	AVER (WRN16-1)	85.82±2.16
	AVER (WRN16-3)	86.00±2.45
	EBKD (WRN16-1)	85.58±2.31
	EBKD (WRN16-3)	85.62±2.37
	CA-MKD (WRN16-1)	84.06±2.50
	CA-MKD (WRN16-3)	85.02±2.64
	Base (WRN16-1)	85.91±2.32
	Base (WRN16-3)	86.18±2.37
	Ann. (WRN16-1)	86.05±2.23
	Ann. (WRN16-3)	86.74±2.25
	CADTP (w/ Ent.) (WRN16-1)	87.11±2.04
	CADTP (w/ Ent.) (WRN16-3)	**88.14**±2.07

outperforms the baselines. As a result, the proposed method improves performance while also allowing for effective model compression.

Table 5.6: Accuracy (%) with Various Knowledge Distillation Methods for Different Structure of Teachers on GENEActiv.

Method	Architecture Difference											
	Depth				Width				Depth+Width			
Teacher1 (1D CNNs)	WRN 16-1 (0.06M, 67.66)	WRN 16-1 (0.06M, 67.66)	WRN 28-1 (0.1M, 68.63)	WRN 40-1 (0.2M, 69.05)	WRN 16-1 (0.06M, 67.66)	WRN 16-3 (0.5M, 68.89)	WRN 28-1 (0.1M, 68.63)	WRN 28-3 (1.1M, 69.23)	WRN 28-1 (0.1M, 68.63)	WRN 28-3 (1.1M, 69.23)	WRN 40-1 (0.2M, 69.05)	WRN 16-1 (0.06M, 67.66)
Teacher2 (2D CNNs)	WRN 28-1 (0.4M, 59.45)	WRN 40-1 (0.6M, 59.67)	WRN 16-1 (0.2M, 58.64)	WRN 16-1 (0.2M, 58.64)	WRN 16-3 (1.6M, 59.80)	WRN 16-1 (0.2M, 58.64)	WRN 28-3 (3.3M, 59.69)	WRN 28-1 (0.4M, 59.45)	WRN 16-3 (1.6M, 59.80)	WRN 40-1 (0.6M, 59.67)	WRN 28-3 (3.3M, 59.69)	WRN 28-3 (3.3M, 59.69)
Student (1D CNNs)	WRN16-1 (0.06M, 67.66±0.45)											
AVER	68.71 ±0.42	68.38 ±0.53	68.66 ±0.26	68.76 ±0.38	68.92 ±0.09	67.98 ±0.29	67.89 ±0.23	68.91 ±0.24	68.29 ±0.16	69.10 ±0.57	69.10 ±0.43	68.07 ±0.27
Ann.	69.78 ±0.06	69.84 ±0.10	70.27 ±0.08	70.23 ±0.14	69.55 ±0.06	70.47 ±0.07	70.02 ±0.10	69.71 ±0.07	70.22 ±0.09	70.06 ±0.20	70.04 ±0.32	69.65 ±0.07
Ann.+Ba.	70.48 ±0.18	71.23 ±0.32	70.28 ±0.25	**71.07** ±0.33	69.47 ±0.27	70.98 ±0.11	70.27 ±0.45	70.49 ±0.64	70.00 ±0.19	71.30 ±0.07	**71.20** ±0.37	70.82 ±0.21
CADTP (w/ Ent.)	**72.17** ±0.06	**71.85** ±0.25	**70.84** ±0.13	70.47 ±0.27	**72.04** ±0.26	**72.23** ±0.54	**70.87** ±0.29	**71.75** ±0.07	**70.76** ±0.26	**71.93** ±0.13	70.87 ±0.31	**71.56** ±0.15
(η)	(900)	(700)	(700)	(700)	(700)	(500)	(700)	(500)	(700)	(700)	(700)	(700)

Table 5.7: Accuracy (%) with Various Knowledge Distillation Methods for Different Structure of Teachers on PAMAP2.

Method	Architecture Difference					
	Depth		Width	Depth+Width		
Teacher1 (1D CNNs)	WRN 28-1 (0.1M, 84.81)	WRN 16-1 (0.06M, 85.27)	WRN 28-3 (1.1M, 84.46)	WRN 16-3 (0.5M, 85.80)	WRN 16-1 (0.06M, 85.27)	WRN 28-3 (1.1M, 84.46)
Teacher2 (2D CNNs)	WRN 16-1 (0.2M, 86.93)	WRN 28-1 (0.4M, 87.45)	WRN 28-1 (0.4M, 87.45)	WRN 28-1 (0.4M, 87.45)	WRN 28-3 (3.3M, 87.88)	WRN 16-1 (0.2M, 86.93)
Student (1D CNNs)	WRN16-1 (0.06M, 82.99±2.50)					
Ann.	85.44 ±2.47	85.84 ±2.29	85.89 ±2.32	85.98 ±2.29	85.86 ±2.31	85.91 ±2.42
CADTP (w/ Ent.)	**85.89** ±2.46	**87.03** ±2.03	**87.11** ±2.40	**87.31** ±2.10	**87.57** ±1.97	**86.98** ±2.41

5.4.3 Various Combinations of Teachers

To explore the effects of different architectures for teachers, various different depth and width of WRNs are used, as described in Table 5.6 and 5.7. For GENEActiv, γ_c of (Teacher1, Teacher2) is 0.07 for (WRN16-3, WRN16-1) and (WRN28-3, WRN40-1), otherwise, the value is 0.2. As depicted in Table 5.6, CADTP produces the best student in almost all cases. When depth of Teacher1 is larger than Teacher2, Ann.+Ba. can generate a better student. For PAMAP2, η is 200 and γ_c of (Teacher1,

Teacher2) is 0.02 for (WRN28-1, WRN16-1), (WRN16-1, WRN28-1), and (WRN28-1, WRN16-3), otherwise, the value is 0.2. As shown in Table 5.7, CADTP shows the better results than Ann. in all cases. Both tables also show that in most cases CADTP performs better when Teacher1 has a smaller or the same depth of model than Teacher2 (e.g. WRN16-1 Teacher1 and WRN16-3 Teacher2). In some cases, Ann.+Ba. does not show much improvement, compared to the other baselines, while CADTP still shows good performance. In distillation with multiple teachers, even though the performance can be affected by the knowledge difference, CADTP alleviates the negative effect, and even produces a better student than its teachers. These findings also support the notion that having larger teachers is not always a good way to improve student performance (Cho and Hariharan (2019)).

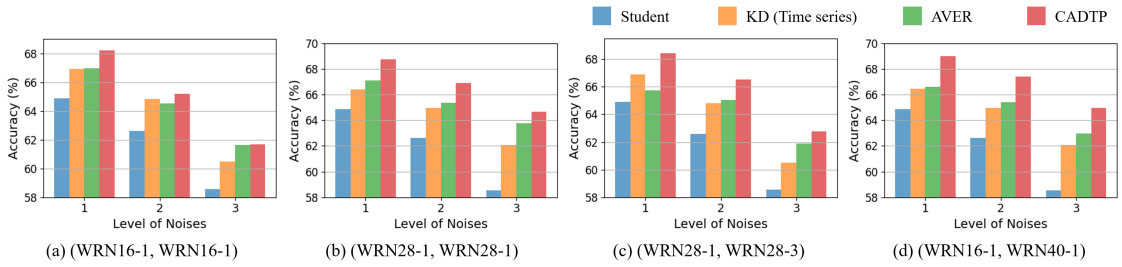
5.4.4 Ablations and Sensitivity Analysis

In this section, I explore the sensitivity of the proposed method. I evaluate CADTP under different settings of corruptions to figure out its ability to withstand noise. To better understand the performance, I investigate the effects of hyperparameters and visualize feature maps. Also, I analyze the generalizability of models.

Analysis of Invariance from Noise

To investigate the ability of models to be robust to different types of noise, I conducted experiments with noisy testing data by injecting continuous missing and Gaussian noise (Jeon *et al.* (2022b); Wen *et al.* (2021a); Wang and Wang (2019)). To account for unknown noise models, noise parameters are determined at random; (κ_R, σ_G) denotes (the percentage of the window size to be removed, the standard deviation for Gaussian noise). The exact parameters are chosen randomly and are less than the defined values. Both noises are applied simultaneously, and the variations are set as

three levels: Level 1 (0.15, 0.06), Level 2 (0.22, 0.09), and Level 3 (0.30, 0.12). Note, the classifiers were trained with the original training set.



(a) (WRN16-1, WRN16-1) (b) (WRN28-1, WRN28-1) (c) (WRN28-1, WRN28-3) (d) (WRN16-1, WRN40-1)

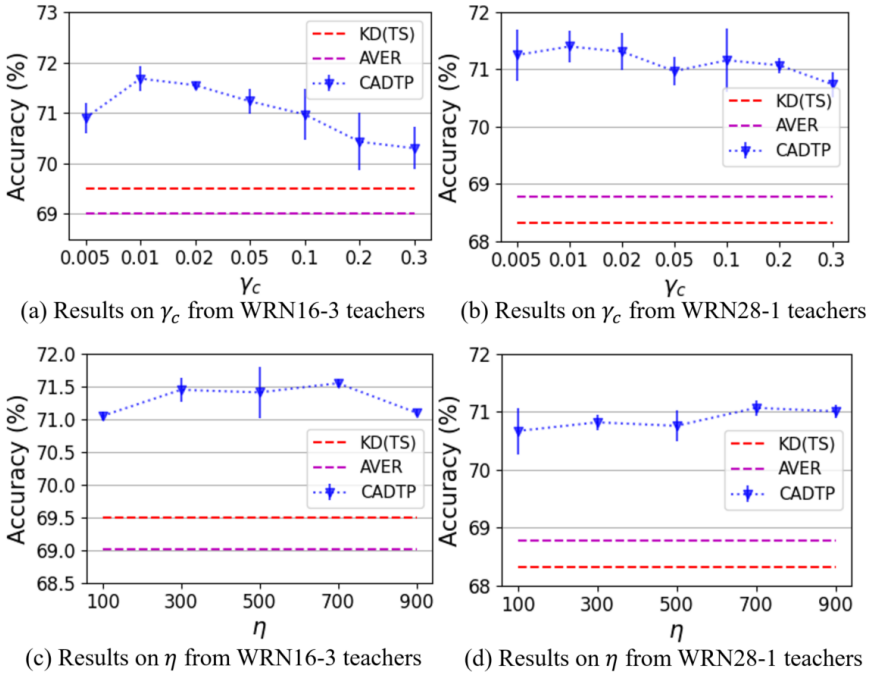
Figure 5.5: Accuracy (%) with Various Knowledge Distillation Methods for Different Noise Severity Levels on GENEActiv. Brackets Denote (Teacher1, Teacher2). Students Are WRN16-1 (1D CNNs).

