AutoAugHAR: Automated Data Augmentation for Sensor-based Human Activity Recognition

YEXU ZHOU, Karlsruhe Institute of Technology, Germany
HAIBIN ZHAO, Karlsruhe Institute of Technology, Germany
YIRAN HUANG, Karlsruhe Institute of Technology, Germany
TOBIAS RÖDDIGER, Karlsruhe Institute of Technology, Germany
MURAT KURNAZ, Karlsruhe Institute of Technology, Germany
TILL RIEDEL, Karlsruhe Institute of Technology, Germany
MICHAEL BEIGL, Karlsruhe Institute of Technology, Germany

Sensor-based HAR models face challenges in cross-subject generalization due to the complexities of data collection and annotation, impacting the size and representativeness of datasets. While data augmentation has been successfully employed in domains like natural language and image processing, its application in HAR remains underexplored. This study presents AutoAugHAR, an innovative two-stage gradient-based data augmentation optimization framework. AutoAugHAR is designed to take into account the unique attributes of candidate augmentation operations and the unique nature and challenges of HAR tasks. Notably, it optimizes the augmentation pipeline during HAR model training without substantially extending the training duration. In evaluations on eight inertial-measurement-units-based benchmark datasets using five HAR models, AutoAugHAR has demonstrated superior robustness and effectiveness compared to other leading data augmentation frameworks. A salient feature of AutoAugHAR is its model-agnostic design, allowing for its seamless integration with any HAR model without the need for structural modifications. Furthermore, we also demonstrate the generalizability and flexible extensibility of AutoAugHAR on four datasets from other adjacent domains. We strongly recommend its integration as a standard protocol in HAR model training and will release it as an open-source tool[1].

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing**; **Human computer interaction (HCI)**; • **Computing methodologies** → *Supervised learning by classification.*

Additional Key Words and Phrases: machine learning, automated data augmentation, human activity recognition

ACM Reference Format:
Yexu Zhou, Haibin Zhao, Yiran Huang, Tobias Röddiger, Murat Kurnaz, Till Riedel, and Michael Beigl. 2024. AutoAugHAR: Automated Data Augmentation for Sensor-based Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 2, Article 48 (June 2024), 27 pages. https://doi.org/10.1145/3659589

[1]The code will be published at https://github.com/JoeZYX/AutoAugHAR.

Authors' Contact Information: Yexu Zhou, Karlsruhe Institute of Technology, Germany, yexu.zhou@kit.edu; Haibin Zhao, Karlsruhe Institute of Technology, Germany, haibin.zhao@kit.edu; Yiran Huang, Karlsruhe Institute of Technology, Germany, yiran.huang@kit.edu; Tobias Röddiger, Karlsruhe Institute of Technology, Germany, roeddiger@kit.edu; Murat Kurnaz, Karlsruhe Institute of Technology, Germany, Kurnaz@teco.edu; Till Riedel, Karlsruhe Institute of Technology, Germany, till.riedel@kit.edu; Michael Beigl, Karlsruhe Institute of Technology, Germany, michael.beigl@kit.edu.

ACM 2474-9567/2024/6-ART48
https://doi.org/10.1145/3659589

1 INTRODUCTION

Human Activity Recognition (HAR) identifies human activities using various data sources. Among the available methodologies, sensor-based HAR has gained prominence, especially with the evolution of wearable and mobile technologies. This approach offers significant advantages over vision-based HAR, particularly in terms of being non-intrusive and unobtrusive. However, the size of sensor-based HAR datasets is often limited. Challenges in data collection and annotation [1], coupled with the high costs of user studies and volunteer participation [39], limit their representativeness and informativeness. While deep learning models for HAR have shown promising results [51, 77, 78], their ability to generalize across different subjects remains a concern [25].

In order to improve the cross-subject generalizability of deep learning models, Data Augmentation (DA) has emerged as a promising solution. DA involves enriching datasets through the creation of virtual training samples via various transformations on original data. The efficacy of DA has been well-established in fields like computer vision [15, 63] and natural language processing [18]. However, in the context of HAR tasks, the application of DA has received limited attention [2, 28, 70]. Applying DA in the HAR domain presents four primary challenges:

1. Inter-Class Similarity Challenge: HAR datasets often exhibit significant similarities between different activities [12], complicating the application of random transformations without inadvertently altering the activity. For example, Jeong et al. [30] shows how augmenting a "walking" segment might lead to a “jogging" misclassification. Experimental findings [28, 30, 67], highlight how aggressive or inappropriate augmentations might bias the data and diminish model performance. Consequently, a key challenge in DA implementation is the judicious selection of DA methods that respect the characteristics of the data and the semantic nuances of the activities.

2. DA Combination Challenge: Research [32, 67, 70] has demonstrated notable improvements in HAR tasks by training models combining multiple DA methods. However, the sheer number of available methods [28] makes exhaustive experimentation with all possible combinations impractical. Efficiently automating the selection and integration of DA algorithms for HAR tasks remains an unresolved issue, requiring further investigation.

