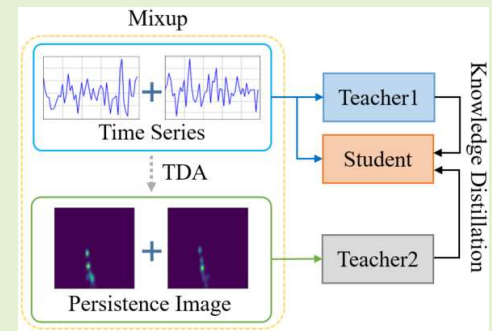


Role of Mixup in Topological Persistence Based Knowledge Distillation for Wearable Sensor Data

Eun Som Jeon, Hongjun Choi, Matthew P. Buman, and Pavan Turaga

Abstract—The analysis of wearable sensor data has enabled many successes in several applications. To represent the high-sampling rate time-series with sufficient detail, the use of topological data analysis (TDA) has been considered, and it is found that TDA can complement other time-series features. Nonetheless, due to the large time consumption and high computational resource requirements of extracting topological features through TDA, it is difficult to deploy topological knowledge in machine learning and various applications. In order to tackle this problem, knowledge distillation (KD) can be adopted, which is a technique facilitating model compression and transfer learning to generate a smaller model by transferring knowledge from a larger network. By leveraging multiple teachers in KD, both time-series and topological features can be transferred, and finally, a superior student using only time-series data is distilled. On the other hand, mixup has been popularly used as a robust data augmentation technique to enhance model performance during training. Mixup and KD employ similar learning strategies. In KD, the student model learns from the smoothed distribution generated by the teacher model, while mixup creates smoothed labels by blending two labels. Hence, this common smoothness serves as the connecting link that establishes a connection between these two methods. Even though it has been widely studied to understand the interplay between mixup and KD, most of them are focused on image based analysis only, and it still remains to be understood how mixup behaves in the context of KD for incorporating multimodal data, such as both time-series and topological knowledge using wearable sensor data. In this paper, we analyze the role of mixup in KD with time-series as well as topological persistence, employing multiple teachers. We present a comprehensive analysis of various methods in KD and mixup, supported by empirical results on wearable sensor data. We observe that applying mixup to training a student in KD improves performance. We suggest a general set of recommendations to obtain an enhanced student.

Index Terms—Knowledge distillation, wearable sensor data, time-series, topological persistence



I. INTRODUCTION

WEARABLE sensor data analysis has enabled many application by utilizing the power of deep learning. However, there are common challenges, such as inter- and intra-person variability, sensor-level noises, dependency on the sampling rate of the sensors, resulting in performance degradation and difficulties for deployment with machine learning. To mitigate these problems, topological data analysis (TDA) methods have been utilized on wearable sensor data analysis [1]–[3], which have resulted in many robust ways to capture detailed time-series information, and can be increasingly applied to many different areas. TDA methods allow for capturing and preserving shape-related information and

have the potential to make sensor data processing pipelines more robust to different types of time-series corruptions [4]–[6]. Topological features can be represented in many ways [7], [8], a common approach is referred to as the persistence image (PI) – which can aid in easily deploy topological persistence in machine learning owing to its 2D image-like form. Prior research has found that persistence images provide additional information that complements the raw time-series data to improve performance in time-series classification problems on wearable sensor data [2], [3], [9]. Applications of topological methods also have touched upon many areas particularly in sensor data analysis [10]–[12].

Although TDA has shown great promise, leveraging topological features by TDA on edge-devices including wearable devices, particularly implementing them on small form factor and memory limited devices, is difficult because of large computational resources and time consumption requirements to extract the topological features [4], [13]. Also, previous studies implement separate models in test-time simultaneously to utilize topological as well as time-series data to improve performance [2], which can increase the complexity of a model. Based on this insight, new methods to create a unified model for maximizing efficiency and integration of topological

E. Jeon is with the Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea, email: (ejeon6@seoultech.ac.kr).

H. Choi is with Lawrence Livermore National Laboratory, Livermore, CA 94550, USA, email: (choi22@llnl.gov).

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

P. Turaga is with the Geometric Media Lab, School of Arts, Media and Engineering and School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85281 USA, email: (pturaga@asu.edu).

features is required.

To address these issues, knowledge distillation (KD) can be adopted as a solution, which generates a small and superior model by transferring knowledge from a large network model. Furthermore, it enables to leverage multimodal data to distill a robust single model. With KD, a teacher trained with topological features can be utilized to provide more diverse information to a student while complementing time-series features. With multiple teachers trained with the raw time-series and topological representations, a single and superior student, using the time-series data alone, can be distilled [3].

In KD, the temperature hyperparameter plays a key role in learning process, which controls the smoothness of distribution and determines the difficulty level of the distillation process. In this context, recently, many studies have delved into the impact of mixup augmentation in KD [14]–[18]. Particularly, for image analysis, Choi *et al.* [15] explored the interplay of mixup with KD and revealed that smoothness serves as the connecting link to understand the effect of mixup in KD. For more details, in KD, the student learns from the smoothed distribution provided by the teacher model, and this distribution is further smoothed by increasing the temperature value. Similarly, mixup generates new smooth labels by blending two given inputs and ground truth labels, which are then further smoothed by strongly interpolated samples (e.g., a high alpha value in the beta distribution). Thus, their behave as a connecting link for promoting smoothness in learning process, which can generate synergetic effects to distill a robust lightweight model [15], [17].

There are different augmentation methods such as regularization effect [19], model invariance [20], and feature learning [21]. However, these techniques are more focus on alleviating noises or data point issues in rotation, which are different from mixup [22] blending multiple samples. Further, even if other augmentations (e.g. cutmix [23] and adversarial training [24]) are effective, mixup offers different benefits in much lower computational overhead and provides solid foundations, particularly in the context of knowledge distillation [25]–[27].

Even though the interplay between two techniques, mixup and KD, is significantly crucial in performance improvement, the majority of previous studies have primarily concentrated on image-based analysis. To the best of our knowledge, the impact of mixup and KD in the context of both time-series and topological representations on wearable sensor data remains unexplored. Furthermore, the behavior of mixup for multiple teachers and different strategies in KD have not been investigated.

In this paper, we study the behavior of mixup in KD with multimodalities using both time-series and topological representations for wearable sensor data analysis. We implement different KD approaches for utilizing time-series as well as topological persistence to train a student. We investigate whether the mixup method can enhance the performance of topological persistence-based KD using various teachers. Additionally, we compare the performance of using mixup in KD to determine if leveraging both representations yields more benefits than relying solely on time-series data.

The contributions of this paper are summarized below:

- We analyze the interplay between mixup and KD for wearable sensor data, and compare different strategies in KD with single-teacher and multiple-teacher based distillation, leveraging time-series as well as topological persistence.
- We study the effects of mixup on training both teacher and student models. We aim to identify which training strategy for utilizing mixup in KD provides the most benefit in the activity classification task and explore whether the effects of mixup are comparable to those of other time domain augmentation methods in KD.
- Through the analysis of multiple strategies for employing mixup with multiple teachers, we propose improved learning approaches by regulating smoothness through temperature and the number of mixup pairs.

The rest of the paper is organized as follows. In section II, we describe mixup and KD techniques with persistence image. In section III, we explain strategies to leverage topological persistence with mixup in KD. In section IV, we present our experimental results and analysis. In section VI, we discuss our findings and conclusions.

II. BACKGROUND

A. Mixup Augmentation

Mixup augmentation [28] is used commonly in deep-learning techniques to alleviate issues of memorization and sensitivity to adversarial examples. Two examples drawn at random from training data are mixed by linear interpolation [28]. Let the training data be $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, where n is the number of samples. Input data is $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and its corresponding label is $y \in \mathcal{Y} = \{1, 2, \dots, K\}$. The sampling process for mixup can be written as follows:

$$\begin{aligned}\tilde{x}_{ij}(\lambda) &= \lambda x_i + (1 - \lambda)x_j, \\ \tilde{y}_{ij}(\lambda) &= \lambda y_i + (1 - \lambda)y_j,\end{aligned}\quad (1)$$

where $\lambda \in [0, 1]$ follows the distribution P_λ where $\lambda \sim \text{Beta}(\alpha, \alpha)$. λ is to specify the extent of mixing. The hyperparameter α controls the strength of interpolation between feature-target pairs. α generates strongly interpolated samples. To train a function f , the following mixup loss function is minimized:

$$\mathcal{L}_{mix}(f) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}_{\lambda \sim P_\lambda} [\mathcal{L}_{CE}(f(\tilde{x}_{ij}(\lambda)), \tilde{y}_{ij}(\lambda))], \quad (2)$$

where \mathcal{L}_{CE} is a standard cross-entropy loss function.

Many different variants of mixup have been studied [23], [29], [30]. Intrinsically, these methods have similarities in that they mix the input data (e.g. images) and labels proportionally to extend the training distribution. The benefits of mixup with time-series data were explored in previous studies [31]–[33]. In this study, we use the conventional mixup to explore the effects on knowledge distillation [28] for time-series data.

