Constrained Adaptive Distillation Based on Topological Persistence for Wearable Sensor Data

Eun Som Jeon, *Graduate Student Member, IEEE*, Hongjun Choi, Ankita Shukla, Yuan Wang, Matthew P. Buman, and Pavan Turaga, *Senior Member, IEEE*

***Abstract*— Wearable sensor data analysis with persistence features generated by topological data analysis (TDA) has achieved great success in various applications, and however, it suffers from large computational and time resources for extracting topological features. In this article, our approach utilizes knowledge distillation (KD) that involves the use of multiple teacher networks trained with the raw time series and persistence images (PIs) generated by TDA. However, direct transfer of knowledge from the teacher models utilizing different characteristics as inputs to the student model results in a knowledge gap and limited performance. To address this problem, we introduce a robust framework that integrates multimodal features from two different teachers and enables a student to learn desirable knowledge effectively. To account for statistical differences in multimodalities, an entropy-based constrained adaptive weighting mechanism is leveraged to automatically balance the effects of teachers and encourage the student model to adequately adopt the knowledge from two teachers. To assimilate dissimilar structural information generated by different style models for distillation, batch and channel similarities within a mini-batch are used. We demonstrate the effectiveness of the proposed method on wearable sensor data.**

***Index Terms*— Knowledge distillation (KD), topological data analysis (TDA), wearable sensor data.**

I. Introduction

CONVERTING wearable sensor data to impactful health applications continues to be challenging. The sources of variability in the raw sensor data include: 1) sensor-level noise characteristics; 2) drifts in sampling rates; 3) gaps in recorded sensor data; 4) intrinsic variability in physiological signals; and 5) 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 [2], [3]. These methods have achieved great success in various fields, such as human activity recognition [4], [5], disease classification [4], [6], and shape and texture classification [7]. In particular, persistence images (PIs) have been widely used for 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 [8].

Beyond just the computational load of TDA, it has also been found that the TDA features have many different data structures such as barcodes and persistence diagrams (PDs), which can be featurized in many ways, but their integration with contemporary machine learning techniques has required independently computing the features and fusing with deep features later [2], [9]. 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 [5]. However, careful use of knowledge distillation (KD) 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 article, we address these issues by employing 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 [10], [11], [12], [13], [14], [15]. Also, KD has been broadly used to design a real-time system [16], [17], [18], [19]. Incorporating multiple teachers in KD has been shown to improve the performance by leveraging various features [15], [20], [21], which are generally implemented in a unimodal manner. We utilize multiple teacher networks trained with the raw time series and PIs generated by TDA. Importantly, a single student is implemented with only time-series data as an input. However, we found two significant challenges in utilizing different teachers in KD.

1) The large discrepancy between the 1-D time-series and 2-D TDA feature representations makes it difficult to

Manuscript received 20 March 2023; revised 4 September 2023; accepted 26 September 2023. Date of publication 3 November 2023; date of current version 13 November 2023. This work was supported in part by NIH under Grant R01GM135927, as part of the Joint Division of Mathematical Science (DMS)/National Institute of General Medical Sciences (NIGMS) Initiative to Support Research at the Interface of the Biological and Mathematical Sciences, and in part by NSF under Grant 2200161. An earlier version of this paper was presented at the Asilomar Conference on Signals, Systems, and Computers, 2022 [DOI: 10.1109/IEEECONF56349.2022.10052019]. The Associate Editor coordinating the review process was Dr. Jingyu Hua. *(Corresponding author: Eun Som Jeon.)*

Eun Som Jeon, Ankita Shukla, and Pavan Turaga are with the Geometric Media Laboratory, School of Arts, Media and Engineering, and the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: ejeon6@asu.edu; Ankita.Shukla@asu.edu; pturaga@asu.edu).

Hongjun Choi is with the Lawrence Livermore National Laboratory, Livermore, CA 94550 USA (e-mail: choi22@llnl.gov).

Yuan Wang is with the Department of Epidemiology and Biostatistics, University of South Carolina, Columbia, SC 29208 USA (e-mail: wang578@mailbox.sc.edu).

Matthew P. Buman is with the College of Health Solutions, Arizona State University, Phoenix, AZ 85004 USA (e-mail: mbuman@asu.edu).

Digital Object Identifier 10.1109/TIM.2023.3329818

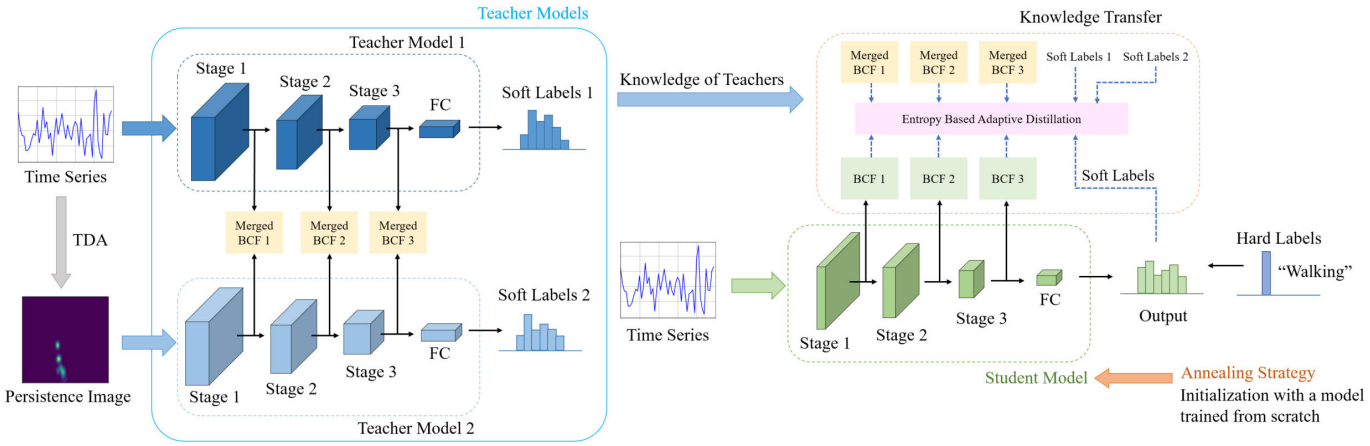


Fig. 1. Overview of CADTP. A compact student model is trained by using two teachers, which are learned with different representations of the same raw time-series data. BCF denotes batch and channel similarity features.

effectively fuse multimodal features and run models on a unified framework.