3. Multi-Modality Challenge: When capturing data from various body locations (e.g., chest, arms, ankles) using different sensor types (e.g., accelerometers, gyroscopes, magnetometers), each modality provides unique insights into the target activity. Applying transformations indiscriminately across modalities during DA can inadvertently compromise the effectiveness of these techniques. Furthermore, optimizing DA combinations for each modality results in an exponential increase in the total number of potential combinations as the number of modalities increases.

4. Intra-Class and Inter-Subject Variability Challenge: The general performance of HAR models often declines when applied to new subjects. The variability within classes and between subjects leads to differences in the distributions of training and test data. Although many DA methods aim to generate diverse synthetic data that approximates the distribution of the original data, they may not sufficiently capture the data distributions encountered in test scenarios.

The present research endeavors to explore the potential and effectiveness of DA in enhancing deep learning HAR models. As static DA solutions have not proven to be generalizable, our aim is to design an efficient automated DA framework tailored to the above challenges:

- **Automatic Augmentation Selection and Combination:** This study introduces a differentiable automatic DA approach for HAR tasks, facilitating the learning process to automatically generate effective combinations of augmentations without excess computational demands.
- **Development of AutoAugHAR Framework:** A two-stage automatic DA framework named AutoAugHAR is developed, explicitly designed to maintain label semantic information. Furthermore, considering the multi-modalities nature, AutoAugHAR allows for the optimization of DA combinations specific to each modality.

- **Rigorous Experiments and Ablation Studies:** Comprehensive experiments and ablation studies are conducted across eight benchmark datasets and five state-of-the-art HAR models in order to support our design choices. The results demonstrate the superiority of AutoAugHAR over the compared solutions on the given HAR tasks.
- **Generalizability and Extensibility:** Further experiments are conducted on four datasets from adjacent domains to demonstrate the generalizable performance and flexible extensibility of the proposed AutoAugHAR.
- **Accessibility of AutoAugHAR:** AutoAugHAR is crafted for broad applicability across diverse deep learning models used in HAR tasks. It does not require any modifications to existing models or structural adjustments, thereby enhancing its usability. Furthermore, the framework is extensible and will be made publicly available as an open-source tool, endorsing its integration as a standard procedure to bolster the generalization performance of deep learning models in HAR tasks.

2 RELATED WORK

Existing DA methods for HAR tasks can be broadly classified into two categories: traditional and advanced [70].

2.1 Traditional Approaches

Traditional DA techniques for HAR tasks typically involve random signal transformations such as adding noise, window slicing, magnitude scaling, and random warping [3, 13, 32], among others. However, these methods are not specifically designed for HAR tasks and lack awareness of the target task and data characteristics [70]. Consequently, the implementation of inappropriate DA methods can distort the original data characteristics, possibly altering the label semantic information [30] and negatively impacting model performance [28]. These observations underscore the importance of selecting appropriate DA methods that minimize the distortion of the characteristics of the original data (as mentioned in challenge 1). *w*-augment [19] proposed sample-adaptive automatic weighting schemes to learn the contribution of each random transformation, which enables the exclusion of excessive or redundant methods. However, this approach was not specifically assessed on HAR data. Additionally, while it learns the importance of individual random transformations, it does not explore their combination.

Research has demonstrated that combining multiple augmentation methods can yield improved performance [30, 67, 70]. However, these studies typically involve manually predefining the combination of methods. Identifying the optimal set of augmentation methods is a combinatorial problem, which is NP-hard and requires substantial computational resources (as mentioned in challenge 2).

To mitigate the label semantics issue arising from random transformations, Abedin et al. [1] proposed the MixUp DA method. The MixUp method randomly linearly mixes two data samples and their labels to generate virtual data. Another approach, ALAE-TAE-CutMix [2], deploys a different data mixing technique by randomly replacing sub-segments in one data sample with corresponding sub-segments from another sample. However, these methods face challenges in generating a diverse range of augmented data, especially considering variations in subjects, activities, sensor placements and sensor elasticity [65]. The substantial intra-class and inter-subject variability prevalent in HAR datasets implies that relying solely on training set through mixing techniques may not adequately bridge the distributional gap between training and test sets (as mentioned in challenge 4). As a result, the exploration of supplementary DA techniques capable of increasing the diversity and mirroring real-world scenario variability becomes indispensable.

2.2 Advanced Approaches

Advanced approaches for DA primarily involve synthesizing data using generative models. For instance, Goubeaud et al. [21] trained a variational autoencoder (VAE) on the original data and subsequently generated new samples to augment the training set. Meanwhile, Activitygan [39] employed a generative adversarial network (GAN)-based approach to generate new training samples. Another emerging class of generative models is the diffusion model, increasingly applied in the image domain and recently adapted for HAR. This includes applications in WiFi channel state information-based HAR [27], and wearable-based HAR [62, 81]. Work by Shao et al. [62] reconfigured the U-net architecture for the denoising model to assess the efficacy of diffusion-based models. Zuo et al. [81] conditioned the diffusion model on statistical information to generate diverse synthetic sensor data. However, a noteworthy limitation of these approaches is that they did not train the virtual data generator in an end-to-end manner alongside the HAR model, potentially resulting in sub-optimal performance (as mentioned in challenge 4).