B. Persistence Image

TDA has been applied in various fields [4], [34]–[36], which can characterize the shape of raw data. One important tool

in TDA is persistent homology, which provides a multiscale description with topological features. When applied to point clouds, these features are often described as cavities characterized by points, triangles, and edges by filtration [8], [37]. The extension to time-series data is via sub level-set filtrations, where level-sets are tracked. The birth and death times of topological features can be represented as a multiset of points in a persistence diagram (PD). Since the number and locations of the points in PDs vary depending on the underlying data, it is difficult to use them directly in machine learning pipelines. To project the features on the stable vector representation, a persistence image can be used, mapping the scatter points based on their persistence value (life time) [4]. Firstly, PD is mapped to an integrable function $\rho : \mathbb{R} \rightarrow \mathbb{R}^2$, called a persistence surface (PS), which is defined as a weighted sum of Gaussian functions. A PI can be created by integrating PS on a grid box that is defined by discretization. The values of PI represent the persistence points of the PD. The example of PD and PI are shown in Fig. 1. Even though TDA can provide additional information to the raw time-series to improve performance, it is challenging to run the method on a resource constrained devices, because extracting PIs by TDA requires a large amount of time and memory. To solve this problem, in this paper, we adopt knowledge distillation that distills a single student utilizing the raw time-series data alone as an input.

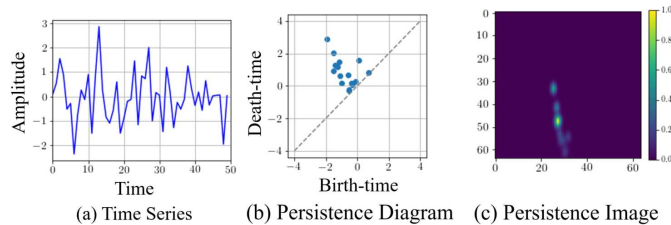


Fig. 1: time-series data and its corresponding PD and PI. Higher persistence in PD is represented with brighter color in PI.

C. Knowledge Distillation

Knowledge distillation trains a smaller (student) model from a larger (teacher) model [38], [39]. The student model is trained by minimizing the difference between its outputs and soft labels, called relaxed knowledge, from a teacher, which improves performance beyond using hard labels (labeled data) alone. The loss function of standard knowledge distillation [39] is:

$$\mathcal{L} = (1 - \tau)\mathcal{L}_{CE}(\sigma(t_s), y_g) + \tau\mathcal{L}_{KD}(f_T, f_S), \quad (3)$$

where t_s is logits of a student model f_S , f_T is a teacher model, y_g is a ground truth label, $\sigma(\cdot)$ is a softmax function, $\mathcal{L}_{KD}(\cdot)$ is KD loss function, and τ is hyper-parameter; $0 < \tau < 1$. The difference between the outputs of the student and the teacher is mitigated by employing Kullback-Leibler divergence loss function, which is described as follows:

$$\mathcal{L}_{KD}(f_T, f_S) = \frac{\tau^2}{n} \sum_{i=1}^n KL(\sigma(\frac{f_T(x_i)}{\mathcal{T}}), \sigma(\frac{f_S(x_i)}{\mathcal{T}})), \quad (4)$$

where $KL(\cdot)$ measures Kullback-Leibler divergence loss, \mathcal{T} is a hyper-parameter, temperature, to smooth the outputs. To obtain the best performance, in this paper, we utilize a teacher trained by early stopping the training process in KD [40].

Not only logits, but also features from intermediate layers can be utilized to knowledge transfer, which is called feature-based distillation [41]. Attention transfer (AT) has been widely used, which uses attention maps extracted by a sum of squared attention mapping function [42]. Tung *et al.* [43] extracts similarities within a mini-batch of samples from a teacher and a student, where those maps have to be matched in distillation process. Even though various techniques have been utilized to improve the performance, they typically address single-modal issues with a single teacher.

Multiple teachers can be utilized to provide more and diverse knowledge to a single student [3], [41], [44], [45]. Using a uni-modal data with different teachers, a student can establish its own knowledge by integrating diverse knowledge from the teachers [46]. However, in some cases, data samples or labels used for training a teacher cannot be leveraged to train or test a student [41]. Jeon *et al.* [3] utilize multiple teachers to train a single student by transferring features from both the persistence image and the raw time-series data. Even though two teachers have different architectural designs and use different types of inputs, their logit information can be transferred with KD loss that can be written as:

$$\mathcal{L}_{KDm}(f_{T_1}, f_{T_2}, f_S) = \eta\mathcal{L}_{KD}(f_{T_1}, f_S) + (1 - \eta)\mathcal{L}_{KD}(f_{T_2}, f_S), \quad (5)$$

where η is a hyper-parameter to control the effects from different teachers, and f_{T_1} and f_{T_2} are teacher models trained with time-series data and PIs, respectively. Then, the total loss function can be written as:

$$\mathcal{L}_m = (1 - \tau)\mathcal{L}_{CE}(\sigma(t_s), y_g) + \tau\mathcal{L}_{KDm}(f_{T_1}, f_{T_2}, f_S). \quad (6)$$

For further improvement in KD, mixup augmentation methods have been widely studied. Specifically, mixup and KD share a common thread in serving smoothness during the training process. To accommodate synergetic effects, the interest in the interplay between mixup and KD grows, which has been analyzed in many studies [14]–[18]. However, most of the studies were conducted with image data only. It is still required to be explored with time-series and multimodalities using different representations. Based on these insights, we investigate the effects of mixup in KD for time-series on wearable sensor data by utilizing a single or multiple teachers. Also, we present compatible or incompatible views through an empirical analysis.

III. ANALYSIS STRATEGIES FOR MIXUP IN KD

To analyze the effect of mixup in persistence based KD, we utilize different approaches that are explained in this section.

A. Leveraging Topological Persistence

1) *Leveraging A Single Teacher*: With the process of standard knowledge distillation, a single teacher trained with PIs can be used to transfer knowledge to a student, as illustrated in

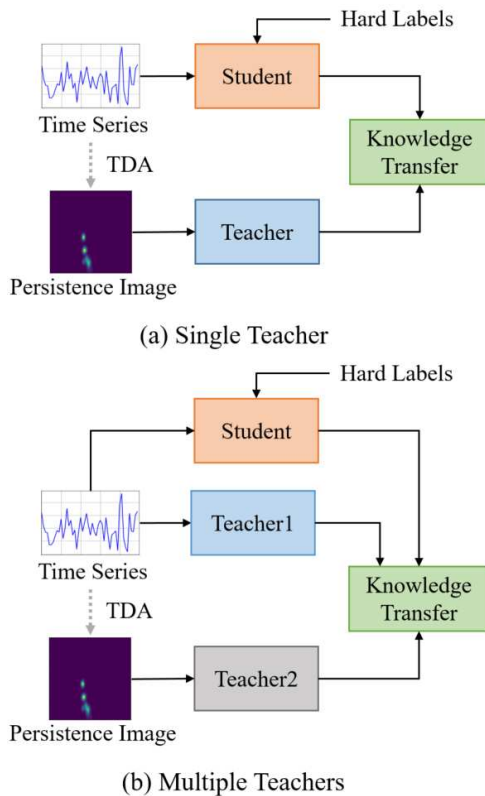


Fig. 2: Strategies to leverage topological persistence in KD. (a) utilizes a single teacher trained with PIs. (b) uses different teachers trained with PIs and the raw time-series data, respectively.

Fig. 2(a). PIs are generated by TDA from the raw time-series data. PIs are 2D images, so the teacher model consists of a 2D kernel of CNNs. To train a student with time-series (1D) data, 1D CNNs can be used. Logit of the teacher and student is leveraged to transfer knowledge.

2) Leveraging Multiple Teachers: Multiple teachers can be used to train a single student. For instance, two teachers, trained with time-series and PIs, can transfer knowledge simultaneously, as described in Fig. 2(b). The student utilizes time-series alone as an input. In this way, the student can obtain benefits from both of these different features, but it still requires only time-series implementation at test time. Since two teachers are trained with different modalities and have different architectural designs, it is difficult to create a unified model and knowledge gap making performance degradation can be produced [41]. To mitigate this issue, we adopt an annealing strategy that trains a student by initializing weight values from a model learned from scratch [3].

B. Mixup Strategy in KD

We set different strategies to utilize mixup in KD, as described in Fig. 3. Details are explained as follows.

- **Mixup for learning from scratch:** To investigate the effects of mixup on time-series, we compare mixup- and non-mixup trained models.

- **Mixup in KD:** To explore the connecting link between mixup and KD, we train a student model with mixup and different temperatures, using various methods in KD.
- **Mixup-trained teacher and student:** We apply mixup not only to a student but also to teachers to figure out the effects of the augmentation method in KD. With different combinations of applying mixup, we investigate which strategy is effective in KD.
- **Distillation with different temperature and partial mixup:** To analyze the effects of smoothness from temperature on mixup in KD, a student is trained with the augmentation method and different temperature parameters. In this way, we figure out how much temperature impacts the performance of mixup in KD. Also, to analyze the smoothness of mixup, we utilize partial mixup (PMU) that uses only a few mixup pairs in a batch, as addressed in the previous study [15]. The method uses small amounts of mixup pairs to control the strength of smoothness, which alleviates excessive smoothness.
- **Mixup for different teachers:** Two teachers generate different knowledge and effects for a student in distillation. To explore the effects of mixup for different modalities, we apply different hyper-parameters to teachers. The training objective for the student in KD with multiple teachers and different mixup hyper-parameters is as follows:

$$\min_{\mathbb{E}_{(x,y) \sim \mathcal{D}}} [\mathbb{E}_{\lambda_1 \sim P_{\lambda_1}} [\eta \{ (1 - \tau) \mathcal{L}_{mix}(f_S) + \tau \mathcal{L}_{KD}(f_{T_1}, f_S) \}] + \mathbb{E}_{\lambda_2 \sim P_{\lambda_2}} [(1 - \eta) \{ (1 - \tau) \mathcal{L}_{mix}(f_S) + \tau \mathcal{L}_{KD}(f_{T_2}, f_S) \}]] , \quad (7)$$

where λ_1 and λ_2 are to specify the extent of mixing, whose α parameters are different.