As illustrated in Figure 5.5, CADTP (with Ent.) shows better performance than baselines in all cases. In most of the cases, student models by AVER perform better than the one from KD trained with time-series data alone, which implies that topological features complement features from the raw time-series data and help improve the robustness to noise. When Teacher1 and Teacher2 have different depth or width, the gap between CADTP and AVER is large. When the capacity or structure between teachers is different, knowledge transfer is more difficult. Thus, CADTP helps a student get beneficial features and improves noise robustness.

Effect of Distillation Hyperparameters on CADTP

γ_c and η are major components of the proposed method to balance the losses for batch and channel similarity maps in distillation. To investigate the sensitivity for these hyperparameters, I conduct experiments with various parameters.

A student (WRN16-1) is trained with two teachers by using different γ_c and η, as illustrated in Figure 5.6. For (a) and (b), the other hyperparameters are set as in



(a) Results on γ_c from WRN16-3 teachers

(b) Results on γ_c from WRN28-1 teachers

(c) Results on η from WRN16-3 teachers

(d) Results on η from WRN28-1 teachers

Figure 5.6: Sensitivity to γ_c and η of the Proposed Method for WRN16-1 Students on GENEActiv.

the previous section. All results of CADTP (with entropy based constrained adaptive distillation) outperform baselines. Their best is shown near $\gamma_c = 0.01$. For PAMAP2, their best also are shown the similar. The results with various η are presented in (c) and (d) with $\gamma_c = 0.02$. The best results are shown when $\eta = 700$. For PAMAP2, smaller number of η (200) shows the best. When the window size is small and the number of channels is large, small η (≤ 500) can be more effective. As shown in these results, to obtain the best result, setting the proper hyperparameters of γ_c and η is important.

Analysis of Constrained Adaptive Distillation

To consider the different feature properties of multiple teachers, the proposed method uses constrained adaptive weights based on entropy. To investigate the effects of

the constrained adaptive distillation, I compare the results between those with and without constraints.

Figure 5.7 shows the averaged probability by logits from models for testing samples of class 0 (Walking (treadmill at 1mph, 0% grade)) on GENEActiv, which are trained with time-series and persistence images. Since two models create completely different distributions, the difference in the ratio of entropy values between the two models is very large.

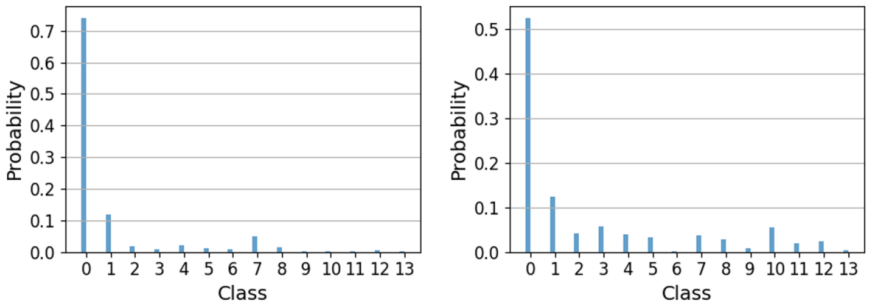


Figure 5.7: Probability Distributions for Models Trained with Different Modalities. Testing Samples of Class 0 Are Used to Measure the Probability.

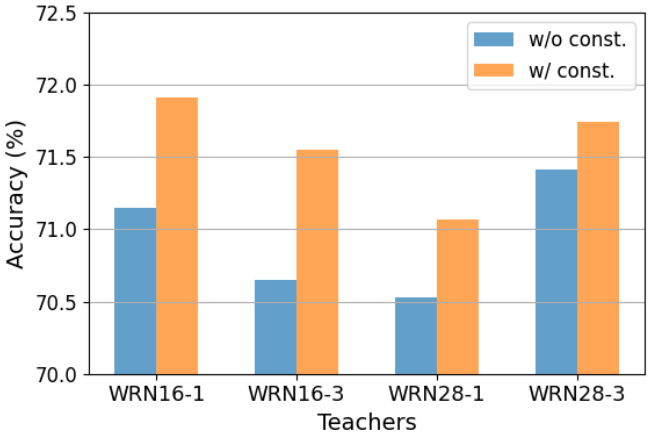


Figure 5.8: Accuracy (%) of the Proposed Method with or Without Constraints on GENEActiv. Students Are WRN16-1 (1D CNNs). "Const." Denotes Constraints.

Evaluation results for training with or without constraints based on entropy are illustrated in Figure 5.8. γ_c is 0.2 for WRN16-1 and WRN28-1 teachers and 0.02 for WRN16-3 and WRN28-3 teachers, respectively. As shown in these results, models trained with constraints perform better than the ones without constraints in all cases. This implies that features contain significant meaningful properties for performance improvements not only when entropy is low but also when it is high. Thus, the constraints empower the student to learn adequate knowledge from different modalities.

Visualization of Feature Maps

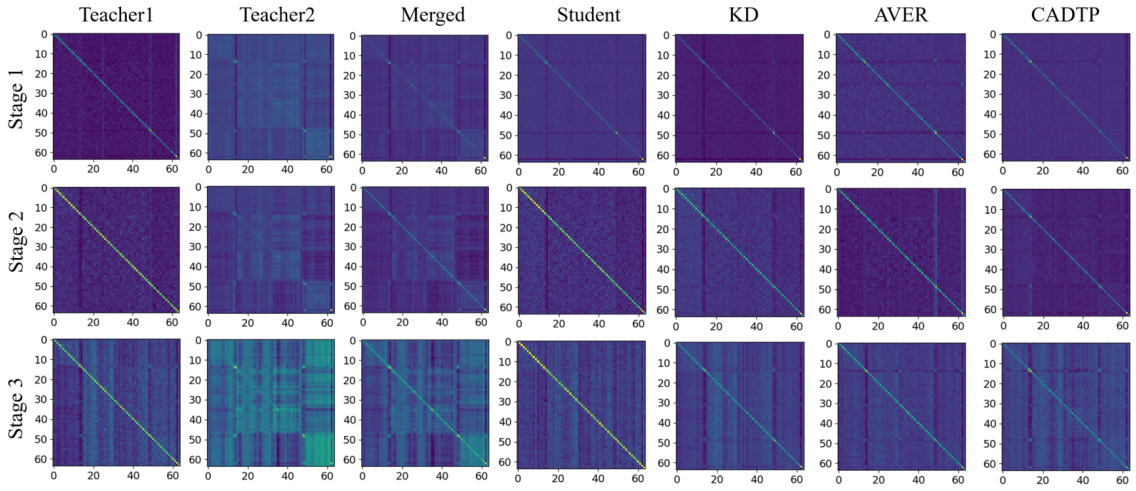


Figure 5.9: Activation Batch Similarity Maps Produced by a Layer for the Indicated Stage of the Network for a Batch on GENEActiv. High Similarities for Samples of the Batch Are Represented with High Values.

To figure out more details of activations for batch and channel similarities, both maps from teachers (WRN16-3) and a student (WRN16-1) are visualized in Figure 5.9 and 5.10, highlighting similarity with high values for input samples. A student by CADTP is trained with entropy based constrained adaptive distillation. A student of KD is trained with time-series data. Student is the result of a model trained from

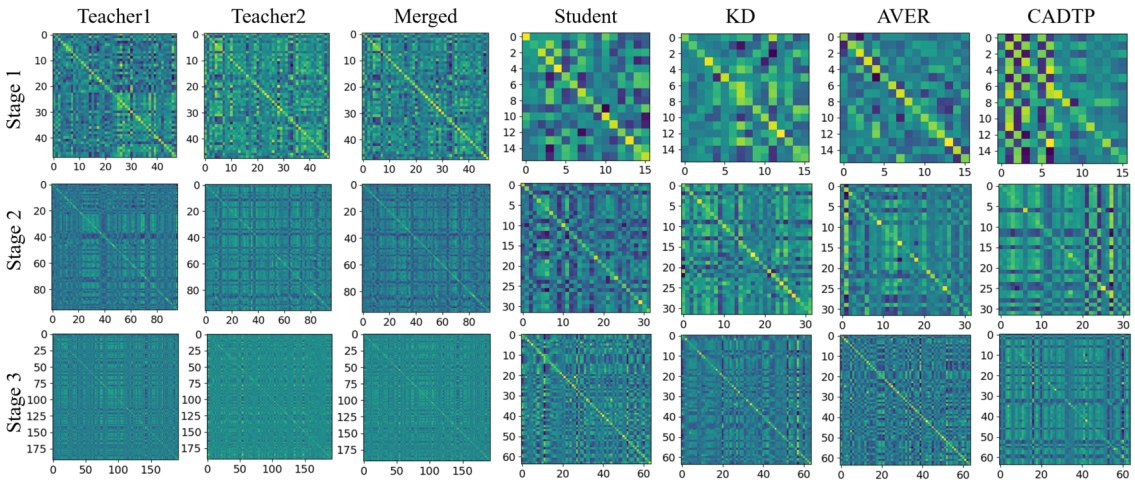


Figure 5.10: Activation Channel Similarity Maps Produced by a Layer for the Indicated Stage of the Network for a Batch on GENEActiv. High Similarities for Samples of the Batch Are Represented with High Values.

scratch. The merged map is generated with constrained α by entropy of two teachers. The maps of two teachers are dissimilar, and the merged map is also different from the student, implying the knowledge gap between them. For batch similarity, intuitively, the blockwise patterns are more prominent for the model (Teacher2) trained with PIs, compared to the one (Teacher1) with time-series data. For channel similarity, the maps from models trained with time-series data and persistence images show contrast in some rows and columns differently. Furthermore, batch and channel maps show large differences, implying that they can convey various types of information. Thus, these can contain a variety of knowledge for the dataset, and it is very important to transfer this knowledge well to students. The merged maps have characteristics of both Teacher1 and Teacher2. A student model trained with CADTP generates maps that show more contrastive patterns compared to baselines, representing blockwise patterns for batch similarity and row or column wise patterns

for channel similarity. This suggests that the proposed method helps a student learn diverse desirable features from different modalities.

Analysis of Model Reliability

To explore the generalizability and regularization effects, I calculated the expected calibration error (ECE) (Guo *et al.* (2017)) and negative log likelihood (NLL) (Guo *et al.* (2017)). ECE is to measure calibration error, which represents the reliability of the model. The probabilistic quality of a model can be computed by NLL. I used students trained by teachers of WRN16-3 and WRN28-1. ECE and NLL with various methods on GENEActiv and PAMAP2 are shown in Table 5.8 and 5.9, respectively. In both cases, the results of AVER outperform KD and a model learned from scratch (Student). This implies that using topological features improves generalizability. CADTP (with Ent.) generates the lowest ECE and NLL in almost all cases. Thus, utilizing topological features in distillation improves the performance, not only for accuracy but also for reliability. Finally, the proposed method aids in generating a better student model.

Table 5.8: ECE (%) and NLL for Various Knowledge Distillation Methods on GENEActiv. Teachers Are WRN16-3 and WRN28-1. Students Are WRN16-1 (1D CNNs).