Addressing this issue, the sample fusion network (SFN) [45] cascaded a long short term memory (LSTM) autoencoder (AE) network to the HAR network, employing a data mixing style to create a combined network that can be trained in an end-to-end manner. Augmented Adversarial Learning framework for HAR (AALH) [33] utilized an adversarial neural network to learn common latent representation from various sensor modalities. Some efforts even attempted to generate virtual data through other forms of data. For example, Vi2IMU [60] attempted to generate virtual Smartwatch IMU Data through Videos. While these advanced approaches offer data- and task-dependent advantages, it is important to acknowledge that they tend to be computationally expensive [28, 70].

Successfully applying the above advanced techniques often demands a high level of expertise in model structure design and training configuration to guarantee the quality of synthetic samples. To mitigate the need for such expert knowledge, fields like image processing have extensively explored automated DA optimization frameworks [14, 24]. However, these frameworks have not been explicitly applied to the HAR field. Furthermore, unlike image data, HAR data involves more complex modalities and is more sensitive to perturbations, factors not accounted for in previous research [14, 24].

3 PRELIMINARIES

In this section, the foundational concepts essential for the implementation of a self-optimizing generalizable DA are presented. Particularly, we introduce the concept of DA sub-policies and define an optimization problem.

3.1 Background

Let O represent a set comprising M candidate time series processing operations. Each operation o within O represents a function capable of transforming the time series samples, given by $\tilde{\boldsymbol{x}} = o(\boldsymbol{x})$. An augmentation sub-policy, denoted as s and composed of N consecutive transformations, is expressed as:

$$\tilde{\boldsymbol{x}} = s(\boldsymbol{x}) = o_N(\cdots o_2(o_1(\boldsymbol{x}))) \tag{1}$$

In this formulation, each operation is sequentially applied to the time series sample $\boldsymbol{x}$. To implement an augmentation sub-policy, it is necessary to determine the number of consecutive operations N and identify the appropriate operation form the set O for each transformation step. For instance, Chung et al. [13] utilized a single transformation method jittering, which means the number of consecutive operations $N = 1$ and the $op = "jittering"$. And Um et al. [67] utilized an augmentation sub-policy encompassing three consecutive operations ($N = 3$), which are $o_1 = rotation$, $o_2 = permutation$, and $o_3 = time - warping$, respectively.

Let $\mathcal{S}$ symbolize the set of all possible augmentation sub-policies. Given M candidate operations and each augmentation sub-policy comprising N consecutive transformations, the total number of augmentation sub-policies

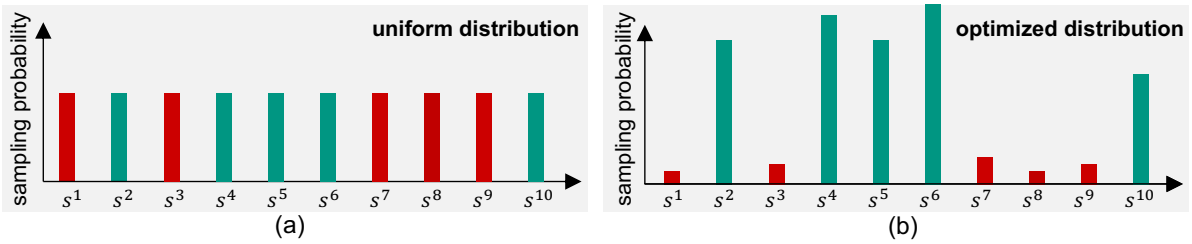


Fig. 1. The categorical distribution for 10 augmentation sub-policies. Those depicted in red represent augmentation sub-policies that exert a negative influence, while those in green denote sub-policies with a positive impact.

is $L = |\mathcal{S}| = P_M^N$, where $\mathcal{S} = \left(s^1, s^2, \cdots, s^L\right)$. Each augmentation sub-policy possesses a distinct combination of the candidate operations. However, a brute-force approach to explore all various augmentation sub-policies demands significant experimentation and computational resources, rendering it inefficient and sub-optimal.

3.2 Naive Approach

To address the challenge of efficiently selecting effective augmentation sub-policies, we use a baseline approach, that is commonly used DA mechanism in computer vision model [15]. Instead of using only a singular, predefined augmentation sub-policy, the entire set of possible sub-policies $\mathcal{S}$ is employed in a stochastic manner during the training phase. During the augmentation phase of each mini-batch iteration, a single augmentation sub-policy is randomly selected from $\mathcal{S}$ and subsequently applied to the mini-batch data.

Let the categorical distribution $\boldsymbol{p}$ represent the likelihood of each sub-policy being sampled. In this context, $\boldsymbol{p} = \left[p^1, p^2, \cdots, p^L\right] = \left[\frac{1}{L}, \frac{1}{L}, \cdots, \frac{1}{L}\right]$ follows a uniform probability distribution, providing an equal chance for all augmentation sub-policies to be utilized. Thus, the entirety of possible augmentation sub-policies is leveraged, offering a significantly increased diversity of the augmented data. Furthermore, this approach mitigates potential model degradation that could arise from the utilization of predetermined, sub-optimal augmentation sub-policies.