In Table I, we provide the floating point operations per second (FLOPs) with networks and processing time for an epoch with batch size of 64 in training process for strategies in Fig. 3. The processing time is measured 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. As explained in the table, Strategy (e) takes the longest time and larger complexity compared to other strategies. Through the training, all of strategies distill the same sized single student even though each strategy is different. In test-time, a single student model is implemented alone, which corresponds to the Student.

More details of settings and experimental results for each strategy are explained in section IV.

IV. EXPERIMENTS

In this section, we describe datasets and implementation details. We utilize various strategies of KD and mixup to investigate the effects on wearable sensor data analysis. We analyze optimized solutions and describe ablations.

A. Dataset Description and Implementation Details

1) Dataset Description: We analyze the strategies with wearable sensor data on GENEActiv and PAMAP2 datasets. These datasets consist with diverse window size and number of

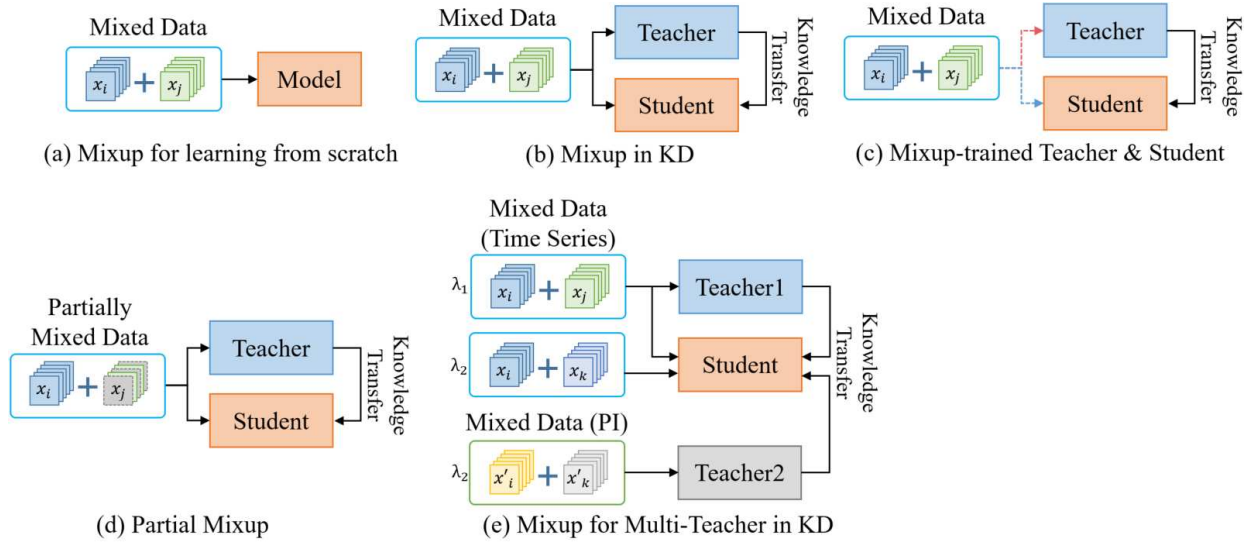


Fig. 3: Approaches for incorporating mixup in KD.

TABLE I: Details of efficiency for different training strategies with mixup and KD, which are explained in Fig. 3. Teachers are WRN16-3 and Student is WRN16-1.

Strategy	GFLOPs		Processing Time (sec)
	Teacher	Student	
(a)	—	0.71	4.54
(b)	6.02		8.19
(c)			8.50
(d)			

Strategy	GFLOPs			Processing Time (sec)
	Teacher1	Teacher2	Student	
(e)	6.02	57.55	0.71	22.48

channels obtained from multiple sensors on different activities. Thus, experiments on these datasets aid in showing various evaluations under different conditions, which helps to explain generalizability and applicability of methods.

GENEActiv. GENEActiv dataset [47] was collected by GENEActiv sensor, using waterproof, a light-weight and worn tri-axial accelerometer. The sampling frequency was 100 Hz. By referring to the previous study [3], [48], we select 14 daily activities for analysis, such as walking, standing, and sitting. Each class has over 9 hundred samples with 500 time steps of window size, corresponding to 5 seconds with full-non-overlapping sliding windows. The number of subjects for training and testing is 130 and 43, respectively, and the number of samples is around 16k and 6k, respectively.

PAMAP2. PAMAP2 dataset [49] was recorded from heart rate, temperature, accelerometers, gyroscopes, and magnetometers, which include 3 Colibri wireless inertial measurement units (IMU). The sampling frequency was 100 Hz for 9 subjects. The recordings are downsampled to 33.3Hz by referring to the previous study [48], [50]. A window size for a sample is 100 time steps or 3 seconds with 22 time steps for segmenting the sequences, which allows semi-non-overlapping sliding windows with 78% overlapping [49]. We use 12 daily activities including lying, sitting, walking, etc. For evaluation

in experiments, we use leave-one-subject-out combinations.

2) Implementation Details: We use the Scikit-TDA python library [51] and the Ripser package to produce PDs and extract PIs [2]. For GENEActiv, the standard deviation for the Gaussian kernel is set to 0.25 and the birth-time range of PI is $[-10, 10]$, respectively, as do the same in the previous studies [2], [3]. For PAMAP2, the parameter for Gaussian kernel is 0.015 and the range for PI is $[-1, 1]$, respectively. Each PI is generated from each channel and the values are normalized by its maximum intensity value. The size of PI is set to 64×64 . For training models, we set the total number of epochs as 200, SGD with momentum of 0.9, a weight decay of 1×10^{-4} , and batch size for 64. To train a model with time-series data (1D data), 1D convolutional layers are utilized. The initial learning rate is 0.05 that decreases by 0.2 at 10 epochs and drops by 0.1 every $\lceil \frac{e}{3} \rceil$ where e is the total number of epochs. A model using image representation for PIs consists of 2D convolutional layers. The initial learning rate is 0.1 that drops by 0.5 at 10 epochs and by 0.2 at every 40 epochs. We measure the performance with WideResNet (WRN) [52] that is popularly utilized in the validation of KD [3], [40], [48]. For default settings, we set τ , η , and \mathcal{T} as 0.7, 0.7, and 4 for GENEActiv, and 0.99, 0.3, and 4 for PAMAP2, referring to the previous study [3], [48] and to consider best performance. We run 3 times and report the averaged accuracy and standard deviation. As a baseline, we implement standard KD [39], attention transfer (AT) [53], and similarity-preserving knowledge distillation (SP) [43], which utilize logit as well as feature from intermediate layers for distillation. Parameters for AT and SP are set as 1500 and 1000 for GENEActiv, and 3500 and 700 for PAMAP2, respectively. A simple knowledge distillation (SimKD) [54] and DIST [55] leveraging intra- and inter-class relations for knowledge transfer are also used as baselines. Also, multi-teacher based approaches such as AVER [46], EBKD [56], and CA-MKD [45], Base [3] are used for baselines. Since two teachers are incorporated with different dimensional layers, only logits are used for distillation of

baselines. When mixup is applied, α is 0.1 for both datasets.

B. Preliminary: Effects of Topological Persistence in KD

In this section, as preliminaries, we conduct experiments with a single and multiple teacher based distillation methods. For multiple teacher based methods, we train models with time-series as well as PIs by leveraging topological persistence. Teachers and students are trained with the various KD strategies explained in the previous section. Note, “TS” and “Ann.” denote using time-series data to train a student model and using two teachers in KD and an annealing strategy [3], respectively. Teacher1 and Teacher2 are teachers trained with time-series and persistence images, respectively.

TABLE II: Accuracy (%) with various knowledge distillation methods on GENEActiv.

Teacher1 (1D CNNs)	Teacher2 (2D CNNs)	Student (1D CNNs)	TS KD	PI KD	TS+PI Base	TS+PI Ann.
WRN16-1 (0.06M, 67.66)	WRN16-1 (0.2M, 58.64)	WRN16-1 (0.06M 67.66)	69.71 ± 0.38	67.83 ± 0.17	69.09 ± 0.37	70.15 ± 0.03
WRN16-3 (0.5M, 68.89)	WRN16-3 (1.6M, 59.80)		69.50 ± 0.10	68.79 ± 0.73	69.24 ± 0.62	70.71 ± 0.12
WRN28-1 (0.1M, 68.63)	WRN28-1 (0.4M, 59.45)		68.32 ± 0.63	68.51 ± 0.01	69.55 ± 0.41	70.44 ± 0.10
WRN28-3 (1.1M, 69.23)	WRN28-3 (3.3M, 59.69)		68.01 ± 0.69	68.46 ± 0.28	69.42 ± 0.58	69.97 ± 0.06

TABLE III: Accuracy (%) for related methods on GENEActiv with 7 classes. For KD, teachers are WRN16-3 and students are WRN16-1.