2) Due to the different architectural designs of teachers and student (e.g., 1-D convolutional neural networks (CNNs) versus 2-D CNNs), it is challenging to extract similar structural features that allow the student to benefit from distillation.

To address these problems, we 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 Fig. 1. First, to obtain topological features, PIs are extracted from PDs. We train two models with time-series data and PIs. In the second step, the pretrained 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, we 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, we 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 article are given as follows.

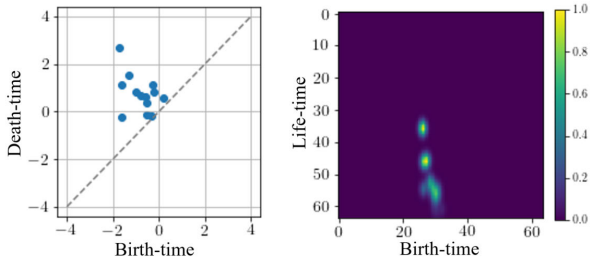
1) We propose a new framework with KD, which transfers time-series and topological features to a student using time-series data only as an input.
2) We propose a technique for adaptive distillation that balances the influence of different teachers based on entropy to effectively transfer knowledge despite the statistical difference in their features.
3) We 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.
4) We 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 this article is organized as follows. In Section II, we provide a brief overview of creating PIs, KD techniques, and an annealing strategy. In Section III, we introduce the proposed new framework for KD. In Section IV, we describe our experimental results and analysis. In Section V, we discuss our findings and conclusions.

II. Background

A. Topological Feature Extraction

The integration of TDA with machine learning has shown robust performance in many applications [9], [22], [23]. 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 [24]. From this filtration, the birth and death of these topological cavities can be described as a point (x, y) in the PD, where x and y are the coordinates of planar scatter points [2], [9]. Applying PDs directly to complex machine learning tasks is challenging because they have intrinsically heterogeneous statistical characteristics. PDs are multisets 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.



(a) (b)

Fig. 2. (a) PD and (b) its corresponding PI. In PD, based on weighting function, points with higher lifetime appear brighter.

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 [2], [8]. 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. Fig. 2 shows an example of a PD and its PI. However, due to the high computational complexity required to extract PIs by TDA [5], this method is difficult to use on small devices with limited power and computational resources. To solve this issue, in this article, we propose a framework based on KD that trains a smaller single student model with topological knowledge to generate good performance as a larger model.

B. Application of TDA for Activity Recognition

There are a lot of works utilizing topological knowledge in applications for activity recognition [5], [6], [25]. These methods use vectorized topological features from PI as inputs to machine learning methods, generally resulting in robustness to signal perturbation. Nawar et al. [6] 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 preprocessing steps for training as well as testing to extract topological features by TDA and transform knowledge into manually defined terms. PI-Net [5] 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 article, we propose a framework to generate a single small model using time-series data only, which does not require preprocessing to generate PI, nor needing to run different models separately at test time.

C. Knowledge Distillation

KD is the process of training a smaller model from the knowledge of a larger model. KD was first introduced by Buciluǎ et al. [26] and further developed by Hinton et al. [27]. 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 [27]. For traditional KD, the loss function for training a student is

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

where $\mathcal{L}_{\mathcal{CE}}$ is the cross-entropy loss, $\mathcal{L}_{\mathcal{KD}}$ is the 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

$$\mathcal{L}_{\mathcal{CE}} = \mathcal{Q}(\sigma(t_S), y) \tag{2}$$

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 the Kullback–Leibler (KL) divergence loss

$$\mathcal{L}_{\mathcal{KD}} = \tau^2 \mathrm{KL}(p_T, p_S) \tag{3}$$

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 and Hariharan [28] investigated the effects of early stopping for KD (ESKD) to distill a better student. To obtain the best performance, we use ESKD that improves the efficacy of KD [28].

As an extension of response-based knowledge using logits, feature-based KD has been used to improve the performance [15], [29], [30], [31]. First, the intermediate representations were introduced in Fitnets [29]. 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 and Mori [31], 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:

$$A = F_b \cdot F_b^{\top}; \quad F_b \in \mathbb{R}^{b\times chw} \tag{4}$$

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, and however, they generally focus on utilizing a single teacher in a unimodal manner.

To transfer more useful information, the use of multiple teachers has been proposed [15], [20], [21]. Since different teachers can produce diverse knowledge, richer knowledge can be leveraged to improve the performance of a student [15]. Despite initial attempts [32], [33], 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 a student, which results in performance degradation [15].

To advance beyond these problems, we develop a framework in KD using a 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 are described in Section III.

D. Simulated Annealing in KD

Kirkpatrick et al. [34] introduced simulated annealing, which has been applied to various fields, including machine learning for solving optimization problems [35]. Jafari et al. [36] 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. [37] 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 fine-tuning epochs increase. An annealing strategy of the proposed method has different aspects, compared to prior studies [36], [37], [38]. For the proposed method, multiple teachers are trained with different modalities—time-series and PI 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, we 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. The overall 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 the student is also mitigated.

III. Proposed Approach

The proposed method utilizes two teachers trained with different data to train a student. First, 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. Second, logits of teacher and student networks are used to calculate entropy for balancing the effects of two teachers, considering statistical differences in multimodalities. 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. In addition, an annealing strategy for KD 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.

A. PI Extraction

To compute PIs, first, we utilize the Scikit-TDA python library [39] and the Ripser package for generating PDs, as described in [5]. 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 versus 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, respectively. Second, we 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.

B. 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, we adopt KD to distill a small model using time-series data alone as an input, to acquire beneficial topological features from a teacher.

1) 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

$$\mathcal{L}_{\mathcal{KD}_m} = \tau^2\big(\alpha \mathrm{KL}(p_{T_1}, p_S) + (1-\alpha)\mathrm{KL}(p_{T_2}, p_S)\big) \quad (5)$$

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.