3.3 Data Augmentation Optimization

The naive approach adopted a uniform probability distribution, treating all augmentation sub-policies equally without considering their potential positive or negative impacts, as demonstrated in Figure 1 (a). Ideally, augmentation sub-policies that exert positive influences should be assigned higher probabilities. Therefore, the primary objective of this study is to automatically optimize the categorical distribution for all augmentation sub-policies, as illustrated in Figure 1 (b). Moreover, this optimization process should be executed without incurring excessive training overheads or necessitating model modifications.

4 METHODOLOGY

To effectively optimize the categorical distribution (combinatory space) of sub-policies, we present a two-stage gradient-based framework, termed AutoAugHAR. This framework takes into account multi-modality characteristics inherent in HAR tasks. Optimization of the categorical distribution and weights of the HAR model is performed end-to-end (see Section 4.1) using gradient descent (Section 4.2) . We have designed a HAR specific search space of DA operators that constitute the augmentation sub-policies (Section 4.3).

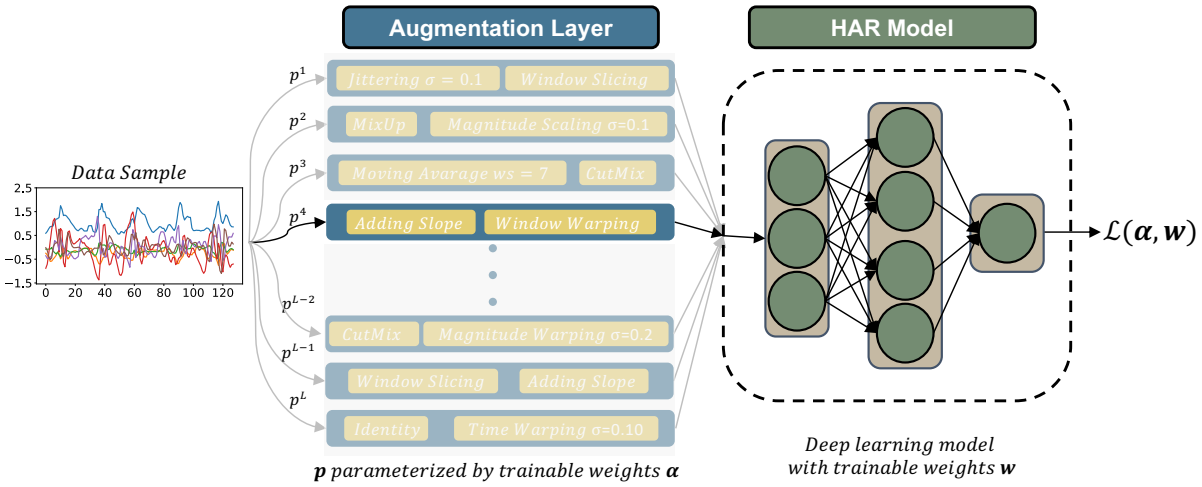


Fig. 2. An overview of the proposed AutoAugHARbasic. Every channel across all modalities undergoes the same augmentation sub-policy. During the data propagation phase, only one augmentation sub-policy is chosen and applied. The selection of this sub-policy depends on the probability associated with each path (sub-policy). The objective of the optimization is to ensure that sub-policies which improve performance have a higher probability compared to those that affect performance.

4.1 Overview of AutoAugHAR

During the training phase, data samples are subjected to DA transformations before being input into the HAR models (see augmentation layer preceding the HAR model in Figure 2). Contrary to prior studies, this framework incorporates a total of L distinct augmentation sub-policies. For each mini-batch iteration, only a sub-policy is selected and executed, ensuring that memory consumption aligns with traditional HAR model training paradigms. The selection is determined by sampling the sub-policy in accordance with the categorical distribution $\boldsymbol{p} = \left[p^1, p^2, \cdots, p^L\right]$. Each element p^i represents the probability of selecting the i-th sub-policy in the sampling process:

$$\boldsymbol{c} = \text{sample}\left(p^1, p^2, \cdots, p^L\right) = \begin{cases} [1, 0, \cdots, 0] & \textit{with probability } p^1 \\ \cdots & \\ [0, 0, \cdots, 1] & \textit{with probability } p^L \end{cases} \tag{2}$$

Upon sampling, the path within the augmentation layer is represented as a one-hot encoded vector, denoted as $\boldsymbol{c}$, contingent on the associated probability:

$$\tilde{\boldsymbol{x}} = \sum_{i=1}^{L} c^i s^i\left(\boldsymbol{x}\right) = \begin{cases} s^1\left(\boldsymbol{x}\right) & \textit{with probability } p^1 \\ \cdots & \\ s^L\left(\boldsymbol{x}\right) & \textit{with probability } p^L \end{cases} \tag{3}$$

At any given instance, only a single augmentation sub-policy path is activated. If the categorical distribution of sub-policies $\boldsymbol{p}$ is uniformly distributed and remains unoptimized, this procedure mirrors the baseline approach detailed in section 3.2. For clarity in subsequent experimental comparisons, we refer to this framework as **AutoAugHARrandom**.