Method		Window length	
		1000	500
TS	Student	89.29 ± 0.32	86.83 ± 0.15
	SVM [57]	86.29	85.86
	Choi <i>et al.</i> [58]	89.43	87.86
	KD	89.88 ± 0.07	88.16 ± 0.15
	AT	90.32 ± 0.09	87.60 ± 0.22
	SP	88.47 ± 0.19	87.69 ± 0.18
	DIST	90.20 ± 0.39	87.05 ± 0.31
TS+PI	SimKD	90.47 ± 0.32	88.16 ± 0.37
	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

As described in Table II, for GENEActiv, Ann. using multiple teachers shows the best in all cases. Among different combinations, WRN16-3 teachers distill a superior student. To compare with previous studies, we tested a combination of teachers (WRN16-3) and students (WRN16-1) on GENEActiv utilizing different window length for 7 classes, where the combination showed the best in past studies [3], [40], [48]. As shown in Table III, Ann. outperforms previous methods. Also, as summarized in Table IV and V, for PAMAP2, Ann. outperforms methods using a single teacher and previous methods. WRN16-3 teachers for Ann. produce best performance. This represent that considering coherent characteristics of a student

TABLE IV: Accuracy (%) with various knowledge distillation methods on PAMAP2.

Teacher1 (1D CNNs)	Teacher2 (2D CNNs)	Student (1D CNNs)	TS KD	PI KD	TS+PI Base	TS+PI Ann.
WRN16-1 (0.06M, 85.27)	WRN16-1 (0.2M, 86.93)	WRN16-1 (0.06M, 82.99)	85.96 ± 2.19	85.04 ± 2.58	85.91 ± 2.32	86.09 ± 2.33
WRN16-3 (0.5M, 85.80)	WRN16-3 (1.6M, 87.23)		86.50 ± 2.21	86.68 ± 2.19	86.18 ± 2.37	87.12 ± 2.26
WRN28-1 (0.1M, 84.81)	WRN28-1 (0.4M, 87.45)		84.92 ± 2.45	85.08 ± 2.44	85.54 ± 2.26	85.89 ± 2.26
WRN28-3 (1.1M, 84.46)	WRN28-3 (3.3M, 87.88)		86.26 ± 2.40	85.39 ± 2.35	86.04 ± 2.34	86.33 ± 2.30

TABLE V: Accuracy (%) for related methods on PAMAP2. For KD, teachers are WRN16-3 and students are WRN16-1.

Method		Accuracy
TS	Student	82.81 ± 2.51
	Chen and Xue [59]	83.06
	Ha <i>et al.</i> [60]	73.79
	Ha and Choi [61]	74.21
	Catal <i>et al.</i> [62]	85.25
	Kim <i>et al.</i> [63]	81.57
	KD	86.38 ± 2.25
TS+PI	AT	84.44 ± 2.22
	SP	84.89 ± 2.10
	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

is important to improve performance. Specifically, training a student from weights of learning from scratch helps to alleviate the knowledge gap that makes it difficult to transfer knowledge to a student from multiple teachers. These results show that topological features implement time-series to improve the performance.

TABLE VI: Accuracy (%) for different structure of teachers on GENEActiv.

Method	Architecture Difference			
	Depth		Width	
Teacher1 (1D CNNs)	WRN 16-1 (0.06M, 67.66)	WRN 28-1 (0.2M, 68.63)	WRN 28-1 (0.1M, 68.63)	WRN 28-3 (1.1M, 69.23)
Teacher2 (2D CNNs)	WRN 28-1 (0.1M, 59.45)	WRN 16-1 (0.2M, 58.64)	WRN 28-3 (3.3M, 59.69)	WRN 28-1 (0.4M, 59.45)
Student (1D CNNs)				
WRN16-1 (0.06M, 67.66 ± 0.45)				
Base	68.71 ± 0.36	67.89 ± 0.27	68.26 ± 0.13	69.09 ± 0.59
	69.95 ± 0.05	70.34 ± 0.14	70.28 ± 0.08	69.95 ± 0.07

Leveraging heterogeneous teachers. We conducted experiments with heterogenous structure of teachers. As illustrated in Fig. VI, one better teacher does not guarantee a better student, which corroborates the previous studies [40]. Even though

teachers have heterogeneous structures, they complement each other to improve the performance, which is shown with better performance than a model learned from scratch (Student).

C. Effect of Mixup in KD

In this section, we explore effects of mixup for learning from scratch and KD, which provides smoothness in training process. To analyze the interplay of mixup and KD, we utilize response based KD methods, including Base and Ann., which does not require to use additional weights and aids in more prominently showing the effects of interplay with mixup. Firstly, we train a model from scratch with mixup. Secondly, we train a student in KD with mixup. Also, to see the effects of smoothness by temperature in KD, we train students with different temperatures.

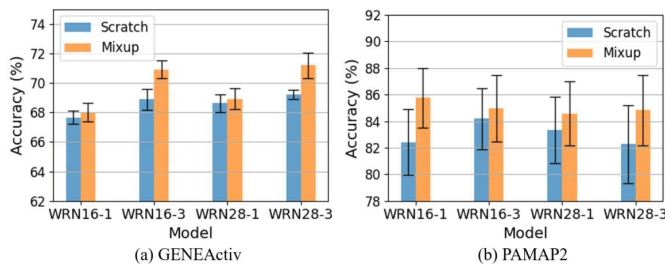


Fig. 4: Results of various models trained from scratch with or without mixup.

We trained various models from scratch with mixup, as illustrated in Fig. 4. In all cases, models trained with mixup show better performance. In Fig. 5, we show the results of various models trained with KD and mixup. WRN16-1 is used as a student. Mixup is applied to train a student in KD. In overall cases, with mixup generates better results. However, in some cases, the performance is worse than without mixup. This implies that mixup affects differently in KD compared to learning from scratch. Specifically, significant characteristics of input data, such as peaky points within a sample, can be softened because of blending different data for mixup, which was similar to results of injecting smoothness as addressed in previous study [48]. In all cases, Ann. shows better performance when mixup is added. This represents that topological features can complement time-series features to improve the performance. In details, persistence image representation can aid to preserve significant information, which generates synergistic effect with time-series features for classification. We also trained models with different temperature hyper-parameters that can generate a smoothness effect for knowledge transfer. As shown in Fig. 6, when $T=12$, all cases show the best. Therefore, temperature can significantly affect to performance in KD.

We plot t-SNE with a WRN16-3 teacher and WRN16-1 student and measure the V-Score [64] of outputs from the penultimate layers in Fig. 7. V-score is a metric to evaluate clustering, implying that a higher value is better clustering. For GENEActiv, classes from 0 to 5 are walking or running at different speeds. Class 7, 8, and 9 are activities related to hand motions such as brushing teeth and driving a car.

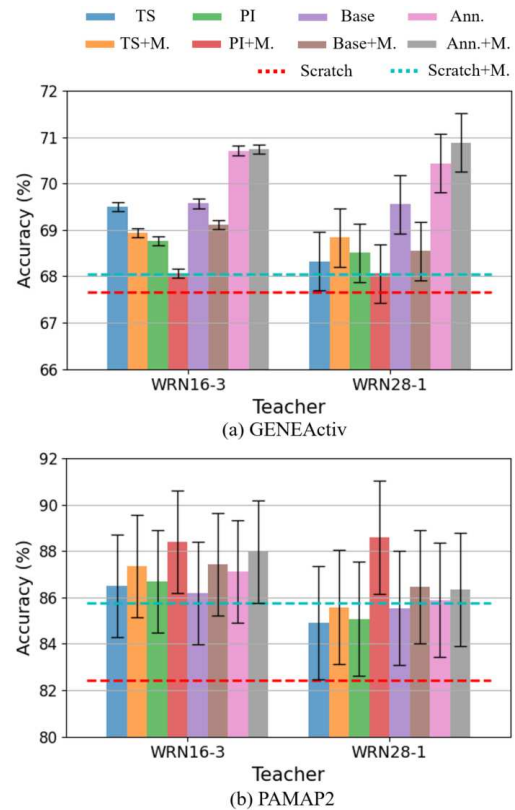


Fig. 5: Results of various models trained with KD and mixup. TS and PI are results of students trained with KD. M. denotes using mixup.

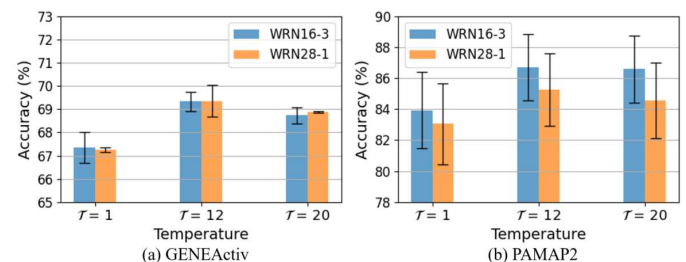


Fig. 6: Results of various models with different temperature in KD.