2) 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, we use the entropy of teachers, which can be utilized as an uncertainty indicator [33]. 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, we propose constrained adaptive distillation based on entropy. If the entropy value of labels is smaller, the effect of the KD loss is more important [33], [40]. 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, we 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, we set α at each sample. The weight α is determined according to the following rule:

$$\alpha_i = \begin{cases} 0.5 + \big(1/(1+e^{-\mathrm{epoch}/\beta}) - 0.5\big)/\kappa & \text{if} \mathcal{H}(t^i_{T_1}) < \mathcal{H}\big(t^i_{T_2}\big) \\ 0.5 - \big(1/(1+e^{-\mathrm{epoch}/\beta}) - 0.5\big)/\kappa & \text{otherwise} \end{cases} \quad (6)$$

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

$$\mathcal{L}_{\mathcal{KD}\text{ent}} = \frac{1}{n}\sum_{i=1}^{n} \tau^2 \left(\alpha_i \text{KL}\left(p^i_{T_1}, p^i_S\right) + (1-\alpha_i)\text{KL}\left(p^i_{T_2}, p^i_S\right)\right) \tag{7}$$

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

3) Extracting Features of Different Teachers: To provide more comprehensive knowledge from the teachers, we 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, we use a method similar to that in [31], which can easily make features match the dimensions of activation maps from different models, as defined in (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, we 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:

$$G = F_c \cdot F_c^{\top}; F_c \in \mathbb{R}^{c\times bhw} \tag{8}$$

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

Fig. 3 shows the batch and channel similarity maps from two teachers. The similarity maps highlight differently and show dissimilar patterns. Thus, these maps 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 [15], [33]. 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, we 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 given as follows:

$$A^{(l)}_T = \alpha_{\text{avg}} A^{(l^{T_1})}_{T_1} + (1-\alpha_{\text{avg}}) A^{(l^{T_2})}_{T_2} \tag{9}$$

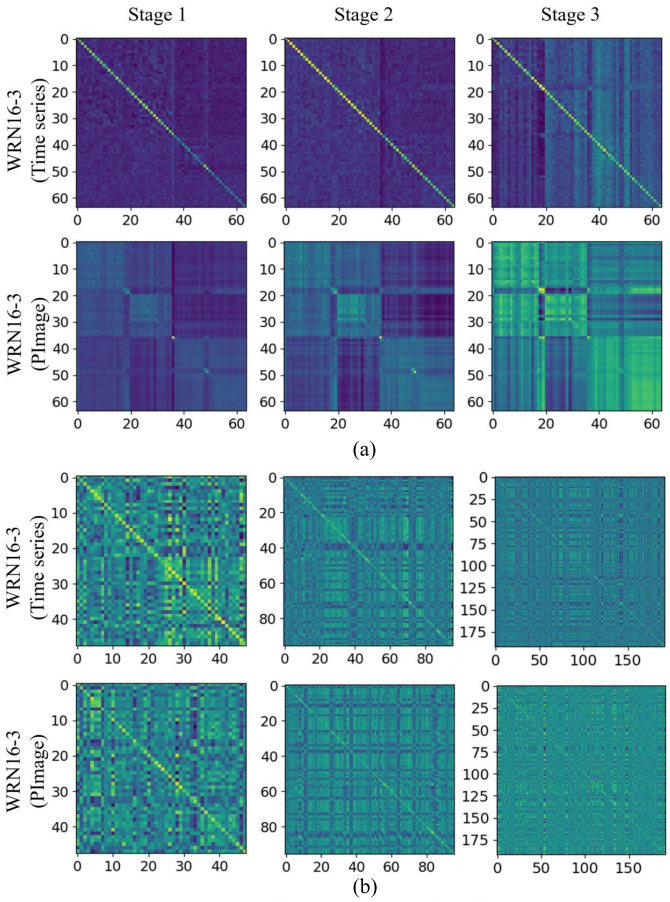


Fig. 3. Examples of activation similarity maps A and G produced by a layer for the indicated stage of the network for a batch on GENEActiv. High similarities for samples within the batch are shown with high values. The blockwise pattern is more prominent for batch similarity maps using PI. The maps with different modalities and similarities represent dissimilar patterns, which implies that these maps can capture the diverse semantics of the dataset. (a) Batch similarity maps. (b) Channel similarity maps.

where $A^{(l)}_T \in \mathbb{R}^{b\times b}$ 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 given as follows:

$$G^{(l)}_T = \alpha_{\text{avg}} G^{(l^{T_1})}_{T_1} + (1-\alpha_{\text{avg}}) G^{(l^{T_2})}_{T_2} \tag{10}$$

where $G^{(l)}_T \in \mathbb{R}^{c^{(l)}\times c^{(l)}}$ 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 sizes, the larger one is resized to match the smaller one. By merging the maps, the similarities between two teachers are more highlighted.

4) 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 sizes, the larger one is resized to meet the size of the smaller one. The overview of transferring knowledge with similarity maps is described in Fig. 4. By minimizing the difference between the teachers and the student, the information from similarity maps is transferred as

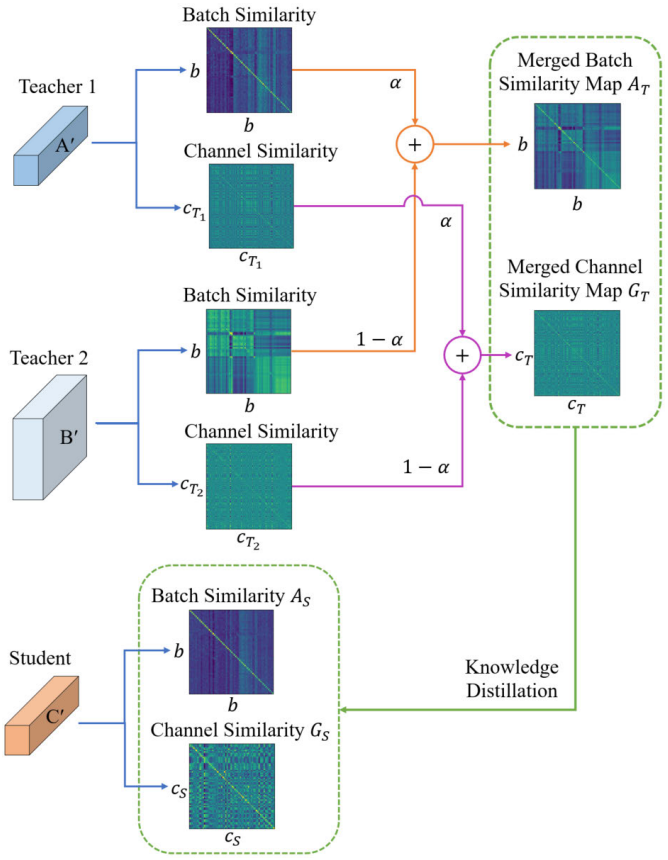


Fig. 4. Framework of extracting and transferring similar features from different teachers. A′ and B′ denote mini-batch features at a layer of Teacher1 and Teacher2, respectively. C′ denotes mini-batch features at a layer of Student.