Method	WRN16-3		WRN28-1	
	ECE	NLL	ECE	NLL
Student	3.548	2.067	3.548	2.067
KD	3.200	1.520	3.064	1.512
AVER	2.940	1.220	2.845	1.148
CADTP (w/o Ent.)	2.665	1.080	**2.661**	1.067
CADTP (w/ Ent.)	**2.625**	**0.991**	2.744	**1.016**

Table 5.9: ECE (%) and NLL for Various Knowledge Distillation Methods on PAMAP2. Teachers Are WRN16-3 and WRN28-1. Students Are WRN16-1 (1D CNNs).

Method	WRN16-3		WRN28-1	
	ECE	NLL	ECE	NLL
Student	2.299	1.287	2.299	1.287
KD	2.183	1.061	2.323	1.329
AVER	2.174	0.910	2.263	1.122
CADTP (w/o Ent.)	2.014	0.932	1.951	0.954
CADTP (w/ Ent.)	**1.630**	**0.793**	**1.729**	**0.779**

5.4.5 Computational Time

I measured the computational time of various methods for testing set on GENEActiv. The models were run on a desktop with a 3.50 GHz CPU (Intel® Xeon(R) CPU E5-1650 v3), 48 GB memory, and an NVIDIA TITAN Xp graphic card (3840 NVIDIA® CUDA® cores and 12 GB memory) (NVIDIA (2016)). I evaluated approximately 6k samples with a batch size of 1. In Table 5.10, the considered accuracy is the best one from Table 5.2 and 5.6. Since generating PIs by TDA is implemented on the CPU, a model trained from scratch with PIs takes the largest amount of time in the table. A WRN16-1 (1D CNNs) student from CADTP takes the lowest time with the best accuracy. The model takes 2.89 ms in averaged time on CPU. If a smaller network is used as a student or a smaller sample window of data is used, it takes much less time. The CPU result further highlight why a model compression

method such as KD is needed for running on small devices with limited power and computational resources.

Table 5.10: Processing Time of Various Models on GENEActiv.

Model	Learning from scratch		KD		CADTP (w/ Ent.)
	TS (1D)	PImage (2D)	TS	PImage	TS+PImage
	WRN28-3	WRN16-3	WRN16-1 (1D CNNs)		
Accuracy (%)	69.23	59.8	69.71	68.76	**72.23**
GPU (sec)	29.94	356.92 (PIs on CPU) +13.63 (model)	**15.23**		
CPU (sec)	1977.89	356.92 (PIs on CPU) +11191.45 (model)	**16.66**		

5.5 Conclusion

In this chapter, I proposed a new framework for constrained adaptive knowledge distillation using topological representations on wearable sensor data, utilizing various similarity features and an annealing strategy. I demonstrated the proposed method, CADTP, with various combinations of teachers and the student in classification. I also analyzed the effectiveness of CADTP with experiments on invariance from noise and feature map visualization. The proposed method showed robust performance in classification and efficiency, which is better than baselines and important in various applications needing implementations on small devices. In future work, the proposed method can include more diverse teachers, which are learned with different representations, such as Gramian Angular Fields (GAF) and Markov Transition Fields (MTF) based images encoded by time-series data. Also, I would like to investigate the effects

of augmentation methods on the image representations to leverage multiple teachers in KD.

Chapter 6

UNCERTAIN FEATIRE RECTIFICATION FOR TOPOLOGICAL KNOWLEDGE DISTILLATION ON WEARABLE SENSOR DATA

6.1 Introduction

Wearable sensor data integrated with machine learning techniques has led to increasing applications in many different application involving human activity modeling and tracking (Nweke *et al.* (2018)). However, developing robust sensor time-series models is still considered challenging because of sensor-level noise (Wang *et al.* (2021)), varying sampling rates (Adams *et al.* (2017)), and inter- and intra-person variability (Cho *et al.* (2021)). To overcome these issues, topological data analysis (TDA) has shown promise and has shown to aid in improving performance for time-series classification (Seversky *et al.* (2016)). Specifically, persistence images (PIs) generated by TDA are stable to signal perturbations (Adams *et al.* (2017); Edelsbrunner and Harer (2022)), which are 2D image representations obtained from the raw time series data. However, the process of TDA to extract PIs requires a large amount of time and computational resources, which makes it difficult to run on small devices with limited computational power. Furthermore, features from TDA and the raw time series data are difficult to integrate and generate a unified model with machine learning because they have heterogeneous characteristics, including dimension sizes of features and statistical characteristics.

To mitigate these problems, I adopt knowledge distillation (KD), to generate a small model (student) from a large model (teacher). KD is effective in wearable sensor data analysis (Chen *et al.* (2018); Cheng *et al.* (2023); Jeon *et al.* (2022b); Gou

et al. (2021)) and can provide multimodal data analysis solutions leveraging multiple teachers or students (Gou *et al.* (2021); Zhang *et al.* (2022)). I use multiple teachers trained with both time-series and PIs to distill a single student. The single student is run with only time-series data as an input. That is, the PI and TDA processes are not incorporated into test time. However, there are significant difficulties in utilizing different teachers in KD: (1) The teachers have common and different characteristics simultaneously, so integrating features from multiple teachers and implementing them in a unified framework are challenging. (2) The teachers are not always perfect and can transfer improper knowledge to the student, which degrades performance.

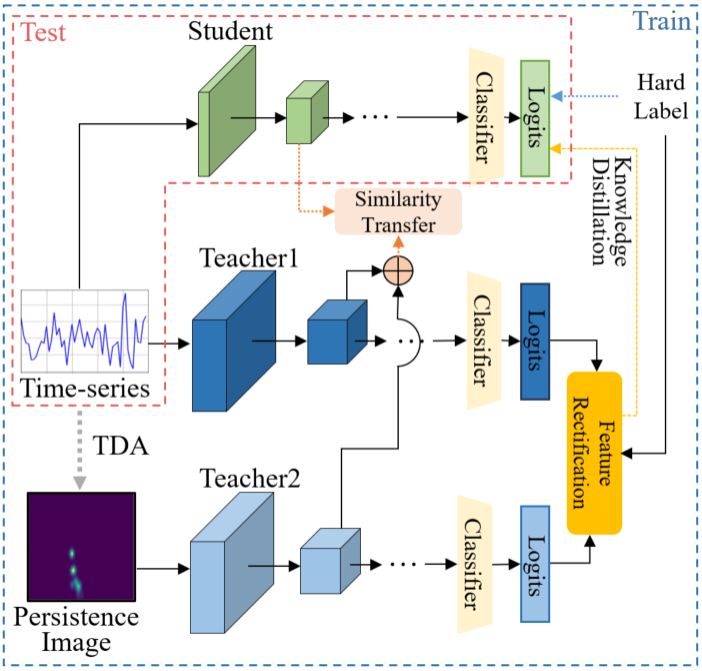


Figure 6.1: An Overview of the Proposed Method.

In this chapter, I build upon a framework that utilizes multimodal inputs in KD learning process, including two different teachers and a single student, recently explored in the context of distilling topological features with raw time-series data (Jeon *et al.* (2022b)). Firstly, PIs are obtained from persistence diagrams produced by

TDA. I train two models with time-series data and PIs, respectively. Secondly, the trained models are utilized as teachers in KD to train a student. To transfer knowledge, logits from teachers are utilized. Even though two teachers are trained with different inputs, their tasks are the same as classification. In this light, their outputs have common or different characteristics. In addition, both teachers are not always perfect and may transfer noisy information. To consider the knowledge discrepancy and uncertainty of two teachers, I develop a new mechanism with uncertainty-aware feature rectification for distillation. An overview of the proposed method is depicted in Figure 6.1.

In more detail, common and different characteristics from two teachers are separated and weighted differently. When one teacher has lower confidence than the other, the teacher's output is rectified by placing more weight on common features and less on the teacher's inherent features. This provides strong supervision and encourages a student to gain more confident knowledge. In the third step, to provide more knowledge in learning process, correlation maps from intermediate features within a mini-batch are used, which have the benefit of matching different structural knowledge. Finally, I find that the proposed method distills a single student model that outperforms several baselines using single or multiple teachers in KD. I demonstrate the effectiveness of the proposed method with empirical analysis and different sizes of datasets.

6.2 Background

6.2.1 Topological Feature Extraction

Persistent homology is a key representation used in TDA, which tracks the variations of n-dimensional holes characterized by a dynamic thresholding process, a

filtration (Edelsbrunner *et al.* (2002)). During the filtration process, the persistence of holes implies a persistence feature, which is projected onto the persistence diagram (PD) encoding the birth and death times as x and y coordinates of planar scatter points (Adams *et al.* (2017); Edelsbrunner and Harer (2022)). Since the location and number of points in the PD are not fixed, PDs are vectorized in many different ways (Ali *et al.* (2023)), including commonly as persistence images (PI) (Edelsbrunner and Harer (2022)). A given PD is converted to a persistence surface (PS) computed by a weighted sum of Gaussian functions centered at the scatter points in the PD. The PS is discretized which as depicted in Figure 6.2. Topological features can complement time-series features to improve performance (Som *et al.* (2020)). However, TDA requires large computational power and time consumption, making it challenging to apply the method to small devices with limited computational resources. I propose a framework based on knowledge distillation to distill a small model to obtain better performance.

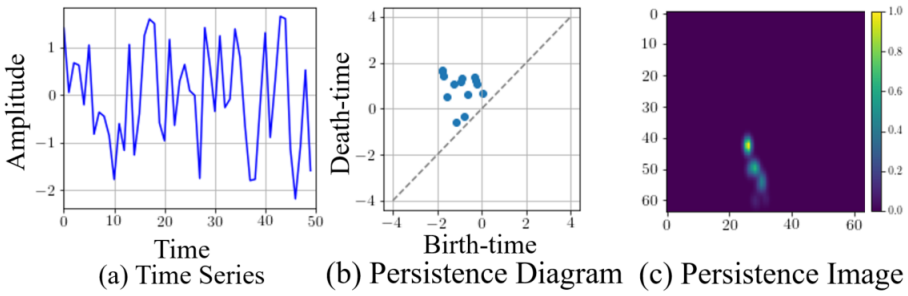


Figure 6.2: Example of Time-series and Its Corresponding PD and PI.

6.2.2 Knowledge Distillation

Traditional KD aims to train a simple network by using a larger or more complex network. To transfer knowledge, both hard and soft labels are utilized. In KD, a

student is trained with the loss function as follows:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{\mathcal{CE}} + \lambda\mathcal{L}_{\mathcal{KD}}, \tag{6.1}$$

where $\mathcal{L}_{\mathcal{CE}}$ is the standard cross entropy loss, $\mathcal{L}_{\mathcal{KD}}$ is KD loss, and λ is a hyper-parameter; $0 < \lambda < 1$. The cross entropy loss to train a student is:

$$\mathcal{L}_{\mathcal{CE}} = \mathcal{H}(\mathcal{Q}(l_S), t), \tag{6.2}$$

where $\mathcal{Q}(\cdot)$ is a softmax function, $\mathcal{H}(\cdot)$ is a cross entropy loss function, l_S is the logits of a student, and t is a ground truth label. The difference of softened outputs between student and teacher are minimized by KL-divergence loss:

$$\mathcal{L}_{\mathcal{KD}} = \tau^2 KL(h_T, h_S), \tag{6.3}$$

where, $h_T = \mathcal{Q}(l_T/\tau)$ is a softened output of a teacher network, $h_S = \mathcal{Q}(l_S/\tau)$ is a softened output of a student, and τ is a hyper-parameter; $\tau > 1$. To get the best performance, in this chapter, early stopped model of a teacher (ESKD) is utilized, improving the efficacy of KD (Cho and Hariharan (2019); Jeon *et al.* (2022b)).