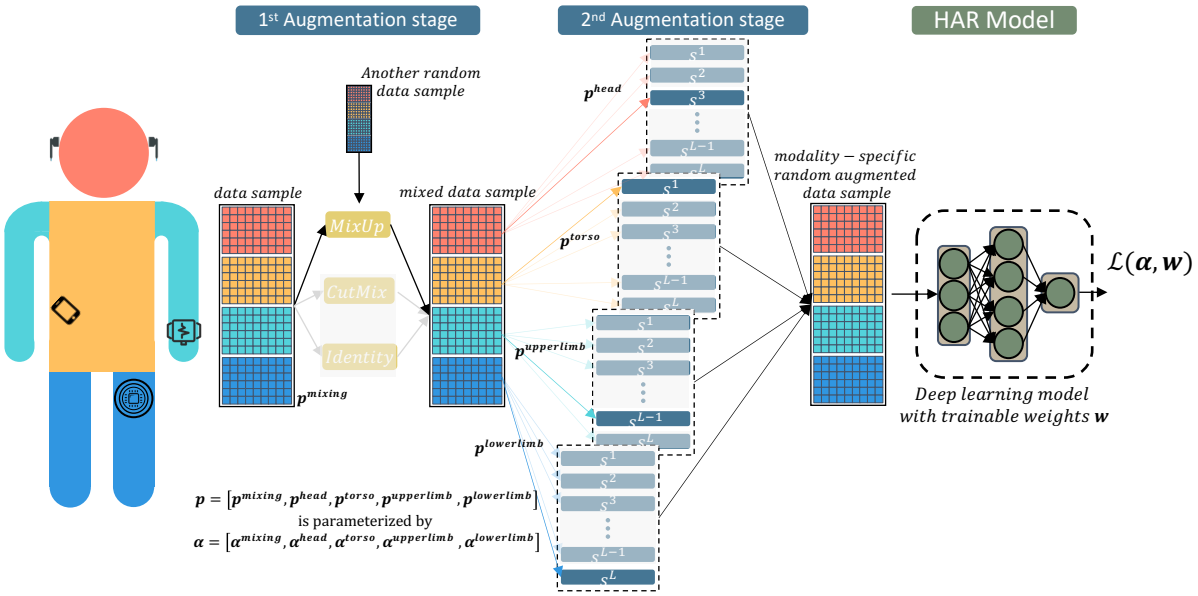


Fig. 3. An overview of the proposed AutoAugHAR. Different colors represent data from different modalities. In the first stage, data from all modalities undergo transformation by one selected operator. In the second stage, data from each modality is individually transformed by different operators and then integrated together.

Categorical distribution $\boldsymbol{p}$ is parameterized by a learnable vector $\boldsymbol{\alpha} = \left[\alpha^1, \alpha^2, \cdots, \alpha^L\right]$. Subsequently, the distribution $\boldsymbol{p}$ is derived by applying the softmax function over $\boldsymbol{\alpha}$ leading to the following probability of selecting the i-th sub-policy:

$$p^i = p_{\boldsymbol{\alpha}}\left(s = s^i\right) = \text{softmax}\left(\alpha^i; \boldsymbol{\alpha}\right) = \frac{\exp\left(\alpha^i\right)}{\sum_{j=1}^{L} \exp\left(\alpha^j\right)} \tag{4}$$

α_i signifies the importance attributed to the i-th sub-policy: a relatively large value of α_i indicates a higher likelihood for selecting the corresponding i-th sub-policy. We define the resulting optimization problem as follows:

$$\min_{(\boldsymbol{\alpha}, \boldsymbol{w})} \mathcal{L}\left(\boldsymbol{\alpha}, \boldsymbol{w}\right) \tag{5}$$

Both $\boldsymbol{p}$ and $\boldsymbol{w}$ are subjected to end-to-end training, minimizing the loss function. The "sample" operation in equation 2 introduces a discontinuity, thereby inhibiting the propagation of gradients to the weights $\boldsymbol{\alpha}$ (see section 4.2). While this optimization strategy enables dynamic weights optimization for augmentation sub-policies during the model's training phase, it still overlooks the distinct attributes of the candidate DA operations within set O and the inherent characteristics of HAR tasks. (We refer to this framework as **AutoAugHARbasic**.)

Given two distinct categories of candidate DA operations, each with varying capacities to preserve label semantics (refer to Section 4.3), and aiming to optimize augmentation sub-polices for each modality, we revised the structure. The updated framework, named **AutoAugHAR**, is depicted in Figure 3. **AutoAugHAR** operates in two stages. The first mixing stage transforms the data using label-preserving sample-pair-based mixing techniques

like MixUp and CutMix. The second stage, which has a slightly higher risk of compromising label semantics, incorporates random DA augmentations to further enrich the diversity of the data.