follows:

$$\mathcal{L}_{\text{sim}} = \frac{1}{|L|} \sum_{(l,l^S)\in L} \left(\frac{\gamma_b}{b^2} \left\| \widetilde{A_T^{(l)}} - \widetilde{A_S^{(l^S)}} \right\|_F^2 + \frac{\gamma_c}{c_{(l)}^2} \left\| \widetilde{G_T^{(l)}} - \widetilde{G_S^{(l^S)}} \right\|_F^2 \right) \quad (11)$$

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 [31]. 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

$$\mathcal{L}_{\text{CADTP}} = (1-\lambda)\mathcal{L}_{\mathcal{CE}} + \lambda \mathcal{L}_{\mathcal{KD}_{\text{ent}}} + \eta \mathcal{L}_{\text{sim}} \quad (12)$$

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

C. 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 [15], [36]. To mitigate the effects of the knowledge gap, we use an annealing strategy in KD for the proposed framework. First, a small model that has the same architecture as a student is learned from scratch. Second, when the weight values of a student model are initialized for the training process in KD, the values are determined by the pretrained 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.

IV. Experiments

In this section, we describe datasets used for evaluation and experimental settings. We demonstrate the proposed method with various teacher–student combinations on wearable sensor data. We analyze the proposed method under different noise levels and various hyperparameters. Furthermore, we investigate the effectiveness of CADTP with visualization of feature maps and generalizability analysis. Finally, we compare and contrast the computational time with different methods.

A. Data Description and Experimental Settings

1) Data Description: We evaluate the proposed method with wearable sensor data on GENEActiv and PAMAP2 datasets.

1) *GENEActiv:* It is an experimental device calibration dataset [41] collected with GENEActiv sensor, which is a lightweight, waterproof, and wrist-worn triaxial accelerometer with a 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. We use 14 daily activities used as in [14]. Each activity class has over 900 samples. We use a full nonoverlapping window size of 500 time steps (5 s). The numbers of subjects for training and testing are 131 and 43, respectively. The numbers of samples for training and testing are approximately 16k and 6k, respectively.
2) *PAMAP2:* It is a publicly accessible dataset [42], which includes measurements of heart rate, temperature, accelerometer, gyroscopes, and magnetometers with 100-Hz sampling frequency for nine subjects (24–32 years of age). The sensors were placed on hands, chest, and ankles of the subjects. We use 12 daily activities with 40 channels, which were recorded from the heart rate and four inertial measurement units (IMUs), where activities are lying, sitting, standing, walking, and so on. To compare with previous methods, the recordings are downsampled to 33.3 Hz. The evaluation protocol on this dataset follows leave-one-subject-out. Data dropping and connection loss occurred because data were collected using wireless sensors, so missing data are included. The dataset has a nonuniform distribution. We utilize 100 time steps (3 s) of a sliding window for a sample with 22 time steps (660 ms) of step size for segmenting the sequences, which allows semi-nonoverlapping sliding windows with 78% overlapping [42].

TABLE I

DETAILS OF TEACHER AND STUDENT NETWORK ARCHITECTURES. THE COMPRESSION RATIO IS CALCULATED WITH TWO TEACHERS

DB	Teacher1 (1D CNNs) & Teacher2 (2D CNNs)	Student	FLOPs (Teacher1)	FLOPs (Teacher2)	FLOPs (Student)	# of params (Teacher1)	# of params (Teacher2)	# of params (Student)	Compression ratio
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%

2) 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 and 10, which are the same as in the previous study [5]. For PAMAP2, the Gaussian function parameter and the 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, we set the total number of epochs as 200, using stochastic gradient descent (SGD) with momentum of 0.9, 64 as the batch size, and a weight decay as 1×10^{-4}. We have different strategies for training models with time-series and image representations. The model trained with time-series data is incorporated with 1-D convolutional layers, and on the other hand, the one trained with image data is designed with 2-D convolutional layers. To train a model with time-series data, the initial learning rate is 0.05, which decreases by 0.2 at ten epochs and drops down by 0.1 every $[t/3]$, 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 ten epochs and drops down by 0.2 at 40, 80, 120, and 160 epochs. To evaluate the performance of the proposed method, we use WideResNet (WRN) [43] to construct different combinations of teachers and student, which is popularly used in the validation of KD [14], [28]. Also, WRN has been used to design real-time system [44], [45], [46]. As the previous works do [14], τ and λ are set as 4 and 0.7 for GENEActiv and as 4 and 0.99 for PAMAP2, respectively. We run three times and the best averaged accuracy and standard deviation are reported for the following experiments. We perform baseline comparisons with traditional KD [27], attention transfer (AT) [30], similarity preserving (SP) KD [31], and simple KD (SimKD) [47], 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, we compare with DIST [48], which considers intraclass and interclass relationships for knowledge transfer. In addition, we compare with multiteacher-based approaches, such as AVER [32], EBKD [33], CA-MKD [21], Base [1], and AdTemp [1]. Since we 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.