Class 12 and 13 are walking up and down stairs, respectively. When a student is trained with mixup, it generates a higher V-Score, compared to Student that is trained from scratch and results with conventional KD. Also, more distance between classes can be observed, which is measured with the V-Score and shown with the distance of the center point of the classes, particularly the gap between class 7, 8, and 9. In addition, some compacted points became more sparse, which is illustrated with class 12. For temperature, a high value of temperature provides more smoothness (soft knowledge) in KD, which can increase V-Score. When T is 12, the result shows the best, where the result is similar to the one of KD with mixup. When T is 1, the result is worse than learning from scratch. Thus, smoothness can affect the performance of KD at large. Based on these results, we can observe that

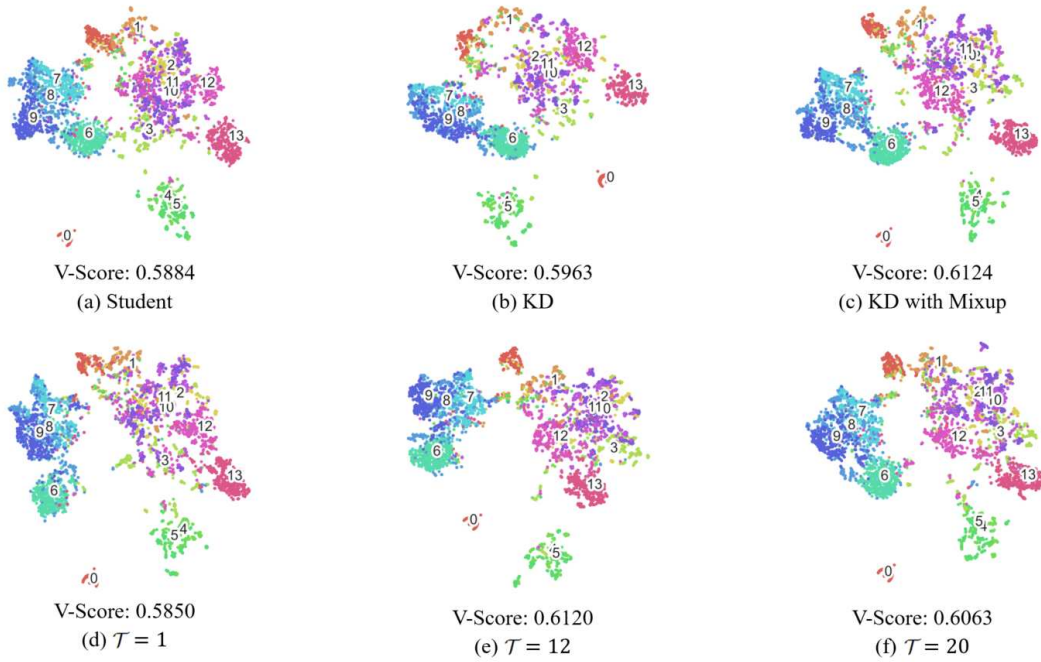


Fig. 7: t-SNE plots of output for various models on GENEActiv. A teacher is WRN16-3 and a student is WRN16-1, which are trained with time-series data. “Student” is a model learned from scratch.

injecting smoothness plays a key role in KD. That is, both mixup and temperature can significantly affect performance in distillation with generating soft knowledge, which can generate a synergistic effect to improve performance.

Augmentations in KD. Additionally, we conducted experiments with different augmentation methods (cutout [65] and cutmix [23]) in KD. The hyperparameter of cutout is 0.2. As explained in Table VII, all augmentations show improved results for learning from scratch. However, with KD, mixup only achieves improvement while other augmentations show degradation. This corroborates the benefits of mixup in KD, explored in prior studies [14]–[18], [25]–[27].

TABLE VII: Accuracy (%) for different augmentations methods on GENEActiv. LS denotes learning from scratch.

Method	Student	Mixup	Cutout	Cutmix
LS	67.66±0.45	68.04±0.63 (0.38↑)	68.67±0.64 (1.01↑)	68.70±0.94 (1.04↑)
KD (WRN16-1)	69.71±0.38	69.82±0.24 (0.11↑)	65.79±0.63 (3.92↓)	65.75±0.65 (3.96↓)
KD (WRN28-1)	68.32±0.63	68.84±0.23 (0.52↑)	65.03±0.81 (3.29↓)	66.18±0.44 (2.14↓)

D. Teacher-Student with Mixup

To explore the effect of mixup-trained teachers as well as students, we set various combinations of using the augmentations in KD. Note, “T”, “S”, “mT” and “mS” denote a teacher model, a student model, a mixup-trained teacher model, and using mixup to train a student model. As explained in previous sections, WRN16-3 teachers generated a superior student compared to other combinations. On the other hand, WRN28-1 model learned from scratch showed less improvement with

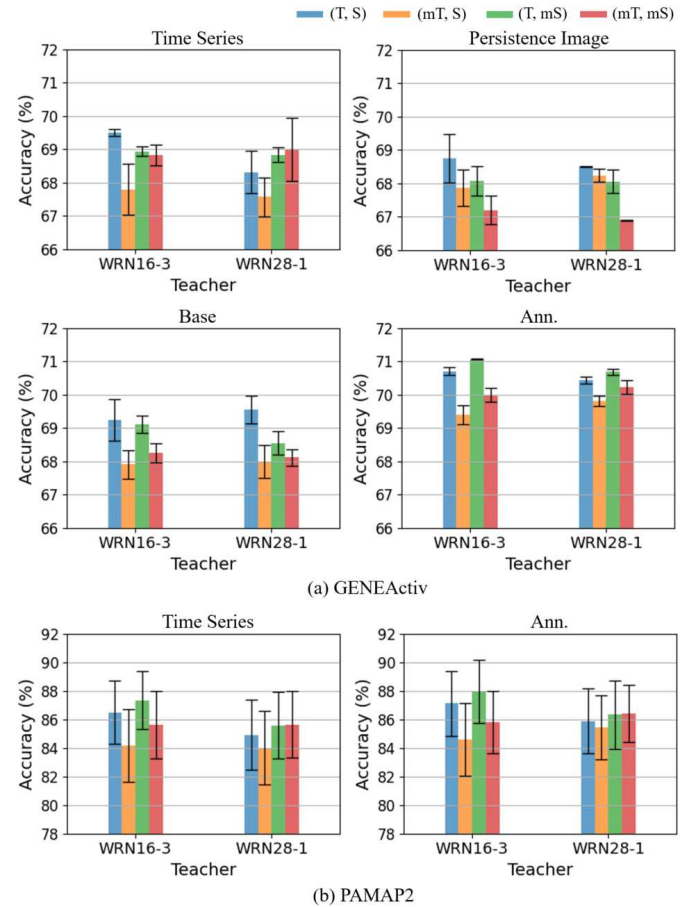


Fig. 8: Results of various approaches in KD, trained with mixup. Brackets denote (Teacher, Student).

mixup than other capacity of models. For further analysis with mixup in KD, we use WRN16-3 and WRN28-1 for teachers and WRN16-1 for a student to consider different depth and width combinations of teacher-student networks and different effects on mixup in KD. As shown in Fig. 8, Ann. shows the best among different approaches in KD. Students distilled by using PI alone and Base show worse performance than the one learned from scratch without using mixup. For Ann., when teachers are trained without mixup and a student is trained with mixup (T, mS), the student outperforms learning from scratch and other combinations of teacher-student trained with/without mixup. These results represent that reducing knowledge gap with an annealing strategy (Ann.) is effective for applying mixup in KD to train a student with multiple teachers. Also, soft knowledge of topological persistence provided by mixup indeed aid to train a student. In addition, this result corroborates the fact that the effects of mixup are similar to those of time domain augmentation methods, such as Gaussian noise, providing smoothness in KD, as analyzed in the previous study [48].

E. Analysis of the Effects of Smoothness

1) *Analysis of Temperature with Mixup-trained Student:* In previous sections, we observed that both temperature and mixup inject smoothness into KD training process. To investigate the compatibility of smoothness with temperature and mixup, we evaluate KD with time-series data (TS+KD) and Ann. with different temperature parameters. The results of GENEActiv is illustrated in Fig. 9. For TS+KD, when \mathcal{T} is 1, with mixup improves the performance, implying that injecting smoothness can aid for training a student in KD. For both KD with time-series and Ann, in without mixup cases, it shows the best when \mathcal{T} is 4 for WRN16-3 teacher and \mathcal{T} is 12 for WRN16-3 teacher. With mixup, it shows the best when \mathcal{T} is 12 for WRN16-3 teacher and \mathcal{T} is 4 for WRN28-1 teacher, which are different from without mixup. In Fig. 10, for PAMAP2, KD with time-series data without mixup performs the best when \mathcal{T} is 12. However, other results show their best when \mathcal{T} is 4. For both datasets, some accuracy results of KD with time-series and mixup are lower than those without mixup. This represents that excessive smoothness can hinder the training process in KD. For Ann. with mixup outperforms without mixup in all cases. This implies that Ann. has better compatibility for utilizing mixup in KD and can allow more smoothness to improve performance than training with time-series alone.