For better mimicking the teacher, representations from intermediate layers can be leveraged in knowledge transfer. There are many different variants of using layer-to-layer and sample-to-sample relationships for KD. Attention maps are computed for knowledge transfer by a sum of squared attention mapping function (AT) (Zagoruyko and Kmodakis (2017)). Tung *et al.* (Tung and Mori (2019)) suggested calculating and comparing the similarity matrix within a mini-batch of a teacher and student to mitigate the difference. The similarity map $M \in \mathbb{R}^{b\times b}$ is generated as follows:

$$M = F \cdot F^{\top}; F \in \mathbb{R}^{b\times chw}, \tag{6.4}$$

where F is reshaped features from an intermediate layer of a model, b is the size of a mini-batch, c is the number of output channels, and h and w are the height and

width of the output, respectively. These methods are widely explored; however, most of them are unimodal solutions that use a single teacher.

Recently, multiple teachers have been employed, and feature integration methods have been addressed for knowledge transfer (Gou *et al.* (2021); You *et al.* (2017); Zhang *et al.* (2022)). However, these methods generally deal with unimodal problems. Also, common and different characteristics between teachers are not considered, which have different effects on the training process. Even though less weight is applied to less confident features, noisy features can be transferred, which degrades the performance. Our proposed method measures uncertainty for two teachers and rectifies less confident features to preserve significant inter-class relationships and obtain more effective knowledge for stronger supervision.

6.3 Methodology

6.3.1 Persistence Image Extraction

To utilize topological features in KD, I extract PIs to train a model. By referring to a previous study (Som *et al.* (2020)), Scikit-TDA python library (Saul and Tralie (2019)) and the Ripser package are used to compute PDs. Each channel of a sample is projected to the PD generating PI by its the birth-time vs. lifetime information. I set the grid size of a PI as 64×64 so that the dimension size of a PI for a sample is $64 \times 64 \times c$. I then train a model with the extracted PIs, and the model is used as a teacher in KD for transferring topological features to a student model.

6.3.2 Logit Transfer for Multiple Teachers

For the proposed method, logit knowledge from two teachers is transferred separately so that no additional layers or concatenation procedures are needed. The logit

knowledge transfer loss can be written as:

$$\mathcal{L}_{\mathcal{KD}l} = \tau^2 \left(\alpha KL(h_{T_1}, h_S) + (1-\alpha) KL(h_{T_2}, h_S)\right), \tag{6.5}$$

where α is a constant value to balance the effects from two teachers, and h_{T_1} and h_{T_2} are softened outputs of teachers learned by time-series data and PIs, respectively.

6.3.3 Uncertainty-aware Feature Rectification

Feature Separation: Multiple teachers trained with different inputs and structures generate different statistical characteristics. However, since two models are trained to deal with the same task for classification, they also have common characteristics. To reduce noisy features and boost the effectiveness of KD, as shown in Figure 6.3, I separate features into three different categories: Teacher1's inherent (q_{T_1}), common (q_f), and Teacher2's inherent (q_{T_2}) features. Inherent features of Teacher1 and Teacher2 (q_{T_1} and q_{T_2}) are obtained by $max(h_{T_1} - h_{T_2}, 0)$ and $max(h_{T_2} - h_{T_1}, 0)$, respectively. The common feature q_f is calculated by $h_{T_1} - q_{T_1}$. I apply weights to three features differently to transfer knowledge effectively.

Feature Rectification: Two teachers are not always guaranteed to provide high-quality and proper knowledge for better performance. To generate stronger knowledge, I utilize confidence scores measured by cross-entropy loss with teachers. Features of teachers are rectified as follows:

$$T_{ij} = \begin{cases} T_{1^j} & \text{if } \mathcal{H}(\mathcal{Q}(l_{T_{1^j}}), t^j) > \mathcal{H}(\mathcal{Q}(l_{T_{2^j}}), t^j) \\ T_{2^j} & \text{otherwise} \end{cases}; \tag{6.6}$$

$$h_{T_{ij}} = \beta_1 q_f^{ij} + \beta_2 q_{T_{ij}},$$

where i indicates a teacher among Teacher1 and 2, j is a number for a sample, and β_1 and β_2 are the hyper-parameters. When both β_1 and β_2 are 1.0, this corresponds to without using feature rectification. When β_2 is lower than β_1, the effect of a teacher's

inherent feature is less than the common feature. After rectification, updated h_{T_1} and h_{T_2} are utilized in equation 6.5 to calculate $\mathcal{L}_{\mathcal{KD}l}$. In this way, teachers transfer less noisy and higher quality knowledge, and the student is learned with stronger supervision.

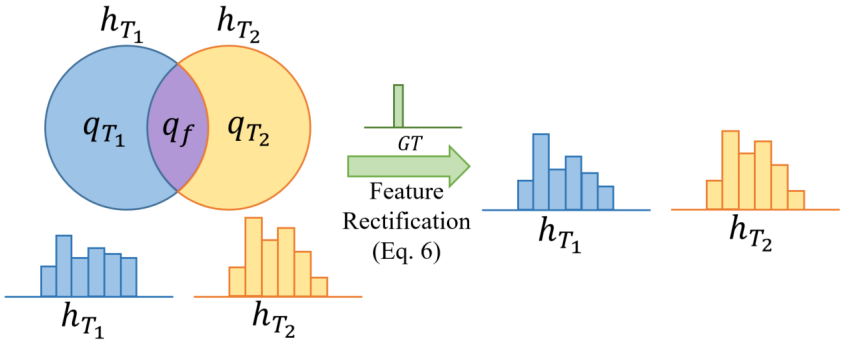


Figure 6.3: Illustration of a Mechanism of Uncertainty-aware Feature Rectification When Cross-entropy Loss from Teacher1 Is Higher than the One from Teacher2.

6.3.4 Knowledge Distillation with Multiple Teachers

For further improvement, I utilize a similarity map to integrate different sizes of features obtained from two teachers with different structures (1D CNNs and 2D CNNs). The size of similarity maps extracted from two teachers within a mini-batch is the same regardless of model structures. I use weighted summation to integrate similarity maps of teachers and transfer the knowledge to a student, which is computed by:

$$M_T^{(l)} = \alpha M_{T_1}^{(l^{T_1})} + (1-\alpha) M_{T_2}^{(l^{T_2})}; \widetilde{M_T^{(l)}} = M_T^{(l)} / \left\| M_T^{(l)} \right\|_2 \tag{6.7}$$

where $M_T^{(l)} \in \mathbb{R}^{b \times b}$ is the merged map from the outputs of a layer pair (l^{T_1} and l^{T_2}) of two teachers M_{T_1} and M_{T_2}. Student tries to mimic teachers by the following loss function:

$$\mathcal{L}_m = \frac{1}{b^2|L|} \sum_{(l,l^S)\in L} \left(\left\| \widetilde{M_T^{(l)}} - \widetilde{M_S^{(l^S)}} \right\|_F^2 \right), \tag{6.8}$$

where L accumulates the layer pairs (l and l^S), $\widetilde{M_S^{(l^S)}}$ is normalized map for a student, and $\|\cdot\|_F$ is the Frobenius norm (Tung and Mori (2019)). To consider a student using only time-series data that is different from teachers, I leverage an annealing strategy that is addressed in a prior study Jeon *et al.* (2022a), which uses weights of a model trained from scratch when a student model is initialized for training. Then, the gap between teachers and a student is reduced, and the student can preserve its inherent characteristics that aid in improving performance.

The final loss function can be written as follows:

$$\mathcal{L}_{\text{total}} = (1-\lambda)\mathcal{L}_{\mathcal{CE}} + \lambda\mathcal{L}_{\mathcal{KD}l} + \kappa\mathcal{L}_m, \tag{6.9}$$

where κ is a hyper-parameter.

6.4 Experiments

Dataset Description: I evaluate the proposed method with two different datasets. GENEActiv (Jeon *et al.* (2022b)) is collected with a light-weight, waterproof, and wrist-worn tri-axial accelerometer. The dataset is comprised of 14 daily activities, as in (Jeon *et al.* (2022b)) with full non-overlapping window size of 500 time-steps. PAMAP2 (Reiss and Stricker (2012)) was recorded from the heart rate and 4 IMUs for 9 subjects with 12 daily activities, consisting of 40 channels. The window size of a sample is 100 time-steps (3 seconds). The evaluation protocol for this dataset is leave-one-subject-out.

Implementation Details: To generate PIs, parameters in PD are set to 0.25 and 0.015, and the birth-time range for PI is [-10, 10] and [-1, 1] for GENEActiv and PAMAP2, respectively, referring to the previous study (Som *et al.* (2020)). To train models, I set the total number of epochs as 200 with 64 as the batch size, using SGD with momentum of 0.9 and 1×10^{-4} for a weight decay. A model (1D CNNs) for

Table 6.1: Accuracy (%) with Various Knowledge Distillation Methods for Different Capacity of Teachers on GENEActiv.

Teacher1 (1D CNNs)		WRN16-1 (0.06M, 67.66)	WRN16-3 (0.5M, 68.89)	WRN28-1 (0.1M, 68.63)	WRN28-3 (1.1M, 69.23)
Teacher2 (2D CNNs)		WRN16-1 (0.2M, 58.64)	WRN16-3 (1.6M, 59.80)	WRN28-1 (0.4M, 59.45)	WRN28-3 (3.3M, 59.69)
Student (1D CNNs)		WRN16-1 (0.06M, 67.66±0.45)			
PI	KD	67.83±0.17	68.76±0.73	68.51±0.01	68.46±0.28
Time-series	KD	69.71±0.38	69.50±0.10	68.32±0.63	68.58±0.66
	AT	68.21±0.64	69.79±0.36	68.09±0.24	67.73±0.27
	SP	67.20±0.36	67.85±0.24	68.71±0.46	67.39±0.49
	SimKD	69.39±0.18	69.89±0.11	68.92±0.40	68.80±0.38
	DIST	68.20±0.28	69.71±0.15	69.23±0.19	68.18±0.60
TS+PImage	AVER	68.99±0.76	68.74±0.35	68.77±0.70	69.02±0.50
	EBKD	68.43±0.25	69.24±0.25	68.45±0.73	67.50±0.40
	CA-MKD	69.33±0.61	69.80±0.16	69.61±0.57	68.81±0.79
	Base	69.09±0.37	69.24±0.62	69.55±0.41	69.42±0.58
	Ann	70.15±0.03	70.71±0.12	70.44±0.10	69.97±0.06
	Ours	**71.32** ±0.14	**71.18**±0.18	**71.42**±0.17	**70.90**±0.28

time-series data is trained with an initial learning rate of 0.05 that decreases by 0.2 at 10 epochs and by 0.1 every $[\frac{y}{3}]$ epoch where y is 200. A model (2D CNNs) for image data is trained with an initial learning rate of 0.1 that decreases by 0.5 at 10 epochs and by 0.2 every 40 epochs. For experiments, I utilize WideResNet (WRN) (Zagoruyko and Komodakis (2016)) which is popularly used for KD evaluation (Cho and Hariharan (2019); Jeon *et al.* (2022b)) and can construct different widths and depths of networks. I set τ and λ as 4 and 0.7 for GENEActiv, and as 4 and 0.99 for

Table 6.2: Accuracy (%) for Related Methods on GENEActiv with 7 Classes.