B. Various Capacities of Teachers

In this section, we evaluate the proposed method with various capacities of teachers that are trained with time-series data and PIs. WRN16-1 (1-D 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 I, representing model complexity and the number of trainable parameters. The results with various teachers on GENEActiv are described in Table II. Note, "time series" and "PImage" denote results of KD methods with Teacher1 trained with 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 Table II are 0.2, 0.01, 0.01,

TABLE II

ACCURACY (%) WITH VARIOUS KD METHODS FOR DIFFERENT CAPACITIES 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)			
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
	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

TABLE III

ACCURACY (%) FOR RELATED METHODS ON GENEACTIV WITH SEVEN CLASSES

	Method	Window length 1000	Window length 500
time-series	SVM [49]	86.29	85.86
	Choi *et al.* [50]	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 IV

ACCURACY (%) WITH VARIOUS KD METHODS FOR DIFFERENT CAPACITIES 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

TABLE V

ACCURACY (%) FOR RELATED METHODS ON PAMAP2

	Method	Accuracy (%)
time-series	Chen and Xue [51]	83.06
	Ha *et al.* [52]	73.79
	Ha and Choi [53]	74.21
	Kwapisz [54]	71.27
	Catal *et al.* [55]	85.25
	Kim *et al.* [56]	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

and 0.2, from left to right. As shown in Table II, 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 in improving 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 [28]. To investigate with different sizes of window lengths and more previous methods, we test the methods with seven classes of GENEActiv dataset, as do the previous study [14], [50]. β and κ are 1.0 and 1.5, respectively. η parameters are 900 for a window size of 500 and 100 for a window size of 1000. γ_c values are 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 III, CADTP achieves the best performing results, indicating that the proposed method aids in performance improvement. The results on PAMAP2 are described in Table IV. β and κ are 0.3 and 2.5, respectively. η is 200. The γ_c values of the teachers in Table IV 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.). Furthermore, as shown in Table V, CADTP outperforms the baselines. As a result, the proposed method improves the performance while also allowing for effective model compression.

C. Various Combinations of Teachers

To explore the effects of different architectures for teachers, we test with different depths and widths of WRNs, as described in Tables VI and VII. 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 shown in Table VI, CADTP produces the best student in almost all cases. When the 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

TABLE VI
ACCURACY (%) WITH VARIOUS KD METHODS FOR DIFFERENT STRUCTURES 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 VII
ACCURACY (%) WITH VARIOUS KD METHODS FOR DIFFERENT STRUCTURES 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

(WRN28-1, WRN16-3); otherwise, the value is 0.2. As shown in Table VII, 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 [28].

D. Ablations and Sensitivity Analysis

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

1) Analysis of Invariance From Noise: To investigate the ability of models to be robust to different types of noise, we conducted experiments with noisy testing data by injecting continuous missing and Gaussian noise [14], [57], [58]. 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 and the standard deviation for Gaussian noise, respectively. 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 that the classifiers were trained with the original training set.

As shown in Fig. 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 depths or widths, 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.

2) 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 with respect to these hyperparameters, we conduct the following experiments.

A student (WRN16-1) is trained with two teachers by using different γ_c and η values, as shown in Fig. 6. In Fig. 6(a) and (b), the other hyperparameters are set as in Section IV-A. 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 are also shown the similar. The results with various η are presented in

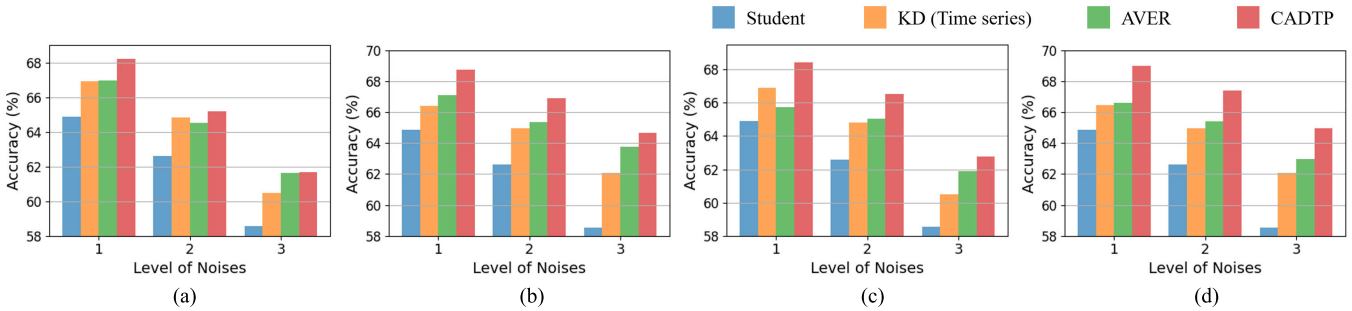


Fig. 5. Accuracy (%) with various KD methods for different noise severity levels on GENEActiv. Brackets denote (Teacher1, Teacher2). Students are WRN16-1 (1-D CNNs). (a) (WRN16-1, WRN16-1). (b) (WRN28-1, WRN28-1). (c) (WRN28-1, WRN28-3). (d) (WRN16-1, WRN40-1).

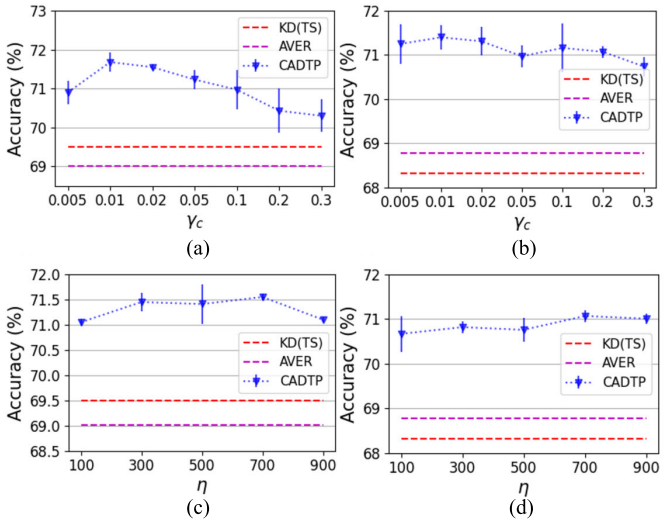


Fig. 6. Sensitivity to γ_c and η of the proposed method for WRN16-1 students on GENEActiv. (a) Results on γ_c from WRN16-3 teachers. (b) Results on γ_c from WRN28-1 teachers. (c) Results on η from WRN16-3 teachers. (d) Results on η from WRN28-1 teachers.

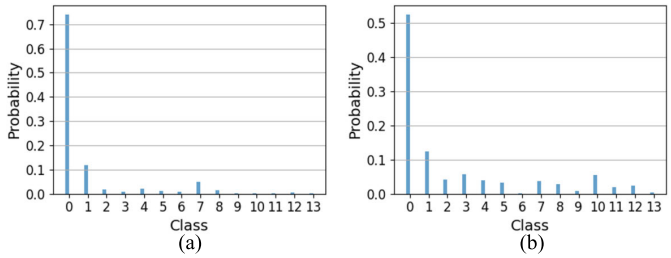


Fig. 7. Probability distributions for models trained with different modalities. Testing samples of class 0 are used to measure the probability. (a) Time-series data. (b) PI.