In contrast to **AutoAugHARbasic**, which optimizes $\boldsymbol{p}$ universally across all modalities, **AutoAugHAR** tailors the distribution of augmentation sub-policies for each modality in the second stage individually. Modalities are classified based on sensor placement, e.g. head, upper limb, lower limb, and torso. During the forward pass of the second stage, data samples are partitioned according to the modality. For each, a path is sampled independently, with the corresponding augmentation sub-policy applied. Subsequently, data from all modalities are integrated and fed to the HAR model. We did not perform individual modality-wise optimization in the first stage. This is attributed to the incompatibility of sample-pair-based mixing techniques for such a purpose (see section 4.3).

For optimization, separate categorical distributions are initialized for the first mixing stage and each modality of the second stage. These are denoted as $\left(\boldsymbol{p}^{mixing}, \boldsymbol{p}^{head}, \boldsymbol{p}^{upperlimb}, \boldsymbol{p}^{lowerlimb}, \boldsymbol{p}^{torso}\right)$ with learnable vectors $\left(\boldsymbol{\alpha}^{mixing}, \boldsymbol{\alpha}^{head}, \boldsymbol{\alpha}^{upperlimb}, \boldsymbol{\alpha}^{lowerlimb}, \boldsymbol{\alpha}^{torso}\right)$. The size and search space of these distributions are detailed in section 4.3.1 and 4.3.2. During the forward pass, within each categorical distribution, a path will be sampled using equation 2. The corresponding sub-policy is then applied to the data according to equation 3. Upon optimization, the first mixing stage as well as all individual modalities of the second stage, will acquire a specifically optimized distribution for applying DA sub-policies. By leveraging this two-stage structure, **AutoAugHAR** seeks to preserve the label semantic information of the data while generating optimized categorical distribution of augmentation policies intrinsic to each modality, thus improving the overall performance.

4.2 Gradient Based Optimization

The loss function in equation 5 is differentiable with respect to the model weights $\boldsymbol{w}$, allowing optimization via stochastic gradient descent. However, the loss is not directly differentiable with respect to the sampling parameter $\boldsymbol{\alpha}$, because the discrete "sample" operation introduces non-differentiable points in the network. This section will describe how to optimize the weights $\boldsymbol{\alpha}$ using a gradient descent approach. Because the forward process and gradient back propagation process of all categorical distributions parameterized by $\boldsymbol{\alpha}$ are same, we decide to omit the use of superscripts (head, upperlimb, lowerlimb and torso) to explain these processes.

In order to facilitate back-propagation through these non-differentiable operations, Straight-Through Estimators (STE) [8] are employed. The basic idea behind STE is to provide a way to back-propagate gradients through these non-differentiable operations while maintaining their original behavior during forward pass and avoiding any gradient vanishing or exploding issues during the backward pass. In order to obtain a differentiable approximation, we apply the Straight-Through Gumbel-Softmax Estimator [29]. Compared to sampling the path with equation 2, the Gumbel-Max trick [22, 44] provides a different way to sample the path (sub-policy):

$$\boldsymbol{c} = \text{Onehot\_Encoding}\left(\underset{i}{\text{argmax}}\left(g^i + \log\left(\alpha^i\right)\right)\right) \tag{6}$$

where g^i are independent samples drawn from a standard Gumbel distribution $g^i \sim \text{Gumbel}\,(0, 1)$. The reparameterization trick refractors the sampling of $\boldsymbol{c}$ into a deterministic distribution function using $\boldsymbol{\alpha}$ and independent noise g from a fixed distribution, maintaining an identical sampling procedure using equation 2. This technique avoids having to back-propagate through the stochastic node g and instead only back-propagate into the deterministic distribution function, updating the parameters $\boldsymbol{\alpha}$.

During the gradient back-propagation, $argmax$ operation is still not differentiable. To address this issue, a differentiable approximation of $argmax$ is needed. Gumbel $softmax$ [29] offers a differentiable approximation to $argmax$, as utilized in various works [17, 40, 72, 73]:

$$p^i = \text{GumbelSoftmax}\left(\alpha^i; \boldsymbol{\alpha}\right) = \frac{\exp\left(\left(\log\left(\alpha^i\right) + g^i\right)/\tau\right)}{\sum_{j=1}^{L}\exp\left(\left(\log\left(\alpha^j\right) + g^j\right)/\tau\right)} \tag{7}$$

τ is the temperature parameter that controls the fidelity of the approximation to discrete one-hot vectors. Consequently, this allows the model to be trained with discrete operations, using equations 6 and 3 for the forward pass and the differentiable equation 7 for gradient back-propagation.

Following the exploration of the differentiable optimization problem, we now present the entire optimization process. To evaluate whether a categorical distribution for augmentation sub-policies is good or not good, it is needed to train the HAR model to converge to obtain the optimal model weights, $\boldsymbol{w}^*\left(\boldsymbol{\alpha}\right) = argmin_{\boldsymbol{w}}\ \mathcal{L}_{train}\left(\boldsymbol{w}, \boldsymbol{\alpha}\right)$. The optimal weight $\boldsymbol{w}^*$ is affected by the categorical distribution parameterized by $\boldsymbol{\alpha}$, if $\boldsymbol{\alpha}$ changes, the corresponding optimal $\boldsymbol{w}^*$ will also change. This implies a typical bi-level optimization problem [4, 42, 75] with $\boldsymbol{\alpha}$ as the upper-level variable and the model weight $\boldsymbol{w}$ as the lower-level variable, mathematically defined as follows.