	Method	Window length	
		1000	500
Time-series	WRN16-1	89.29±0.32	86.83±0.15
	WRN16-3	89.53±0.15	87.95±0.25
	WRN16-8	89.31±0.21	87.29±0.17
	ESKD	89.88±0.07	88.16±0.15
	Full KD	89.84±0.21	87.05±0.19
	AT	90.32±0.09	87.60±0.22
	SP	88.47±0.19	87.69±0.18
	SimKD	90.47±0.32	88.16±0.37
	DIST	90.20±0.39	87.05±0.31
TS+PImage	AVER	90.06±0.33	87.05±0.37
	EBKD	89.82±0.14	87.66±0.28
	CA-MKD	90.13±0.34	88.04±0.26
	Ann	90.71±0.15	88.26±0.24
	Ours	**91.26**±0.15	**88.94**±0.11

PAMAP2, respectively, referring to the previous study (Jeon *et al.* (2022b)). α is 0.7 and 0.3 for GENEActiv and PAMAP2, respectively. β_1 and β_2 are set to 1.0 and 0.75, respectively. I run 3 times and the best averaged accuracy and standard deviation are reported. For baselines, traditional KD (Hinton *et al.* (2015)), AT (Zagoruyko and Kmodakis (2017)), SP (Tung and Mori (2019)), SimKD (Chen *et al.* (2022)), and DIST (Huang *et al.* (2022)) are used to compare with using a single teacher for KD. Also, I evaluate with methods using multi-teacher such as AVER (You *et al.* (2017)), EBKD (Kwon *et al.* (2020)), CA-MKD (Zhang *et al.* (2022)), Base (Jeon *et al.* (2022a)), and Ann (Jeon *et al.* (2022a)). Two teachers have different structures, so

only logits are utilized in KD for baselines. κ for GENEActiv is 700. For PAMAP2, 500 is for ours and 200 is for baselines to obtain the best result.

Table 6.3: Accuracy (%) for Related Methods on PAMAP2.

	Method	Accuracy (%)
Time-series	WRN16-1	82.81±2.51
	WRN16-3	84.18±2.28
	WRN16-8	83.39±2.26
	ESKD	86.38±2.25
	Full KD	84.31±2.24
	AT	84.44±2.22
	SP	84.89±2.10
TS+PImage	AVER	86.00±2.45
	EBKD	85.62±2.37
	CA-MKD	85.02±2.64
	Base	86.18±2.37
	Ann	87.12±2.26
	Ours	**87.21**±2.42

6.4.1 Results on Various Capacity of Teachers

I investigate the effectiveness of the proposed method on the capacity of teachers. Note, brackets denote the number of trainable parameters and accuracy for the model. As shown in Table 6.1, the proposed method outperforms all baselines, including a single teacher and multiple teacher-based KD. To evaluate the method on different window lengths and the number of classes, the method is implemented on 7 classes of GENEActiv dataset and PAMAP2 dataset, using WRN16-3 teachers. Note, TS denotes time-series. Students are WRN16-1. As described in Table 6.2 and 6.3,

ours show the best accuracy in all cases. Thus, the proposed method aids in model compression as well as improving classification accuracy.

6.4.2 Results on Different Combinations of Teachers

To explore the performance using teachers with different architectures, I set up several different combinations of teachers, considering the width and depth of networks. As described in Table 6.4 and 6.5, the proposed method outperforms baselines in all cases. In some cases, a student distilled by ours even outperforms its teachers. This implies that ours can guide a student effectively even when teachers have many differences and more knowledge gaps, which generate negative effects.

Table 6.4: Accuracy (%) with Various Knowledge Distillation Methods for Different Structure of Teachers on GENEActiv.

Method	Difference of Architecture						
	Depth		Width		Depth+Width		
Teacher1	WRN16-1	WRN28-1	WRN16-1	WRN16-3	WRN28-1	WRN40-1	WRN16-1
(1D CNNs)	(0.06M, 67.66)	(0.1M, 68.63)	(0.06M, 67.66)	(0.5M, 68.89)	(0.1M, 68.63)	(0.2M, 69.05)	(0.06M, 67.66)
Teacher2	WRN28-1	WRN16-1	WRN16-3	WRN16-1	WRN16-3	WRN28-3	WRN28-3
(2D CNNs)	(0.4M, 59.45)	(0.2M, 58.64)	(1.6M, 59.80)	(0.2M, 58.64)	(1.6M, 59.80)	(3.3M, 59.69)	(3.3M, 59.69)
Student	WRN16-1						
(1D CNNs)	(0.06M, 67.66±0.45)						
AVER	68.71±0.42	68.66±0.26	68.92±0.09	67.98±0.29	68.29±0.16	69.10±0.43	68.07±0.27
Ann	69.95±0.05	70.34±0.14	69.68±0.14	71.06±0.02	70.28±0.13	70.49±0.05	69.65±0.04
Ours	**70.66**±0.33	**71.52**±0.04	**70.28**±0.45	**71.42**±0.20	**70.83**±0.83	**71.35**±0.07	**71.18**±0.20

Table 6.5: Accuracy (%) with Various Knowledge Distillation Methods for Different Structure of Teachers on PAMAP2.

Method	Difference of Architecture				
	Depth	Width	Depth+Width		
Teacher1 (1D CNNs)	WRN28-1 (0.1M, 84.81)	WRN28-3 (1.1M, 84.46)	WRN16-3 (0.5M, 85.80)	WRN16-1 (0.06M, 85.27)	WRN28-3 (1.1M, 84.46)
Teacher2 (2D CNNs)	WRN16-1 (0.2M, 86.93)	WRN28-1 (0.4M, 87.45)	WRN28-1 (0.4M, 87.45)	WRN28-3 (3.3M, 87.88)	WRN16-1 (0.2M, 86.93)
Student (1D CNNs)	WRN16-1 (0.06M, 82.99$\pm$2.50)				
Ann	85.97$\pm$2.33	85.59$\pm$2.28	85.82$\pm$2.26	85.94$\pm$2.31	85.86$\pm$2.42
Ours	**87.38**$\pm$2.10	**88.30**$\pm$2.14	**88.14**$\pm$2.23	**87.71**$\pm$1.97	**86.80**$\pm$2.36

6.4.3 Ablation Study

Utilizing Similarity Features: I explore improvements in utilizing similarity maps from intermediate features for ours and a baseline implementing without feature rectification. As illustrated in Figure 6.4, utilizing similarity shows better performance for both methods. Also, ours with similarity features outperforms Ann (Jeon *et al.* (2022a)). This implies that intermediate features provide more information for teachers. Also, the uncertain feature rectification module can be combined with similarity maps in distillation to improve performance.

Feature Rectification Hyper-parameters: I evaluate the proposed method on different feature rectification hyper-parameters. I set β_1 to 1.0 and β_2 to different values to explore the sensitivity for the parameter. As described in Figure 6.5, all re-

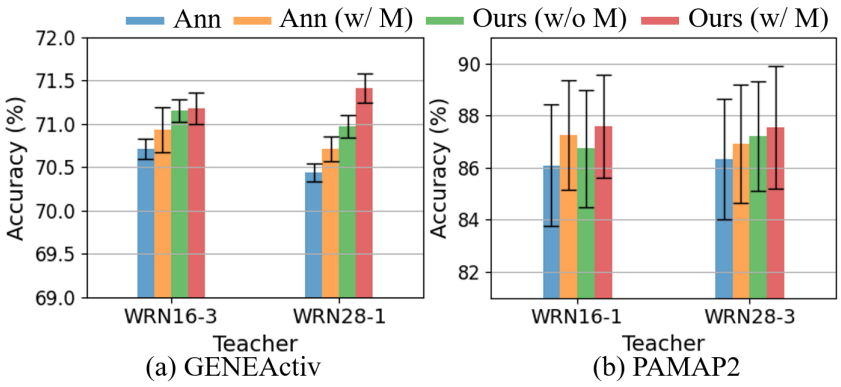


Figure 6.4: Accuracy (%) of Students Trained with or Without Using Similarity Map (M). Students Are WRN16-1.

sults of our method outperform baselines implementing without feature rectification, and the best is shown with 0.75 of β_2.

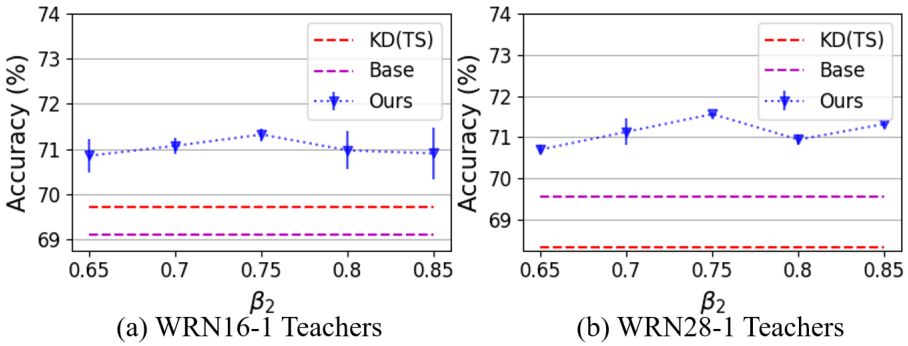


Figure 6.5: Accuracy (%) with Various Feature Rectification Parameters (β_2) on GENEActiv. Students Are WRN16-1.

6.5 Conclusion

In this chapter, I propose a novel approach based on uncertainty-aware feature rectification, for knowledge distillation with topological features. I showed the robustness of the proposed method through empirical evaluation using various capaci-

ties and combinations of teachers and students. Compared to baselines, our method shows improved performance.

Chapter 7

DISCUSSION AND FUTURE WORK

In this thesis, I discussed solutions in knowledge distillation with geometric approaches for multimodal data analysis to enable a small model to improve the efficiency and performance of classification.

In Chapter 2, I explored the role of augmentation strategies in KD with time-series data across different sizes and window lengths of datasets. Augmentation methods, which are relevant time-series data such as removal and Gaussian filtering, were applied to teachers and students with various combinations. I evaluated performance by using different capacities of teacher networks. I showed that a high-capacity teacher network does not necessarily ensure better performance in a student network. I further showed that training with augmentation methods and early stopping for KD (ESKD) are effective when dealing with time-series data. In most cases, when the augmentation method was applied to train a student and a teacher trained with original data was used, performance showed the best. Also, using a combination of augmentation methods in the KD training process showed better performance than utilizing a single augmentation strategy. Thus, I conclude that using a combination of augmentation methods for training KD can perform better in general.