2) *Partial Mixup:* To control the effects of smoothness on training procedures, we use PMU to alleviate excessive smoothness, which can degrade performance. We utilize different amounts of mixup pairs such as 0%, 10%, 50%, and 100%, where 0% means mixup is not applied and 100% denotes all samples of mixup pairs are used for training (FMU). Mixup is applied when a student is trained. As described in Table VIII, when teacher models are WRN16-3, less amounts of mixup pairs can distill a better student. When teacher models are WRN28-1, 50% of PMU shows the best. In Table IX, for PAMAP2, FMU shows the best.

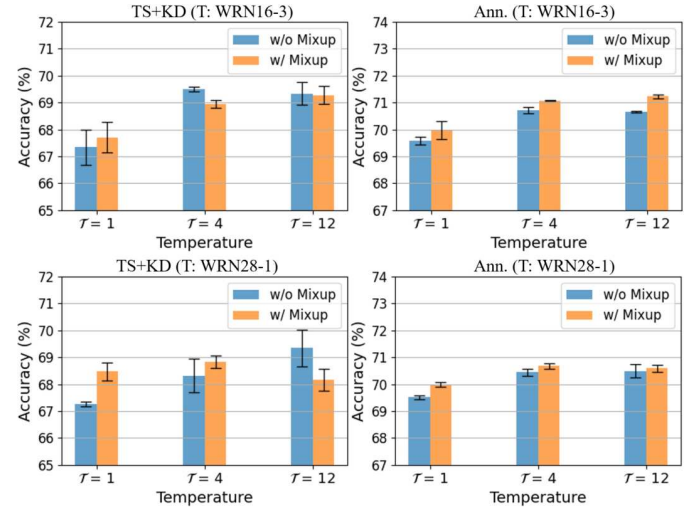


Fig. 9: Results of various models with different temperature and mixup in KD on GENEActiv. Mixup is applied when a student is trained.

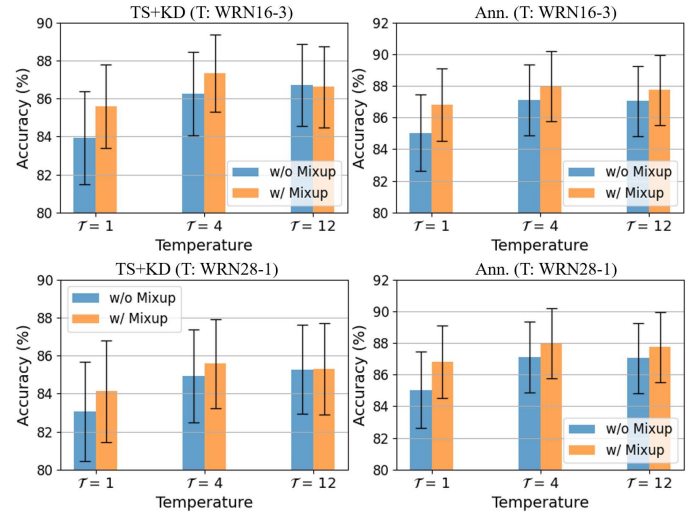


Fig. 10: Results of various models with different temperature and mixup in KD on PAMAP2. Mixup is applied when a student is trained.

However, for WRN28-1, PMU with 10% of Ann. distills the best student. These results show that fewer mixup pairs can generate better performance. Also, if complexity of a dataset is high, mixup pairs contributes more to improving performance. On the other hand, KD with time-series data and Ann. have different optimal proportions of mixup pairs. This may be because Ann. uses both representations, including both time-series with 1D data and topological representations with 2D data, for training. Mixup influences different representations differently, so utilizing two teachers can provide more diverse relaxed knowledge for distillation, which is different from using one single teacher.

F. Mixup for Different Teachers

Since two teachers can provide different effects on distillation, we use different hyper-parameters for mixup to

TABLE VIII: Accuracy (%) with various mixup pair proportions on GENEActiv.

Teachers	Method	No mixup	PMU 0.1	PMU 0.5	FMU
WRN16-3	TS+KD	69.50	69.20	69.11	68.94
		± 0.10	± 0.06	± 0.27	± 0.15
	Ann.	70.71	71.13	70.73	71.07
WRN28-1	TS+KD	68.32	69.17	69.05	68.84
		± 0.63	± 0.36	± 0.15	± 0.23
	Ann.	70.44	70.75	70.82	70.68
		± 0.10	± 0.02	± 0.05	± 0.10

TABLE IX: Accuracy (%) with various mixup pair proportions on PAMAP2.

Teachers	Method	No mixup	PMU 0.1	PMU 0.5	FMU
WRN16-3	TS+KD	86.50	86.75	86.05	87.34
		± 2.21	± 2.10	± 2.27	± 2.03
	Ann.	87.12	87.63	87.54	87.98
WRN28-1	TS+KD	84.92	85.42	85.36	85.58
		± 2.45	± 2.30	± 2.48	± 2.26
	Ann.	85.89	86.69	86.47	86.35
		± 2.26	± 2.20	± 2.29	± 2.39

knowledge transfer from two teachers when a student is trained in KD. We utilize Ann. that shows the best in most of the cases presented in the previous sections. Note, α_1 and α_2 are hyper-parameters of mixup for Teacher1 and Teacher2. As summarized in Table X and XI, applying different mixup hyper-parameters can distill a better student.

TABLE X: Accuracy (%) with various hyper-parameter pairs of mixup for teachers on GENEActiv. Ann. is used for KD.

		Teachers	
α_1	α_2	WRN16-3	WRN28-1
0.1	0.1	70.72 \pm 0.06	70.88 \pm 0.04
0.1	0.15	70.93 \pm 0.11	70.79 \pm 0.12
0.15	0.1	70.99 \pm 0.03	70.88 \pm 0.18
0.15	0.15	70.96 \pm 0.16	71.16 \pm 0.05
0.15	0.2	71.07 \pm 0.14	71.01 \pm 0.16
0.2	0.15	71.22 \pm 0.12	71.00 \pm 0.07
0.2	0.2	71.17 \pm 0.22	70.93 \pm 0.21

TABLE XI: Accuracy (%) with various hyper-parameter pairs of mixup for teachers on PAMAP2. Ann. is used for KD.

		Teachers	
α_1	α_2	WRN16-3	WRN28-1
0.1	0.1	87.98 \pm 2.21	86.35 \pm 2.39
0.1	0.15	87.99 \pm 2.29	86.72 \pm 2.41
0.15	0.1	87.94 \pm 2.26	86.00 \pm 2.43
0.15	0.15	87.67 \pm 2.21	86.70 \pm 2.35

As depicted in Table XII and XIII, we evaluate with different teachers having different architectural designs of depth and width for networks. Mix. denotes applying mixup for training a student. α of mixup is 0.1. When α is applied differently for teachers (diff. α), (α_1, α_2) is (0.15, 0.2) for GENEActiv

and (0.1, 0.15) for PAMAP2. In all cases, applying different mixup hyper-parameters can distill a better student.

TABLE XII: Accuracy (%) with various knowledge distillation methods and different hyper-parameter of mixup for teachers on GENEActiv.

Teacher1 (1D CNNs)	Teacher2 (2D CNNs)	Student (1D CNNs)	TS+PI			
			Base	Ann.	Ann. +Mix.	Ann. +Mix. (diff. α)
WRN16-1 (0.06M, 67.66)	WRN28-1 (0.4M, 59.45)	WRN16-1 (0.06M, 67.66)	68.71 \pm 0.36	69.95 \pm 0.05	70.67 \pm 0.05	70.92 \pm 0.24
WRN28-1 (0.1M, 68.63)	WRN28-3 (3.3M, 59.69)		68.26 \pm 0.13	70.28 \pm 0.08	70.74 \pm 0.15	70.86 \pm 0.13
WRN40-1 (0.2M, 69.05)	WRN28-3 (3.3M, 59.69)		68.90 \pm 0.50	70.49 \pm 0.05	70.91 \pm 0.05	71.21 \pm 0.06

TABLE XIII: Accuracy (%) with various knowledge distillation methods and different hyper-parameter of mixup for teachers on PAMAP2.

Teacher1 (1D CNNs)	Teacher2 (2D CNNs)	Student (1D CNNs)	TS+PI			
			Base	Ann.	Ann. +Mix.	Ann. +Mix. (diff. α)
WRN16-1 (0.06M, 85.27)	WRN28-1 (0.4M, 87.45)	WRN16-1 (0.06M, 82.99)	85.78 \pm 2.29	85.33 \pm 2.22	86.47 \pm 2.35	87.09 \pm 2.16
WRN28-3 (1.1M, 84.46)	WRN28-1 (0.4M, 87.45)		85.69 \pm 2.41	85.59 \pm 2.28	87.06 \pm 2.17	87.80 \pm 2.09
WRN16-3 (0.5M, 85.80)	WRN28-1 (0.4M, 87.45)		85.48 \pm 2.37	85.82 \pm 2.26	86.80 \pm 2.23	87.29 \pm 2.20

To figure out if using different mixup hyper-parameters can complement the partial mixup method, we apply different proportions of mixup pairs for training a student with different mixup hyper-parameters. In Table XIV, FMU shows the best for both cases of teachers. With small proportions of mixup pairs, a large degradation of performance is shown, where the results are lower than training without mixup. When the complexity of the dataset is low and the size of the model is small, partial mixup can yield an adverse effect on training a student, which may produce pairs of inputs that are not expressive enough to learn. In Table XV, 50% of mixup pairs show the best. These results imply that using the proper mixup pair proportion for training a student is important to improve their performance in KD. Also, considering the effects on different relaxed knowledge of a mixup from two teachers can generate a better student.