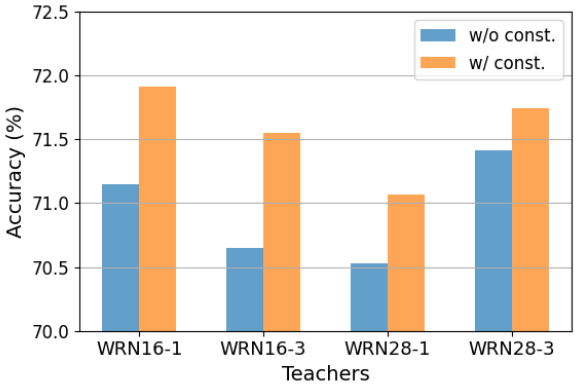


Fig. 8. Accuracy (%) of the proposed method with or without constraints on GENEActiv. Students are WRN16-1 (1-D CNNs). "Const." denotes constraints.

Fig. 6(c) and (d) with $\gamma_c = 0.02$. The best results are shown when $\eta = 700$. For PAMAP2, the smaller number of η (200) shows the best. When the window size is small and the number of channels is large, small η ($\leq$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.

3) Analysis of Constrained Adaptive Distillation: To consider the different feature-level properties of multiple teachers, the proposed method uses constrained adaptive weights based on entropy. To investigate the effects of the constrained adaptive distillation, we compare the results between those with and without constraints.

Fig. 7 shows the averaged probability by logits from models for testing samples of class 0 [walking (treadmill at 1 mi/h, 0% grade)] on GENEActiv, which are trained with time series and PIs. Since two models create completely different distributions, the difference in the ratio of entropy values between the two models is very large.

Evaluation results for training with or without constraints based on entropy are shown in Fig. 8. γ_c is 0.2 for WRN16-1 and WRN28-1 teachers and 0.02 for WRN16-3 and WRN28-3 teachers. 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.

4) Visualization of Feature Maps: To understand the details of activations for batch and channel similarities, both maps from teachers (WRN16-3) and a student (WRN16-1) are visualized in Figs. 9 and 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 and is the result of a model trained from scratch. The merged map is generated with constrained α by the 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.

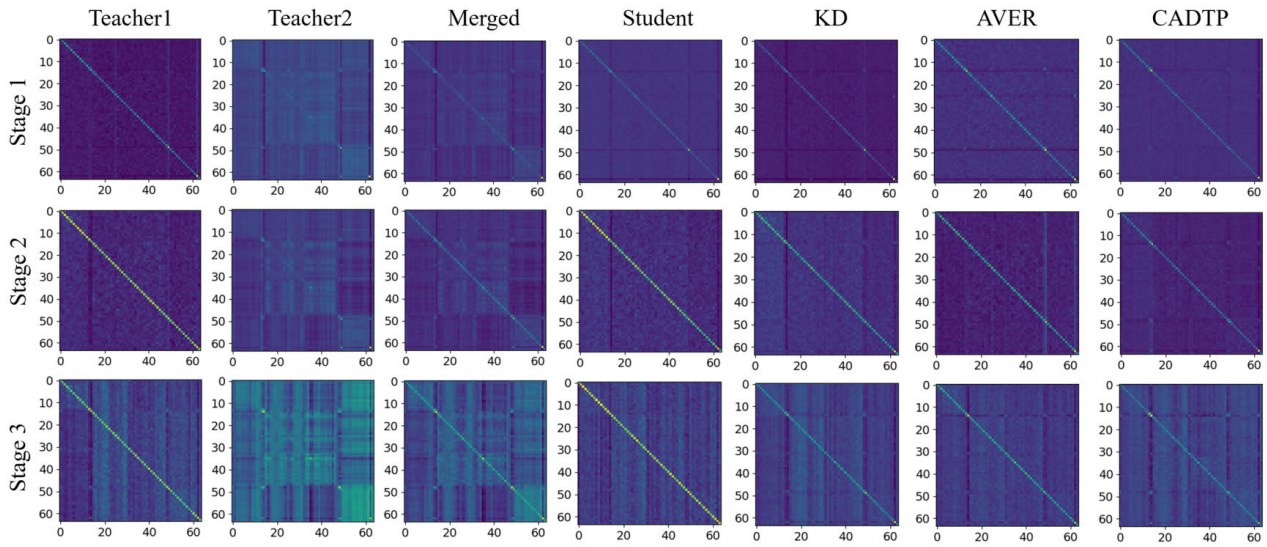


Fig. 9. Activation batch similarity maps produced by a layer for the indicated stage of the network for a batch on GENEActiv. High similarities for samples of the batch are represented with high values.

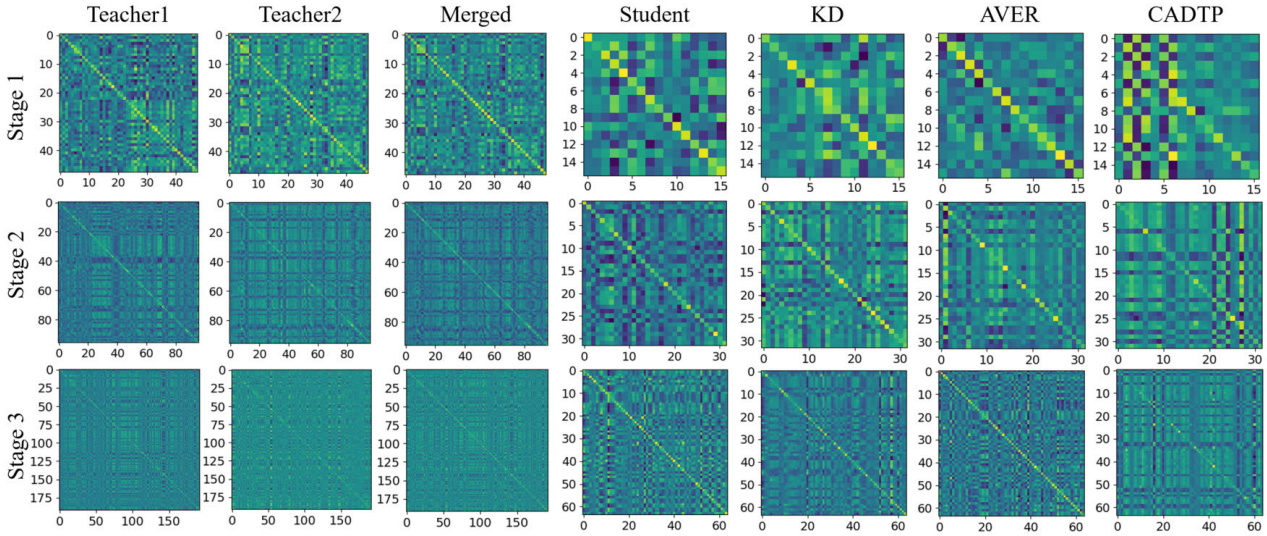


Fig. 10. Activation channel similarity maps produced by a layer for the indicated stage of the network for a batch on GENEActiv. High similarities for samples of the batch are represented with high values.