In Chapter 3, I showed how a geometric approach can help to extract attentive knowledge for improving the performance of knowledge transfer. Leveraging an angular distribution for KD was proposed, which projects features onto the hypersphere and inserts an angular margin to enlarge the gap between positive and negative features. This aids in extracting better representations of features and training a student with more distinguished and disentangled features from a teacher. The proposed

method was evaluated with four public image datasets using various combinations of networks in classification as well as various aspects such as feature visualization, t-SNE metric, and generalizability. In overall cases, the proposed method outperformed baselines. Also, the method was evaluated for compatibility by combining existing methods. With augmentation methods as well as other distillation methods, performance with the proposed method was improved. This proved that the method can perform with various existing methods and generate synergetic effects, providing more informative features to improve performance.

In Chapter 4, utilizing topological features by a geometric method in KD using multiple teachers was proposed. With KD, two teachers were trained with multimodal data, including time-series and image representations. Image data, called persistence images, was generated by TDA, which has topological features that can significantly complement time-series features and have robustness in noises. To accommodate different teachers in one framework, an annealing strategy was introduced to reduce the knowledge gaps between networks and preserve the inherent beneficial characteristics of a student network implemented with time-series data only. To provide plentiful information to the student, relationships of similarities from intermediate features were leveraged, where the relationships representing more expressive features were computed by orthogonality properties within similarities. In this framework, a robust student was distilled, which uses time series data only as the input without requiring access to image representation from persistence features. The proposed method was evaluated with datasets having different numbers of classes and window lengths and showed better results than baselines. Further, the method outperformed baselines in model reliability and the ability of invariances from noises, which implies better model generalizability.

In Chapter 5, a constrained adaptive weighting mechanism was introduced to improve performance in the KD learning process using multimodal data. Two teachers were trained with different data, including time-series and persistence image data, which were addressed in Chapter 4. To provide more plentiful and informative features, both batch and channel similarities within a mini-batch were leveraged for knowledge transfer. To control the effects of different teachers, a constrained weighting mechanism based on entropy values was developed. Through feature map visualization, knowledge differences between models were explained. Even though there was a knowledge gap between models, this framework distilled a robust student model that is implemented with time-series data alone. I demonstrated this framework with an evaluation of hyper-parameters, invariance from noise, and model reliability. The proposed method produced more effective results than baselines.

In Chapter 6, uncertainty-based feature rectification for KD on multimodal data was proposed. Multiple teachers have been utilized simultaneously to improve the KD learning process; however, teachers do not always generate high-quality knowledge. Also, two teachers were trained with different types of data, but their target task was the same. So, common and different characteristics were included in the outputs of two teachers. Based on these insights, the outputs of teachers are rectified to provide strong knowledge to a student. Firstly, common and different features of teachers were separated, and different weights were applied for rectification by the uncertainty score measured from cross-entropy. In this way, strong supervision can be provided to distill a robust student. The proposed framework was evaluated with several recent works and showed robust effectiveness in KD.

In conclusion, the works in this thesis have been addressed to understand knowledge distillation with geometric approaches for multimodal data analysis. I demonstrated the effectiveness of the proposed works in various aspects, such as feature map

visualization and combining them with existing methods. The solutions are evaluated empirically and outperform many standard baselines on various configurations of models and different types of data.

The proposed methods in this thesis have great potential to be expanded into various research that I discuss for future research below.

Uncertainty-aware feature rectification for knowledge transfer using relationships of inter- and intra-class. As an extended study in Chapter 6, more richer and stronger information can be provided to a student for improving KD performance. Based on rectified features with inter-class addressed in Chapter 6, relationship of intra-class can be transferred jointly, which is computed by the cosine similarity of transposed matrices implying inter-class distribution. Also, feature similarity maps are rectified by common and different features with a mean score of uncertainty within a mini-batch. Then, stronger knowledge can be generated, considering the comprehensive KD learning process, including both inter- and intra-class relationships and intermediate features. Furthermore, a geometric approach can encourage a student to mimic teachers by using graph relationships with direct pair matching or geodesic-based matching. With additional features representing relationships of features, diverse and better interpretable representations, such as orthogonality properties, can be utilized, which aid in improving distillation performance. Thus, the student can obtain more enhanced ability to mimic teachers than using linear metrics, which match feature maps directly.

Analysis of leveraging image representation with multiple teachers in KD on time-series data. To improve the performance of models on time-series data, image representations encoded by time-series data have been utilized simultaneously while complementing time-series features (Som *et al.* (2020); Jeon *et al.* (2022a); Zhang *et al.* (2023)). In KD, using persistence images generated by TDA

was introduced by previous studies (Jeon *et al.* (2022a)) and showed improved performance using multiple teachers. Gramian Angular Fields (GAF) have also been widely utilized in KD for action recognition (Ni *et al.* (2022); Liu *et al.* (2021)). However, it still remains to be seen which image representation provides a better quality of knowledge for the KD learning process. Based on this motivation, I would like to investigate which image representations can make significant contributions to improving performance in the KD process. I would like to utilize more knowledge with different representations to provide richer information to improve performance in distillation. Also, the effects of augmentation methods on the image representations to leverage multiple teachers in KD can be explored.

REFERENCES

Abdel-Hamid, O., A.-r. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu, "Convolutional neural networks for speech recognition", IEEE/ACM Transactions on Audio, Speech, and Language Processing **22**, 10, 1533–1545 (2014).

Adams, H., T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta and L. Ziegelmeier, "Persistence images: A stable vector representation of persistent homology", Journal of Machine Learning Research **18** (2017).

Ahn, S., S. X. Hu, A. Damianou, N. D. Lawrence and Z. Dai, "Variational information distillation for knowledge transfer", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 9163–9171 (2019).

Ali, D., A. Asaad, M.-J. Jimenez, V. Nanda, E. Paluzo-Hidalgo and M. Soriano-Trigueros, "A survey of vectorization methods in topological data analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–14 (2023).

Angarano, S., F. Salvetti, M. Martini and M. Chiaberge, "Generative adversarial super-resolution at the edge with knowledge distillation", Engineering Applications of Artificial Intelligence **123**, 106407 (2023).

Baghersalimi, S., A. Amirshahi, F. Forooghifar, T. Teijeiro, A. Aminifar and D. Atienza, "Many-to-one knowledge distillation of real-time epileptic seizure detection for low-power wearable internet of things systems", arXiv preprint arXiv:2208.00885 (2022).

Bengio, Y., A. Courville and P. Vincent, "Representation learning: A review and new perspectives", IEEE transactions on pattern analysis and machine intelligence **35**, 8, 1798–1828 (2013).

Bergmeir, C., R. J. Hyndman and J. M. Benítez, "Bagging exponential smoothing methods using stl decomposition and box–cox transformation", International journal of forecasting **32**, 2, 303–312 (2016).

Buciluǎ, C., R. Caruana and A. Niculescu-Mizil, "Model compression", in "Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)", pp. 535–541 (2006).

Cao, H., V. Y. Tan and J. Z. Pang, "A parsimonious mixture of gaussian trees model for oversampling in imbalanced and multimodal time-series classification", IEEE Transactions on Neural Networks and Learning Systems **25**, 12, 2226–2239 (2014).

Catal, C., S. Tufekci, E. Pirmit and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition", Applied Soft Computing **37**, 1018–1022 (2015).

Chalapathy, R. and S. Chawla, "Deep learning for anomaly detection: A survey", arXiv preprint arXiv:1901.03407 (2019).

Chen, D., J.-P. Mei, H. Zhang, C. Wang, Y. Feng and C. Chen, "Knowledge distillation with the reused teacher classifier", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 11933–11942 (2022).

Chen, Q., Q. Liu and E. Lin, "A knowledge-guide hierarchical learning method for long-tailed image classification", Neurocomputing **459**, 408–418 (2021).

Chen, Y. and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer", in "Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics", pp. 1488–1492 (2015).

Chen, Z., L. Zhang, Z. Cao and J. Guo, "Distilling the knowledge from handcrafted features for human activity recognition", IEEE Transactions on Industrial Informatics **14**, 10, 4334–4342 (2018).

Cheng, L., S. Luo, X. Yu, H. Ghayvat, H. Zhang and Y. Zhang, "Eeg-clnet: collaborative learning for simultaneous measurement of sleep stages and osa events based on single eeg signal", IEEE Transactions on Instrumentation and Measurement (2023).

Cho, J. H. and B. Hariharan, "On the efficacy of knowledge distillation", in "Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)", pp. 4794–4802 (2019).

Cho, S., I. Ensari, C. Weng, M. G. Kahn and K. Natarajan, "Factors affecting the quality of person-generated wearable device data and associated challenges: Rapid systematic review", JMIR Mhealth Uhealth **9**, 3, e20738, URL https://mhealth.jmir.org/2021/3/e20738 (2021).

Choi, H., A. Som and P. Turaga, "AMC-loss: Angular margin contrastive loss for improved explainability in image classification", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops", pp. 838–839 (2020).

Choi, H., Q. Wang, M. Toledo, P. Turaga, M. Buman and A. Srivastava, "Temporal alignment improves feature quality: an experiment on activity recognition with accelerometer data", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops", pp. 349–357 (2018).

Clark, K., M.-T. Luong, U. Khandelwal, C. D. Manning and Q. Le, "Bam! born-again multi-task networks for natural language understanding", in "Proceedings of the Annual Meeting of the Association for Computational Linguistics", pp. 5931–5937 (2019).

Cortes, C. and V. Vapnik, "Support-vector networks", Machine learning **20**, 3, 273–297 (1995).

Cubuk, E. D., B. Zoph, D. Mane, V. Vasudevan and Q. V. Le, "Autoaugment: Learning augmentation strategies from data", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 113–123 (2019).

Cui, X., V. Goel and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling", IEEE/ACM Transactions on Audio, Speech, and Language Processing **23**, 9, 1469–1477 (2015).

Dalal, N. and B. Triggs, "Histograms of oriented gradients for human detection", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 886–893 (2005).

Darlow, L. N., E. J. Crowley, A. Antoniou and A. J. Storkey, "Cinic-10 is not imagenet or cifar-10", arXiv preprint arXiv:1810.03505 (2018).

Das, D., H. Massa, A. Kulkarni and T. Rekatsinas, "An empirical analysis of the impact of data augmentation on knowledge distillation", arXiv preprint arXiv:2006.03810 (2020).

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in "IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 248–255 (Ieee, 2009).

Deng, J., J. Guo, N. Xue and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 4690–4699 (2019).

Dong, Z., K. Hou, Z. Liu, X. Yu, H. Jia and C. Zhang, "A sample-efficient opf learning method based on annealing knowledge distillation", IEEE Access **10**, 99724–99733 (2022).