$$\min_{\alpha}\ \mathcal{L}_{val}\left(\boldsymbol{w}^*\left(\boldsymbol{\alpha}\right), \boldsymbol{\alpha}\right) \quad s.t.\ \ \boldsymbol{w}^*\left(\boldsymbol{\alpha}\right) = \underset{\boldsymbol{w}}{argmin}\ \ \mathcal{L}_{train}\left(\boldsymbol{w}, \boldsymbol{\alpha}\right) \tag{8}$$

where $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ denote the training and validation loss, respectively. The objective is to determine the categorical distribution parameterized by $\boldsymbol{\alpha}$, which minimizes the validation loss, where the weights of the HAR model $\boldsymbol{w}$ are obtained by minimizing the training loss. Using the performance of the validation set as a reward for updating upper-level variable is a common practice [41, 42, 79, 80]. In this case, minimizing the validation loss through $\boldsymbol{\alpha}$ encourages the optimized categorical distribution for augmentation sub-policies preserve the data semantic information and also fill the gap between seen (training) and unseen (validation) data. This mitigates the risk of generating augmentation sub-policies that might over-fit the training set.

Algorithm 1 Training Procedure

Variables:
$\boldsymbol{\alpha}$ - Categorical distribution for augmentation sub-policies $\boldsymbol{\alpha} = \left(\boldsymbol{\alpha}^{mixing}, \boldsymbol{\alpha}^{head}, \boldsymbol{\alpha}^{upperlimb}, \boldsymbol{\alpha}^{lowerlimb}, \boldsymbol{\alpha}^{torso}\right)$
$\boldsymbol{w}$ - weights of the model
ξ_w - Learning rate for updating $\boldsymbol{w}$
ξ_α - Learning rate for updating $\boldsymbol{\alpha}$
$epoch_{search}$ - Number of epoch for optimization the augmentation sub-policies

1: **for** i=1 to $epoch_{search}$ **do**
2: Augmentation sub-policy sampling
3: **for** Sample a mini-batch of data **do**
4: Update categorical distribution parameter $\boldsymbol{\alpha}$: $\boldsymbol{\alpha} = \boldsymbol{\alpha} - \xi_\alpha \nabla_{\boldsymbol{\alpha}} \mathcal{L}_{val}\left(\boldsymbol{w}^*, \boldsymbol{\alpha}\right)$
5: Update model weights $\boldsymbol{w}$: $\boldsymbol{w} = \boldsymbol{w} - \xi_w \nabla_{\boldsymbol{w}} \mathcal{L}_{train}\left(\boldsymbol{w}, \boldsymbol{\alpha}\right)$
6: Augmentation sub-policy sampling
7: **end for**
8: **end for**

During the training process, $\boldsymbol{w}$ and $\boldsymbol{\alpha}$ are alternately fine-tuned through gradient descent. The training protocol is explained in Algorithm 1. Initially, an augmentation sub-policy is sampled for the initial mini-batch data loading in accordance with equation 6 (line 2). Following this step, $\boldsymbol{\alpha}$ undergoes an update through gradient calculations (line 4). Subsequently, the weights $\boldsymbol{w}$ of the model are updated on the basis of the updated $\boldsymbol{\alpha}$ (line 5).

Conclusively, the augmentation sub-policy is re-sampled for the forthcoming mini-batch data loading (line 6). This alternating optimization procedure is repeated until the maximum optimization epoch is reached.

However, during the $\boldsymbol{\alpha}$ update step, the calculation of the gradient of $\boldsymbol{\alpha}$ requires a computationally intensive internal optimization. $\boldsymbol{w}^*$ are derived by minimizing training loss. To avoid extensive optimization, a one-step optimization technique is employed to approximate $\boldsymbol{w}^*$, as outlined below.

$$\nabla_{\boldsymbol{\alpha}} \mathcal{L}_{val}\left(\boldsymbol{w}^*\left(\boldsymbol{\alpha}\right), \boldsymbol{\alpha}\right) \tag{9}$$

$$\approx \nabla_{\boldsymbol{\alpha}} \mathcal{L}_{val}\left(\boldsymbol{w} - \xi_w \nabla_{\boldsymbol{\alpha}} \mathcal{L}_{train}\left(\boldsymbol{w}, \boldsymbol{\alpha}\right), \boldsymbol{\alpha}\right) \tag{10}$$

$$= \nabla_{\boldsymbol{\alpha}} \mathcal{L}_{val}\left(\boldsymbol{w}^{'}, \boldsymbol{\alpha}\right) - \xi_w \nabla^2_{\boldsymbol{\alpha},\boldsymbol{w}} \mathcal{L}_{train}\left(\boldsymbol{w}, \boldsymbol{\alpha}\right) \nabla_{\boldsymbol{w}^{'}} \mathcal{L}_{val}\left(\boldsymbol{w}^{'}, \boldsymbol{\alpha}\right) \tag{11}$$