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

5) Analysis of Model Reliability: To explore the generalizability and regularization effects, we calculated the expected calibration error (ECE) [59] and negative log likelihood (NLL) [59]. ECE measures calibration error, which represents the reliability of the model. The probabilistic quality of a model can be computed by NLL. We used students trained by teachers of WRN16-3 and WRN28-1. ECE and NLL with various methods on GENEActiv and PAMAP2 are shown in Tables VIII and IX, 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 VIII

ECE (%) AND NLL FOR VARIOUS KD METHODS ON GENEACTIV. TEACHERS ARE WRN16-3 AND WRN28-1. STUDENTS ARE WRN16-1 (1-D 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 IX

ECE (%) AND NLL FOR VARIOUS KD METHODS ON PAMAP2. TEACHERS ARE WRN16-3 AND WRN28-1. STUDENTS ARE WRN16-1 (1-D 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**

TABLE X

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**		

E. Computational Time

We 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[1] Xeon[1] CPU E5-1650 v3), 48-GB memory, and an NVIDIA TITAN Xp graphic card (3840 NVIDIA[1] CUDA[1] cores and 12-GB memory) [60]. We evaluated approximately 6k samples with a batch size of 1. In Table X, the considered accuracy is the best one from Tables II and VI. 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 (1-D 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 highlights why a model compression method such as KD is needed for running on small devices with limited power and computational resources.

V. CONCLUSION

In this article, we proposed a new framework for constrained adaptive KD using topological representations on wearable sensor data, utilizing various similarity features and an annealing strategy. We demonstrated the proposed method, CADTP, with various combinations of teachers and student in classification. We 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 (GAFs) and Markov transition field (MTF)-based images encoded by time-series data. Finally, investigating the effects of augmentation methods on the image representations to leverage multiple teachers is also a potential avenue for further work.

[1]Registered trademark.

REFERENCES

[1] E. S. Jeon, H. Choi, A. Shukla, Y. Wang, M. P. Buman, and P. Turaga, "Topological knowledge distillation for wearable sensor data," in *Proc. 56th Asilomar Conf. Signals, Syst., Comput.*, Oct. 2022, pp. 837–842.

[2] H. Adams et al., "Persistence images: A stable vector representation of persistent homology," *J. Mach. Learn. Res.*, vol. 18, pp. 1–35, Jan. 2017.

[3] R. Turkeš, J. Nys, T. Verdonck, and S. Latré, "Noise robustness of persistent homology on greyscale images, across filtrations and signatures," *PLoS ONE*, vol. 16, no. 9, Sep. 2021, Art. no. e0257215.

[4] B. Rieck et al., "Uncovering the topology of time-varying fMRI data using cubical persistence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6900–6912.

[5] A. Som, H. Choi, K. N. Ramamurthy, M. P. Buman, and P. Turaga, "PI-Net: A deep learning approach to extract topological persistence images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 834–835.

[6] A. Nawar, F. Rahman, N. Krishnamurthi, A. Som, and P. Turaga, "Topological descriptors for Parkinson's disease classification and regression analysis," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2020, pp. 793–797.

[7] W. Guo, K. Manohar, S. L. Brunton, and A. G. Banerjee, "Sparse-TDA: Sparse realization of topological data analysis for multi-way classification," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1403–1408, Jul. 2018.

[8] F. Hensel, M. Moor, and B. Rieck, "A survey of topological machine learning methods," *Frontiers Artif. Intell.*, vol. 4, May 2021, Art. no. 681108.

[9] H. Edelsbrunner and J. L. Harer, *Computational Topology: An Introduction*. Providence, RI, USA: American Mathematical Society, 2022.

[10] Z. Chen, L. Zhang, Z. Cao, and J. Guo, "Distilling the knowledge from handcrafted features for human activity recognition," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4334–4342, Oct. 2018.

[11] K. Zhang et al., "Compacting deep neural networks for Internet of Things: Methods and applications," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 11935–11959, Aug. 2021.

[12] P. Qi, 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 J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 189–204, Jan. 2023.

[13] L. Cheng, 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 Trans. Instrum. Meas.*, vol. 72, pp. 1–10, 2023.

[14] E. S. Jeon, 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 Things J.*, vol. 9, no. 14, pp. 12848–12860, Jul. 2022.

[15] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021.

[16] C. Thai, 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," *Inf. Sci.*, vol. 616, pp. 330–347, Nov. 2022.

[17] S. Baghersalimi, 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," 2022, *arXiv:2208.00885*.

[18] S. Angarano, F. Salvetti, M. Martini, and M. Chiaberge, "Generative adversarial super-resolution at the edge with knowledge distillation," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106407.

[19] F. Remigereau, D. Mekhazni, S. Abdoli, L. T. Nguyen-Meidine, R. M. O. Cruz, and E. Granger, "Knowledge distillation for multi-target domain adaptation in real-time person re-identification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 3557–3853.

[20] Y. Liu, W. Zhang, and J. Wang, "Adaptive multi-teacher multi-level knowledge distillation," *Neurocomputing*, vol. 415, pp. 106–113, Nov. 2020.

[21] H. Zhang, D. Chen, and C. Wang, "Confidence-aware multi-teacher knowledge distillation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 4498–4502.

[22] S. Gholizadeh and W. Zadrozny, "A short survey of topological data analysis in time series and systems analysis," 2018, *arXiv:1809.10745*.