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", arXiv preprint arXiv:2010.11929 (2020).

Dutta, A., O. Ma, M. P. Buman and D. W. Bliss, "Learning approach for classification of geneactiv accelerometer data for unique activity identification", in "2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)", pp. 359–364 (IEEE, 2016).

Edelsbrunner, H. and J. L. Harer, *Computational topology: an introduction* (American Mathematical Society, 2022).

Edelsbrunner, H., D. Letscher and A. Zomorodian, "Topological persistence and simplification", Discrete Computational Geometry pp. 511 – 533 (2002).

Eileen, K. T. L., Y. Kuah, K.-H. Leo, S. Sanei, E. Chew and L. Zhao, "Surrogate rehabilitative time series data for image-based deep learning", in "Proceedings of the European Signal Processing Conference", pp. 1–5 (2019).

Esteban, C., S. L. Hyland and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans", arXiv preprint arXiv:1706.02633 (2017).

Fawaz, H. I., G. Forestier, J. Weber, L. Idoumghar and P.-A. Muller, "Deep learning for time series classification: a review", Data Mining and Knowledge Discovery **33**, 4, 917–963 (2019).

Furlanello, T., Z. Lipton, M. Tschannen, L. Itti and A. Anandkumar, "Born again neural networks", in "Proceedings of the International Conference on Machine Learning (ICML)", pp. 1607–1616 (2018).

Gao, J., X. Song, Q. Wen, P. Wang, L. Sun and H. Xu, "Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks", arXiv preprint arXiv:2002.09545 (2020).

Gholizadeh, S. and W. Zadrozny, "A short survey of topological data analysis in time series and systems analysis", arXiv preprint arXiv:1809.10745 (2018).

Gil-Martín, M., R. San-Segundo, F. Fernandez-Martinez and J. Ferreiros-López, "Improving physical activity recognition using a new deep learning architecture and post-processing techniques", Engineering Applications of Artificial Intelligence **92**, 103679 (2020).

Goldblum, M., L. Fowl, S. Feizi and T. Goldstein, "Adversarially robust distillation", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 34, pp. 3996–4003 (2020).

Gou, J., B. Yu, S. J. Maybank and D. Tao, "Knowledge distillation: A survey", International Journal of Computer Vision **129**, 6, 1789–1819 (2021).

Guo, C., G. Pleiss, Y. Sun and K. Q. Weinberger, "On calibration of modern neural networks", in "Proceedings of the International Conference on Machine Learning (ICML)", pp. 1321–1330 (2017).

Guo, W., K. Manohar, S. L. Brunton and A. G. Banerjee, "Sparse-tda: Sparse realization of topological data analysis for multi-way classification", IEEE Transactions on Knowledge and Data Engineering **30**, 7, 1403–1408 (2018).

Ha, S. and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors", in "Proceedings of the International Joint Conference on Neural Networks", pp. 381–388 (2016).

Ha, S., J.-M. Yun and S. Choi, "Multi-modal convolutional neural networks for activity recognition", in "Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics", pp. 3017–3022 (2015).

Han, S., H. Mao and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding", in "Proceedings of the International Conference on Learning and Representations (ICLR)", (2016).

Han, Z., J. Zhao, H. Leung, K. F. Ma and W. Wang, "A review of deep learning models for time series prediction", IEEE Sensors Journal **21**, 6 (2019).

He, K., X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 770–778 (2016).

Hensel, F., M. Moor and B. Rieck, "A survey of topological machine learning methods", Frontiers in Artificial Intelligence **4**, 681108 (2021).

Heo, B., M. Lee, S. Yun and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons", in "Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)", vol. 33, pp. 3779–3787 (2019).

Hinton, G., O. Vinyals and J. Dean, "Distilling the knowledge in a neural network", in "Proceedings of the NeurIPS Deep Learning and Representation Learning Workshop", vol. 2 (2015).

Huang, G., Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 4700–4708 (2017).

Huang, T., S. You, F. Wang, C. Qian and C. Xu, "Knowledge distillation from a stronger teacher", Advances in Neural Information Processing Systems **35**, 33716–33727 (2022).

Jafari, A., M. Rezagholizadeh, P. Sharma and A. Ghodsi, "Annealing knowledge distillation", in "Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics: Main Volume", pp. 2493–2504 (2021).

Jang, I., S. Kim, H. Kim, C.-W. Park and J. H. Park, "An experimental study on reinforcement learning on iot devices with distilled knowledge", in "Proceedings of the International Conference on Information and Communication Technology Convergence", pp. 869–871 (2020).

Jang, Y., H. Lee, S. J. Hwang and J. Shin, "Learning what and where to transfer", in "Proceedings of the International Conference on Machine Learning", vol. 97, pp. 3030–3039 (2019), URL https://proceedings.mlr.press/v97/jang19b.html.

Jeon, E. S., H. Choi, A. Shukla, Y. Wang, M. P. Buman and P. Turaga, "Topological knowledge distillation for wearable sensor data", in "Proceedings of the Asilomar Conference on Signals, Systems, and Computers", pp. 837–842 (2022a).

Jeon, E. S., A. Som, A. Shukla, K. Hasanaj, M. P. Buman and P. Turaga, "Role of data augmentation strategies in knowledge distillation for wearable sensor data", IEEE Internet of Things Journal **9**, 14, 12848–12860 (2022b).

Ji, M., B. Heo and S. Park, "Show, attend and distill: Knowledge distillation via attention-based feature matching", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 35, pp. 7945–7952 (2021).

Jordao, A., A. C. Nazare Jr, J. Sena and W. R. Schwartz, "Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art", arXiv preprint arXiv:1806.05226 (2018).

Kang, Y., R. J. Hyndman and F. Li, "Gratis: Generating time series with diverse and controllable characteristics", Statistical Analysis and Data Mining: The ASA Data Science Journal **13**, 4, 354–376 (2020).

Kania, K. and U. Markowska-Kaczmar, "American sign language fingerspelling recognition using wide residual networks", in "Artificial Intelligence and Soft Computing: 17th International Conference, ICAISC 2018, Zakopane, Poland, June 3-7, 2018, Proceedings, Part I 17", pp. 97–107 (Springer, 2018).

Kegel, L., M. Hahmann and W. Lehner, "Feature-based comparison and generation of time series", in "Proceedings of the International Conference on Scientific and Statistical Database Management", pp. 1–12 (2018).

Khan, A., A. Sohail, U. Zahoora and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks", Artificial Intelligence Review **53**, 8, 5455–5516 (2020).

Kim, H.-J., M. Kim, S.-J. Lee and Y. S. Choi, "An analysis of eating activities for automatic food type recognition", in "Proceedings of the Asia Pacific Signal and Information Processing Association Annual Summit and Conference", pp. 1–5 (2012).

Kirkpatrick, S., C. D. Gelatt Jr and M. P. Vecchi, "Optimization by simulated annealing", science **220**, 4598, 671–680 (1983).

Krizhevsky, A. and G. Hinton, "Learning multiple layers of features from tiny images", Tech. Rep. TR-2009, University of Toronto, Toronto, Ontario (2009).

Kwapisz, J. R., G. M. Weiss and S. A. Moore, "Activity recognition using cell phone accelerometers", ACM SigKDD Explorations Newsletter **12**, 2, 74–82 (2011).

Kwon, K., H. Na, H. Lee and N. S. Kim, "Adaptive knowledge distillation based on entropy", in "Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)", pp. 7409–7413 (2020).

Le, Y. and X. Yang, "Tiny imagenet visual recognition challenge", CS 231N **7**, 7, 3 (2015).

Le Guennec, A., S. Malinowski and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks", in "ECML/PKDD workshop on advanced analytics and learning on temporal data", (2016).

Lee, Y., H. Kim, E. Park, X. Cui and H. Kim, "Wide-residual-inception networks for real-time object detection", in "2017 IEEE Intelligent Vehicles Symposium (IV)", pp. 758–764 (IEEE, 2017).

Li, B., F. Wu, S.-N. Lim, S. Belongie and K. Q. Weinberger, "On feature normalization and data augmentation", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 12383–12392 (2021a).

Li, H., K. Ota and M. Dong, "Learning IoT in edge: Deep learning for the internet of things with edge computing", IEEE Network **32**, 1, 96–101 (2018).

Li, Z., Y. Ming, L. Yang and J.-H. Xue, "Mutual-learning sequence-level knowledge distillation for automatic speech recognition", Neurocomputing **428**, 259–267 (2021b).

Liu, W., Y. Wen, Z. Yu, M. Li, B. Raj and L. Song, "Sphereface: Deep hypersphere embedding for face recognition", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 212–220 (2017).

Liu, W., Y. Wen, Z. Yu and M. Yang, "Large-margin softmax loss for convolutional neural networks", in "Proceedings of the International Conference on Machine Learning (ICML)", vol. 48, pp. 507–516 (2016).

Liu, Y., K. Wang, G. Li and L. Lin, "Semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition", IEEE Transactions on Image Processing **30**, 5573–5588 (2021).

Liu, Y., W. Zhang and J. Wang, "Adaptive multi-teacher multi-level knowledge distillation", Neurocomputing **415**, 106–113 (2020).

Long, M., Z. Cao, J. Wang and M. I. Jordan, "Conditional adversarial domain adaptation", Advances in Neural Information Processing Systems **31** (2018).

Lowe, D. G., "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision **60**, 2, 91–110 (2004).

Molchanov, P., S. Tyree, T. Karras, T. Aila and J. Kautz, "Pruning convolutional neural networks for resource efficient inference", in "Proceedings of the International Conference on Learning Representations", (2017).

Naeini, M. P., G. Cooper and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning", in "Proceedings of the AAAI Conference on Artificial Intelligence", pp. 2901–2907 (2015).

Nawar, A., F. Rahman, N. Krishnamurthi, A. Som and P. Turaga, "Topological descriptors for parkinson's disease classification and regression analysis", in "Proceedings of the Annual International Conference of the IEEE Engineering in Medicine & Biology Society", pp. 793–797 (2020).

Ni, J., R. Sarbajna, Y. Liu, A. H. Ngu and Y. Yan, "Cross-modal knowledge distillation for vision-to-sensor action recognition", in "Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing", pp. 4448–4452 (2022).

NVIDIA, "Nvidia titan xp", Accessed: January 20, 2022. Available: https://www.nvidia.com/en-us/titan/titan-xp/ (2016).

Nweke, H. F., Y. W. Teh, M. A. Al-Garadi and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges", Expert Systems with Applications **105**, 233–261 (2018).

Pachauri, D., C. Hinrichs, M. K. Chung, S. C. Johnson and V. Singh, "Topology-based kernels with application to inference problems in alzheimer's disease", IEEE transactions on medical imaging **30**, 10, 1760–1770 (2011).

Park, D. S., W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition", in "Proceedings of the Interspeech", pp. 2613–2617 (2019a).

Park, S., K. Yoo and N. Kwak, "On the orthogonality of knowledge distillation with other techniques: From an ensemble perspective", arXiv preprint arXiv:2009.04120 (2020).