G. Analysis of Optimized Solution

1) *Parametric Plots:* A solution space comparison for two models can give a valuable understanding of their behavior in training or testing and how these models are related. One of the useful tools for the analysis is the parametric plot that has been widely studied [66]–[68].

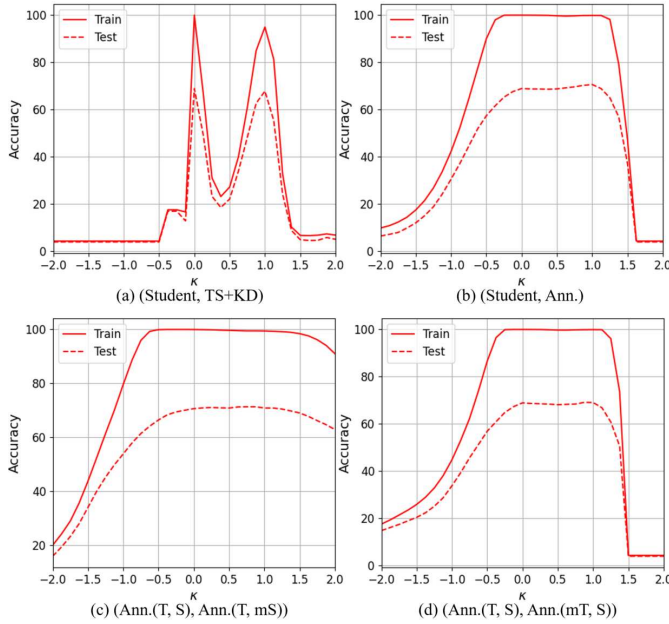
In Fig. 11, we plot classification accuracy for with function $\psi((1 - \kappa)z_a^* + \kappa z_b^*)$ for $\kappa \in [-2, 2]$, where z_a^* and z_b^* are different solutions. Teachers are WRN16-3 and students are

TABLE XIV: Accuracy (%) with various hyper-parameter pairs of mixup on GENEActiv. Ann. is used for KD.

α_1	α_2	Mixup	Teachers	
			WRN16-3	WRN28-1
0.15	0.2	FMU	71.07 \pm 0.14	71.01 \pm 0.16
0.15	0.2	PMU(50%)	70.57 \pm 0.17	70.46 \pm 0.10
0.15	0.2	PMU(10%)	70.55 \pm 0.14	70.73 \pm 0.24
0.2	0.15	FMU	71.22 \pm 0.12	71.00 \pm 0.07
0.2	0.15	PMU(50%)	70.37 \pm 0.05	70.42 \pm 0.03
0.2	0.15	PMU(10%)	70.64 \pm 0.04	70.38 \pm 0.23

TABLE XV: Accuracy (%) with various hyper-parameter pairs of mixup on PAMAP2. Ann. is used for KD.

α_1	α_2	Mixup	Teachers	
			WRN16-3	WRN28-1
0.1	0.15	FMU	87.99 \pm 2.29	86.72 \pm 2.41
0.1	0.15	PMU(50%)	88.13 \pm 2.19	86.73 \pm 2.23
0.1	0.15	PMU(10%)	87.88 \pm 2.29	86.68 \pm 2.26

**Fig. 11:** Parametric plots with accuracy (%) for various pairs of models on GENEActiv. Brackets denote solutions (z_a^* , z_b^*). $\kappa = 0$ implies to z_a^* and $\kappa = 1$ to z_b^* . “Student” is a model learned from scratch.

WRN16-1, which produced the best overall performance in the previous sections. In Fig. 11(a), when κ is 0.5, the accuracy of training and testing is lower than 30%, which represents that the solution spaces of learning from scratch and KD with time-series data are different, whereas the result in Fig. 11(b) shows approximately 70% at $\kappa = 0.5$. This implies that the solution space of Ann. is similar to that of Student. As illustrated in Fig. 11(c), it shows more flattened results. The result at around $\kappa = 1.0$ shows a more gentle slope than the one at $\kappa = 0$, which indicates that using mixup to train a student in KD leads to get benefits for failure prediction and mitigates reliable overfitting. When a mixup trained teacher is used, the student’s solution space is similar to that of a non-mixup trained teacher.

Based on (c) and (d), we can observe that utilizing mixup trained students (T, mS) leads to a better solution space that is relatively less susceptible to perturbations than using mixup trained teachers (mT, S).

2) Mixup Hyper-parameter α : To explore the performance on α of mixup and its sensitivity, we train various models with learning from scratch, KD, and Ann. using different settings of α , which is described in Table XVI. The optimal α parameters for models trained with time-series and topological persistence are different. When α value is between the optimal one of TS and PI ($\alpha \in [0.1, 0.4]$), Ann. performs better than training with the other value ($\alpha = 0.05$). Therefore, setting the proper α leads to getting the best performance, and an intermediate α can generate the best performance when different teachers are applied.

TABLE XVI: Test accuracy (%) under different settings of α on GENEActiv. WRN16-1 is used for learning from scratch and a student.

Method		Mixup α				
		0.05	0.1	0.2	0.4	
Scratch	TS	67.99	68.04	69.28	69.35	
		± 0.41	± 0.63	± 0.19	± 0.52	
	PI	59.23	59.08	59.71	59.47	
		± 0.41	± 0.77	± 0.58	± 0.19	
KD (16-3)	TS	69.02	68.94	69.15	69.39	
		± 0.22	± 0.15	± 0.13	± 0.21	
	PI	67.31	68.08	66.77	68.02	
		± 0.28	± 0.44	± 0.66	± 0.35	
	Ann.	70.63	70.72	71.17	71.35	
		± 0.03	± 0.06	± 0.22	± 0.14	
	KD (28-1)	TS	68.95	68.84	68.74	69.16
			± 0.44	± 0.23	± 0.39	± 0.55
PI		67.77	68.06	67.92	67.83	
		± 0.50	± 0.34	± 0.49	± 0.28	
Ann.		70.81	70.88	70.93	70.76	
		± 0.26	± 0.04	± 0.21	± 0.19	

V. DISCUSSION

We explored the interplay between mixup and KD on diverse strategies with multimodal representations including topological features for wearable sensor data analysis. To achieve more improved synergistic effects, partial mixup can be utilized, which prevents excessive smoothing effects that generate degradation. As an extended research, these strategies introduced in this paper are applicable to diverse computer vision tasks [69], [70], such as image recognition, object tracking and detection, and segmentation. For example, when a model for image recognition is trained with our strategy, the trained model can be utilized as a backbone model in a framework for many different computer vision tasks. Also, this study can be explored on vision based or different types of sensor signal, using motion capture or ECG, based human activity recognition. These can be more investigated as a future work.

VI. CONCLUSION

In this paper, we explored the role of mixup in topological based KD with different approaches. We confirmed that mixup

and temperature in KD have a connecting link that imposes smoothness for training process. Excessive smoothness produced inferior supervision that hinders training a student in KD. We observed that utilizing topological features can complement time-series to improve the end performance. Also, using topological persistence showed better compatibility when using mixup in KD.

Further, two teachers transfer different statistical knowledge so that their optimal parameters for augmentation in distillation can be different, where teachers are trained with time-series and topological features, respectively. We would like to extend a framework using multiple teachers to find optimal hyperparameters of mixup and partial mixup adaptively, considering different statistical characteristics of teachers. In addition, our findings provide insights for developing further advanced distillation methods for various fields including wearable sensor data analysis and computer vision tasks.

ACKNOWLEDGMENT

This work was supported in part by the National Institutes of Health under Grant R01GM135927 as part of the Joint DMS/NIGMS Initiative to Support Research at the Interface of the Biological and Mathematical Sciences, NSF grant 2200161, and in part by Seoul National University of Science and Technology.