[23] S. Zeng, F. Graf, C. Hofer, and R. Kwitt, "Topological attention for time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24871–24882.

[24] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," *Discrete Comput. Geometry*, vol. 28, no. 4, pp. 511–533, Nov. 2002.

[25] D. Pachauri, C. Hinrichs, M. K. Chung, S. C. Johnson, and V. Singh, "Topology-based kernels with application to inference problems in Alzheimer's disease," *IEEE Trans. Med. Imag.*, vol. 30, no. 10, pp. 1760–1770, Oct. 2011.

[26] C. Bucilua, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2006, pp. 535–541.

[27] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[28] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4794–4802.

[29] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–13.

[30] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–13.

[31] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1365–1374.

[32] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1285–1294.

[33] K. Kwon, H. Na, H. Lee, and N. S. Kim, "Adaptive knowledge distillation based on entropy," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7409–7413.

[34] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[35] X.-S. Yang, *Nature-Inspired Optimization Algorithms*. New York, NY, USA: Academic, 2020.

[36] A. Jafari, M. Rezagholizadeh, P. Sharma, and A. Ghodsi, "Annealing knowledge distillation," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 2493–2504.

[37] Z. Dong, K. Hou, Z. Liu, X. Yu, H. Jia, and C. Zhang, "A sample-efficient OPF learning method based on annealing knowledge distillation," *IEEE Access*, vol. 10, pp. 99724–99733, 2022.

[38] K. Clark, M.-T. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, "BAM! Born-again multi-task networks for natural language understanding," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5931–5937.

[39] N. Saul and C. Tralie, "Scikit-TDA: Topological data analysis for Python," 2019, doi: 10.5281/zenodo.2533369.

[40] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[41] Q. Wang, 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 *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2016, pp. 2631–2635.

[42] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.

[43] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–12.

[44] Y. Lee, H. Kim, E. Park, X. Cui, and H. Kim, "Wide-residual-inception networks for real-time object detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 758–764.

[45] M. Song et al., "Frustration recognition from speech during game interaction using wide residual networks," *Virtual Reality Intell. Hardw.*, vol. 3, no. 1, pp. 76–86, Feb. 2021.

[46] K. Kania and U. Markowska-Kaczmar, "American sign language fingerspelling recognition using wide residual networks," in *Artificial Intelligence and Soft Computing*. Zakopane, Poland: Springer, Jun. 2018, pp. 97–107.

[47] D. Chen, J.-P. Mei, H. Zhang, C. Wang, Y. Feng, and C. Chen, "Knowledge distillation with the reused teacher classifier," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11933–11942.

[48] T. Huang, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge distillation from a stronger teacher," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 33716–33727.

[49] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[50] H. Choi, 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 349–357.

[51] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 1488–1492.

[52] S. Ha, J.-M. Yun, and S. Choi, "Multi-modal convolutional neural networks for activity recognition," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 3017–3022.

[53] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 381–388.

[54] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.

[55] C. Catal, S. Tufekci, E. Pirmit, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," *Appl. Soft Comput.*, vol. 37, pp. 1018–1022, Dec. 2015.

[56] H.-J. Kim, M. Kim, S.-J. Lee, and Y. S. Choi, "An analysis of eating activities for automatic food type recognition," in *Proc. Asia Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Dec. 2012, pp. 1–5.

[57] Q. Wen et al., "Time series data augmentation for deep learning: A survey," in *Proc. Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 4653–4660.

[58] X. Wang and C. Wang, "Time series data cleaning: A survey," *IEEE Access*, vol. 8, pp. 1866–1881, 2020.

[59] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1321–1330.

[60] NVIDIA. (2016). *NVIDIA Titan XP*. Accessed: Feb. 9, 2023. [Online]. Available: https://www.nvidia.com/en-us/titan/titan-xp/

Eun Som Jeon (Graduate Student Member, IEEE) received the B.E. and M.E. degrees in electronics and electrical engineering from Dongguk University, Seoul, South Korea, in 2014 and 2016, respectively. She is currently pursuing the Ph.D. degree in computer engineering (electrical engineering) with the Geometric Media Laboratory, Arizona State University, Tempe, AZ, USA.

She worked at Korea Telecom (Institute of Convergence Technology), Seoul. Her current research interests include time series and image data analysis, human behavior analysis, deep learning, and artificial analysis.

Hongjun Choi received the bachelor's and master's degrees in electronic engineering from Korea University, Seoul, South Korea, in 2014 and 2016, respectively, and the Ph.D. degree in electrical engineering from Arizona State University, Tempe, AZ, USA, in 2023.

He is currently a Post-Doctoral Researcher at the Lawrence Livermore National Laboratory, Livermore, CA, USA. His research interests are in the fields of machine learning, computer vision with applications in human activity analysis, and image classification using geometric methods.

Ankita Shukla received the master's and Ph.D. degrees in electronics and communication from IIIT-Delhi, New Delhi, India, in 2020 and 2014, respectively.

She is currently a Post-Doctoral Researcher at Arizona State University, Tempe, AZ, USA. Her research interests include machine learning, computer vision, time-series data analysis, and geometric methods.

Yuan Wang received the Ph.D. degree in statistics from the University of Wisconsin–Madison, Madison, WI, USA, in 2018.

She is currently an Assistant Professor of biostatistics with the Arnold School of Public Health, University of South Carolina, Columbia, SC, USA. Her methodological research has been focused on topological signal processing, inference, and learning, with applications to electroencephalographic signals and magnetic resonance imaging data in brain disorders such as epilepsy and poststroke aphasia.

Matthew P. Buman is currently a Professor with the College of Health Solutions, Arizona State University, Tempe, AZ, USA. His research interests reflect the dynamic interplay of behaviors in the 24-h day, including sleep, sedentary behavior, and physical activity. His work focuses on the measurement of these behaviors using wearable technologies, interventions that singly or in combination target these behaviors, and the environments that impact these behaviors.

Pavan Turaga (Senior Member, IEEE) received the bachelor's degree in electronics and communication engineering from IIT Guwahati, Guwahati, India, in 2004, and the master's and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 2007 and 2009, respectively.

He is currently a Professor with the School of Arts, Media and Engineering, Arizona State University, Tempe, AZ, USA. His research interests include computer vision and computational imaging with applications in activity analysis, dynamic scene analysis, and time-series data analysis with geometric methods.