Park, W., D. Kim, Y. Lu and M. Cho, "Relational knowledge distillation", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 3967–3976 (2019b).

Peng, B., X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou and Z. Zhang, "Correlation congruence for knowledge distillation", in "Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)", pp. 5007–5016 (2019).

Plastiras, G., M. Terzi, C. Kyrkou and T. Theocharidcs, "Edge intelligence: Challenges and opportunities of near-sensor machine learning applications", in "Proceedings of the IEEE International Conference on Application-specific Systems, Architectures and Processors", pp. 1–7 (2018).

Qi, P., X. Zhou, Y. Ding, Z. Zhang, S. Zheng and Z. Li, "Fedbkd: Heterogenous federated learning via bidirectional knowledge distillation for modulation classification in iot-edge system", IEEE Journal of Selected Topics in Signal Processing **17**, 1, 189–204 (2023).

Reich, S., D. Mueller and N. Andrews, "Ensemble distillation for structured prediction: Calibrated, accurate, fast-choose three", arXiv preprint arXiv:2010.06721 (2020).

Reiss, A. and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring", in "Proceedings of the International Symposium on Wearable Computers", pp. 108–109 (2012).

Remigereau, F., D. Mekhazni, S. Abdoli, R. M. Cruz, E. Granger *et al.*, "Knowledge distillation for multi-target domain adaptation in real-time person re-identification", in "2022 IEEE International Conference on Image Processing (ICIP)", pp. 3853–3557 (IEEE, 2022).

Rieck, B., T. Yates, C. Bock, K. Borgwardt, G. Wolf, N. Turk-Browne and S. Krishnaswamy, "Uncovering the topology of time-varying fmri data using cubical persistence", Advances in Neural Information Processing Systems **33**, 6900–6912 (2020).

Romero, A., N. Ballas, S. E. Kahou, A. Chassang, C. Gatta and Y. Bengio, "Fitnets: Hints for thin deep nets", in "Proceedings of the International Conference on Learning and Representations (ICLR)", pp. 1–13 (2015).

Rosenberg, A. and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure", in "Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning", pp. 410–420 (2007).

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 4510–4520 (2018).

Saul, N. and C. Tralie, "Scikit-tda: Topological data analysis for python", URL https://doi.org/10.5281/zenodo.2533369 (2019).

Selvaraju, R. R., 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/CVF International Conference on Computer Vision (ICCV)", pp. 618–626 (2017).

Seversky, L. M., S. Davis and M. Berger, "On time-series topological data analysis: New data and opportunities", in "Proceedings of the IEEE conference on computer vision and pattern recognition workshops", pp. 59–67 (2016).

Simonyan, K. and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv:1409.1556 (2014).

Som, A., H. Choi, K. N. Ramamurthy, M. P. Buman and P. Turaga, "Pi-net: A deep learning approach to extract topological persistence images", in "Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops", pp. 834–835 (2020).

Song, M., A. Mallol-Ragolta, E. Parada-Cabaleiro, Z. Yang, S. Liu, Z. Ren, Z. Zhao and B. W. Schuller, "Frustration recognition from speech during game interaction using wide residual networks", Virtual Reality & Intelligent Hardware **3**, 1, 76–86 (2021).

Stanton, S. D., P. Izmailov, P. Kirichenko, A. A. Alemi and A. G. Wilson, "Does knowledge distillation really work?", in "Advances in Neural Information Processing Systems (NeuRIPS)", (2021), URL https://openreview.net/forum?id=7J-fKoXiReA.

Steven Eyobu, O. and D. S. Han, "Feature representation and data augmentation for human activity classification based on wearable imu sensor data using a deep lstm neural network", Sensors **18**, 9, 2892 (2018).

Sun, Y., Y. Chen, X. Wang and X. Tang, "Deep learning face representation by joint identification-verification", in "Proceedings of the International Conference on Neural Information Processing Systems (NeuRIPS)", vol. 2, pp. 1988–1996 (2014).

Tai, C., T. Xiao, Y. Zhang, X. Wang and E. Weinan, "Convolutional neural networks with low-rank regularization", in "Proceedings of the International Conference on Learning Representations (ICLR)", (2016).

Tarvainen, A. and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results", in "Proceedings of the International Conference on Neural Information Processing Systems", pp. 1195–1204 (2017).

Thai, C., V. Tran, M. Bui, D. Nguyen, H. Ninh and H. Tran, "Real-time masked face classification and head pose estimation for rgb facial image via knowledge distillation", Information Sciences **616**, 330–347 (2022).

Tian, Y., D. Krishnan and P. Isola, "Contrastive representation distillation", arXiv preprint arXiv:1910.10699 (2019).

Tolstikhin, I. O., N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision", Advances in Neural Information Processing Systems **34**, 24261–24272 (2021).

Tripathi, A. M. and K. Paul, "Data augmentation guided knowledge distillation for environmental sound classification", Neurocomputing **489**, 59–77 (2022).

Tung, F. and G. Mori, "Similarity-preserving knowledge distillation", in "Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)", pp. 1365–1374 (2019).

Turkeš, R., J. Nys, T. Verdonck and S. Latré, "Noise robustness of persistent homology on greyscale images, across filtrations and signatures", Plos one **16**, 9, e0257215 (2021).

Um, T. T., F. M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek and D. Kulić, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks", in "Proceedings of the 19th ACM International Conference on Multimodal Interaction", pp. 216–220 (2017).

van der Maaten, L. and G. Hinton, "Visualizing data using t-sne", Journal of Machine Learning Research **9**, 86, 2579–2605 (2008).

Wan, S., L. Qi, X. Xu, C. Tong and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones", Mobile Networks and Applications **25**, 2, 743–755 (2020).

Wang, D., D. Wen, J. Liu, W. Tao, T.-W. Chen, K. Osa and M. Kato, "Fully supervised and guided distillation for one-stage detectors", in "Proceedings of the Asian Conference on Computer Vision (ACCV)", pp. 171–188 (2020a).

Wang, F., J. Cheng, W. Liu and H. Liu, "Additive margin softmax for face verification", IEEE Signal Processing Letters **25**, 7, 926–930 (2018a).

Wang, H., Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li and W. Liu, "Cosface: Large margin cosine loss for deep face recognition", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 5265–5274 (2018b).

Wang, J., Y. Chen, R. Chakraborty and S. X. Yu, "Orthogonal convolutional neural networks", in "Proceedings of the IEEE/CVF conference on computer vision and pattern recognition", pp. 11505–11515 (2020b).

Wang, K., X. Gao, Y. Zhao, X. Li, D. Dou and C.-Z. Xu, "Pay attention to features, transfer learn faster cnns", in "Proceedings of the International Conference on Learning Representations (ICLR)", pp. 1–14 (2020c).

Wang, L. and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2021).

Wang, Q., S. Lohit, M. J. Toledo, M. P. Buman and P. Turaga, "A statistical estimation framework for energy expenditure of physical activities from a wrist-worn accelerometer", in "Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society", vol. 2016, pp. 2631–2635 (2016).

Wang, T., L. Yuan, X. Zhang and J. Feng, "Distilling object detectors with fine-grained feature imitation", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 4933–4942 (2019).

Wang, X. and C. Wang, "Time series data cleaning: A survey", IEEE Access **8**, 1866–1881 (2019).

Wang, Y., R. Behroozmand, L. P. Johnson, L. Bonilha and J. Fridriksson, "Topological signal processing and inference of event-related potential response", Journal of Neuroscience Methods **363**, 109324 (2021).

Wen, Q., L. Sun, X. Song, J. Gao, X. Wang and H. Xu, "Time series data augmentation for deep learning: A survey", arXiv preprint arXiv:2002.12478 (2020).

Wen, Q., L. Sun, F. Yang, X. Song, J. Gao, X. Wang and H. Xu, "Time series data augmentation for deep learning: A survey", in "Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI", pp. 4653–4660 (2021a).

Wen, T., S. Lai and X. Qian, "Preparing lessons: Improve knowledge distillation with better supervision", Neurocomputing **454**, 25–33 (2021b).

Wen, Y., K. Zhang, Z. Li and Y. Qiao, "A discriminative feature learning approach for deep face recognition", in "Proceedings of the European Conference on Computer Vision (ECCV)", pp. 499–515 (2016).

Wu, J., C. Leng, Y. Wang, Q. Hu and J. Cheng, "Quantized convolutional neural networks for mobile devices", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 4820–4828 (2016).

Xiong, W., L. Wu, F. Alleva, J. Droppo, X. Huang and A. Stolcke, "The microsoft 2017 conversational speech recognition system", in "Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing", pp. 5934–5938 (2018).

Yang, C., L. Xie, S. Qiao and A. L. Yuille, "Training deep neural networks in generations: A more tolerant teacher educates better students", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 33, pp. 5628–5635 (2019).

Yang, X.-S., *Nature-inspired optimization algorithms* (Academic Press, 2020).

Yen, P. T.-W. and S. A. Cheong, "Using topological data analysis (tda) and persistent homology to analyze the stock markets in singapore and taiwan", Frontiers in Physics p. 20 (2021).

Yim, J., D. Joo, J. Bae and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 4133–4141 (2017).

You, S., C. Xu, C. Xu and D. Tao, "Learning from multiple teacher networks", in "Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 1285–1294 (2017).

Yun, S., D. Han, S. J. Oh, S. Chun, J. Choe and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features", in "Proceedings of the IEEE/CVF international conference on computer vision", pp. 6023–6032 (2019).

Zagoruyko, S. and N. Kmodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer", in "Proceedings of the International Conference on Learning and Representations (ICLR)", pp. 1–13 (2017).

Zagoruyko, S. and N. Komodakis, "Wide residual networks", in "Proceedings of the British Machine Vision Conference (BMVC)", pp. 87.1–87.12 (2016).

Zeng, S., F. Graf, C. Hofer and R. Kwitt, "Topological attention for time series forecasting", Advances in Neural Information Processing Systems **34**, 24871–24882 (2021).

Zhang, H., D. Chen and C. Wang, "Confidence-aware multi-teacher knowledge distillation", in "Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)", pp. 4498–4502 (2022).

Zhang, H., M. Cisse, Y. N. Dauphin and D. Lopez-Paz, "mixup: Beyond empirical risk minimization", in "Proceedings of the International Conference on Learning and Representations (ICLR)", (2018a).

Zhang, K., H. Ying, H.-N. Dai, L. Li, Y. Peng, K. Guo and H. Yu, "Compacting deep neural networks for internet of things: Methods and applications", IEEE Internet of Things Journal **8**, 15, 11935–11959 (2021).

Zhang, Q., Z. Qi, P. Cui, M. Xie and J. Din, "Detection of single-phase-to-ground faults in distribution networks based on gramian angular field and improved convolutional neural networks", Electric Power Systems Research **221**, 109501 (2023).

Zhang, Y., T. Xiang, T. M. Hospedales and H. Lu, "Deep mutual learning", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 4320–4328 (2018b).

Zheng, A. and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists* (O'Reilly Media, Inc., 2018).