REFERENCES

- [1] A. Nawar, 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*, 2020, pp. 793–797.
- [2] A. Som, H. Choi, K. N. Ramamurthy, M. P. Buman, and P. Turaga, "Pinet: A deep learning approach to extract topological persistence images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 834–835.
- [3] E. S. Jeon, 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*, 2022, pp. 837–842.
- [4] H. Adams, 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*, vol. 18, 2017.
- [5] L. M. Seversky, 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*, 2016, pp. 59–67.
- [6] E. Munch, "A user's guide to topological data analysis," *Journal of Learning Analytics*, vol. 4, no. 2, 2017.
- [7] D. Barnes, L. Polanco, and J. A. Perea, "A comparative study of machine learning methods for persistence diagrams," *Frontiers of Artificial Intelligence*, vol. 4, p. 681174, 2021.
- [8] H. Edelsbrunner and J. L. Harer, *Computational topology: an introduction*. American Mathematical Society, 2022.
- [9] E. S. Jeon, H. Choi, A. Shukla, Y. Wang, M. P. Buman, and P. Turaga, "Constrained adaptive distillation based on topological persistence for wearable sensor data," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2023.
- [10] B. Rieck, 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*, vol. 33, pp. 6900–6912, 2020.
- [11] F. Jiang, B. Xu, Z. Zhu, and B. Zhang, "Topological data analysis approach to extract the persistent homology features of ballistocardiogram signal in unobstructive atrial fibrillation detection," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 6920–6930, 2022.
- [12] Y. Yan, Y.-S. Liu, C.-D. Li, J.-H. Wang, L. Ma, J. Xiong, X.-X. Zhao, and L. Wang, "Topological descriptors of gait nonlinear dynamics toward freezing-of-gait episodes recognition in parkinson's disease," *IEEE Sensors Journal*, vol. 22, no. 5, pp. 4294–4304, 2022.
- [13] F. Chazal and B. Michel, "An introduction to topological data analysis: Fundamental and practical aspects for data scientists," *Frontiers in Artificial Intelligence*, vol. 4, 2021.
- [14] H. Wang, S. Lohit, M. N. Jones, and Y. Fu, "What makes a "good" data augmentation in knowledge distillation - a statistical perspective," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 456–13 469, 2022.
- [15] H. Choi, E. S. Jeon, A. Shukla, and P. Turaga, "Understanding the role of mixup in knowledge distillation: An empirical study," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 2319–2328.
- [16] X. Li, H. Xiong, C. Xu, and D. Dou, "Smile: Self-distilled mixup for efficient transfer learning," *arXiv preprint arXiv:2103.13941*, 2021.
- [17] C. Yang, Z. An, H. Zhou, L. Cai, X. Zhi, J. Wu, Y. Xu, and Q. Zhang, "Mixskd: Self-knowledge distillation from mixup for image recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 534–551.
- [18] G. Xu, Z. Liu, and C. C. Loy, "Computation-efficient knowledge distillation via uncertainty-aware mixup," *Pattern Recognition*, vol. 138, p. 109338, 2023.
- [19] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural computation*, vol. 7, no. 1, pp. 108–116, 1995.
- [20] S. Chen, E. Dobriban, and J. H. Lee, "A group-theoretic framework for data augmentation," *Journal of Machine Learning Research*, vol. 21, no. 245, pp. 1–71, 2020.
- [21] R. Shen, S. Bubeck, and S. Gunasekar, "Data augmentation as feature manipulation," in *International conference on machine learning*. PMLR, 2022, pp. 19 773–19 808.
- [22] Z. Allen-Zhu and Y. Li, "Towards understanding ensemble, knowledge distillation and self-distillation in deep learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=Uuf2q9TFXGA>
- [23] S. Yun, 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*, 2019, pp. 6023–6032.
- [24] Z. Allen-Zhu and Y. Li, "Feature purification: How adversarial training performs robust deep learning," in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2022, pp. 977–988.
- [25] T. Zhao, Y. Liu, L. Neves, O. Woodford, M. Jiang, and N. Shah, "Data augmentation for graph neural networks," in *Proceedings of the aaai conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 015–11 023.
- [26] D. Zou, Y. Cao, Y. Li, and Q. Gu, "The benefits of mixup for feature learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 43 423–43 479.
- [27] L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov, "Knowledge distillation: A good teacher is patient and consistent," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 925–10 934.
- [28] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proceedings of the International Conference on Learning Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=r1Ddp1-Rb>
- [29] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Proceedings of the International Conference on Machine Learning*, 2019, pp. 6438–6447.
- [30] J.-H. Kim, W. Choo, H. Jeong, and H. O. Song, "Co-mixup: Saliency guided joint mixup with supermodular diversity," *arXiv preprint arXiv:2102.03065*, 2021.
- [31] L. N. Darlow, A. Joosen, M. Asenov, Q. Deng, J. Wang, and A. Barker, "Tsmix: time series data augmentation by mixing sources," in *Proceedings of the 3rd Workshop on Machine Learning and Systems*, 2023, pp. 109–114.
- [32] K. Aggarwal and J. Srivastava, "Embarrassingly simple mixup for time-series," *arXiv preprint arXiv:2304.04271*, 2023.
- [33] Y. Zhou, L. You, W. Zhu, and P. Xu, "Improving time series forecasting with mixup data augmentation," in *ECML PKDD 2023 International Workshop on Machine Learning for Irregular Time Series*, 2023. [Online]. Available: <https://www.amazon.science/publications/improving-time-series-forecasting-with-mixup-data-augmentation>

- [34] Y. Wang, R. Behroozmand, L. P. Johnson, L. Bonilha, and J. Fridriksson, "Topological signal processing and inference of event-related potential response," *Journal of Neuroscience Methods*, vol. 363, p. 109324, 2021.
- [35] S. Gholizadeh and W. Zadrozny, "A short survey of topological data analysis in time series and systems analysis," *arXiv preprint arXiv:1809.10745*, 2018.
- [36] S. Zeng, F. Graf, C. Hofer, and R. Kwitt, "Topological attention for time series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 871–24 882, 2021.
- [37] B. J. Stolz, "Outlier-robust subsampling techniques for persistent homology," *Journal of Machine Learning Research*, vol. 24, 2023.
- [38] C. Buciluă, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006, pp. 535–541.
- [39] G. Hinton, 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, no. 7, 2015.
- [40] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4794–4802.
- [41] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.
- [42] S. Zagoruyko and N. Komodakis, "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)*, 2017, pp. 1–13.
- [43] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1365–1374.
- [44] Y. Liu, W. Zhang, and J. Wang, "Adaptive multi-teacher multi-level knowledge distillation," *Neurocomputing*, vol. 415, pp. 106–113, 2020.
- [45] H. Zhang, 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)*, 2022, pp. 4498–4502.
- [46] S. You, 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*, 2017, pp. 1285–1294.
- [47] 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 *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2016, pp. 2631–2635.
- [48] 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 of Things Journal*, vol. 9, no. 14, pp. 12 848–12 860, 2022.
- [49] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proceedings of the International Symposium on Wearable Computers*, 2012, pp. 108–109.
- [50] A. Jordao, 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.
- [51] N. Saul and C. Tralie, "Scikit-tda: Topological data analysis for python," 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.2533369>
- [52] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference*, 2016.
- [53] N. Komodakis and S. Zagoruyko, "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)*, 2017, pp. 1–13.
- [54] D. Chen, 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*, 2022, pp. 11 933–11 942.
- [55] T. Huang, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge distillation from a stronger teacher," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 716–33 727, 2022.
- [56] K. Kwon, 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)*, 2020, pp. 7409–7413.
- [57] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [58] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 349–357.
- [59] Y. Chen 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*, 2015, pp. 1488–1492.
- [60] S. Ha, 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*, 2015, pp. 3017–3022.
- [61] S. Ha 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*, 2016, pp. 381–388.
- [62] C. Catal, S. Tufekci, E. Pirmitt, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," *Applied Soft Computing*, vol. 37, pp. 1018–1022, 2015.
- [63] H.-J. Kim, 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*, 2012, pp. 1–5.
- [64] A. Rosenberg 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*, 2007, pp. 410–420.
- [65] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.
- [66] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, "Qualitatively characterizing neural network optimization problems," *arXiv preprint arXiv:1412.6544*, 2014.
- [67] F. Zhu, Z. Cheng, X.-Y. Zhang, and C.-L. Liu, "Rethinking confidence calibration for failure prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 518–536.
- [68] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," in *Proceedings of the International Conference on Learning and Representations (ICLR)*, 2017.
- [69] W. Han, X. Dong, Y. Zhang, D. Crandall, C.-Z. Xu, and J. Shen, "Asymmetric convolution: An efficient and generalized method to fuse feature maps in multiple vision tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [70] X. Dong, J. Shen, F. Porikli, J. Luo, and L. Shao, "Adaptive siamese tracking with a compact latent network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 7, pp. 8049–8062, 2022.



Eun Som Jeon is an assistant professor in the Department of Computer Science and Engineering at Seoul National University of Science and Technology, Seoul, Korea. She received the bachelor's and master's degrees in Electronics and Electrical Engineering from Dongguk University, Seoul, Korea in 2014 and 2016, respectively, and the Ph.D. degree in Computer Engineering (Electrical Engineering) from Arizona State University, Tempe, AZ, USA, in 2023. She worked in Korea Telecom (Institute of Convergence Technology), Seoul, Korea. She worked as a Post-Doctoral Researcher at Arizona State University, Tempe, AZ, USA. Her current research interests include time-series and image data analysis, human behavior analysis, deep learning, and artificial intelligence.



Hongjun Choi is a postdoctoral researcher at Lawrence Livermore National Laboratory. He received his Ph.D. degree in Electrical Engineering at Arizona State University in 2023. He received his Bachelor's and Master's degrees in Electronic Engineering from Korea University in 2014 and 2016, respectively. His research interests are in the fields of machine learning, computer vision with applications in human activity analysis, and image classification using geometric methods.



Matthew P. Buman is a professor in the College of Health Solutions at Arizona State University. His research interests reflect the dynamic interplay of behaviors in the 24-hour 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 is a professor in the School of Arts, Media and Engineering at Arizona State University. He received a bachelor's degree in electronics and communication engineering from the Indian Institute of Technology Guwahati, India, in 2004, and a master's and doctorate in electrical engineering from the University of Maryland, College Park in 2007 and 2009, respectively. 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.