here, $\boldsymbol{w}* \approx \boldsymbol{w}^{'} = \boldsymbol{w} - \xi_w \nabla_{\boldsymbol{\alpha}} \mathcal{L}_{train}\left(\boldsymbol{w}, \boldsymbol{\alpha}\right)$ is approximated using a single virtual gradient step over the training set. By applying the chain rule for derivatives, equation 11 is derived. However, the second term in the equation 11 contains an expensive matrix-vector product with a computational complexity of $O\left(|\boldsymbol{\alpha}|\,|\boldsymbol{w}|\right)$. Fortunately, the complexity can be significantly reduced using a finite difference approximation. Let ϵ be a small scalar, and $\boldsymbol{w}^{\pm} = \boldsymbol{w} \pm \epsilon \nabla_{\boldsymbol{w}^{'}} \mathcal{L}_{val}\left(\boldsymbol{w}^{'}, \boldsymbol{\alpha}\right)$, then:

$$\nabla^2_{\boldsymbol{\alpha},\boldsymbol{w}} \mathcal{L}_{train}\left(\boldsymbol{w}, \boldsymbol{\alpha}\right) \nabla_{\boldsymbol{w}^{'}} \mathcal{L}_{val}\left(\boldsymbol{w}^{'}, \boldsymbol{\alpha}\right) \approx \frac{\nabla_{\boldsymbol{\alpha}} \mathcal{L}_{train}\left(\boldsymbol{w}^{+}, \boldsymbol{\alpha}\right) - \nabla_{\boldsymbol{\alpha}} \mathcal{L}_{train}\left(\boldsymbol{w}^{-}, \boldsymbol{\alpha}\right)}{2\epsilon} \tag{12}$$

The evaluation of the finite difference requires only two forward passes for the weights and two backward passes for $\boldsymbol{alpha}$. following the settings in [40, 42] , we let $\epsilon = 0.01/\left\|\nabla_{\boldsymbol{w}^{'}} \mathcal{L}_{val}\left(\boldsymbol{w}^{'}, \boldsymbol{\alpha}\right)\right\|_2$.

4.3 Candidate Operations And Search Space

Drawing from a thorough review of the relevant literature, we have identified a set of 17 operators, which are both diverse and computationally efficient. All considered candidate DA operators are graphically depicted in Figure 4 and their hyper-parameter settings are shown in the Table 1. The settings of these hyper-parameters are summarized from related works [1, 2, 19, 28]. It is important to note that the same operators but with different hyper-parameters are regarded as distinct and unique entities. The candidate operators can be systematically categorized into two primary categories: label-preserving augmentation operators and random transformation operators. In the following sections, we will go into the details of each of these two categories of operators.

Table 1. Candidate operators and the settings of hyper-parameters.

Method	*Parameter*	*value/range*	*Method*	*Parameter*	*value/range*
Jittering	σ	0.05	Jittering	σ	0.10
Jittering	σ	0.15	Identity	\	
Moving Average	ws	3	Moving Average	ws	5
Moving Average	ws	7			
Magnitude Scaling	σ	0.1	Magnitude Scaling	σ	0.2
Magnitude Warping	σ	0.2	Magnitude Warping	σ	0.4
Window Slicing	λ	$[0.7, 0.9]$	Slope-Like Trend	$slope$	$[-0.1, 0.1]$
Time Warping	σ	0.1	Time Warping	σ	0.2
Mixup	α	0.3	CutMix	α	0.8

4.3.1 Label-preserving Operators. Label-preserving transformation operators aim to produce virtual data that retain the intrinsic characteristics of the original data. A widely applied technique in this regard is the use of sample-pair-based methods, which involves mixing signals and labels from two input data samples.

In the **MixUp** approach, for two given samples $(\boldsymbol{x}_1, y_1)$ and $(\boldsymbol{x}_2, y_2)$, a virtual training sample $(\tilde{\boldsymbol{x}}, \tilde{y})$ is created through linear interpolation between the input sample pair as follows: $\tilde{\boldsymbol{x}} = \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$ and $\tilde{y} = \lambda y_1 + (1-\lambda) y_2$. The mixing ratio λ is a stochastic value drawn from the beta distribution $\mathcal{B}(\alpha, \alpha)$, determining the mixing intensity. Adhering to the configuration presented in [1], the parameter α is specified as 0.3.

In the **CutMix** methodology, segments from two samples are swapped to produce a new virtual sample. Given two samples, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, each comprising T time steps. A randomly selected region, spanning a length of λT, is delineated within $\boldsymbol{x}_1$, from which the respective sub-segment is cropped. This cropped sub-segment subsequently replaces the counterpart in sample $\boldsymbol{x}_2$, leading to the formation of a mixed virtual sample. Unlike the MixUp mechanism, CutMix generates virtual samples without altering the raw data values. The samples generated by CutMix have steep transitions between activities. The label associated with this augmented data is derived from the formula, $\tilde{y} = \lambda y_1 + (1-\lambda) y_2$. Here, the coefficient λ, which controls the degree of mixing, is also drawn from the beta distribution $\mathcal{B}(\alpha, \alpha)$. As referenced in [2], the parameter α is set at 0.8.

In addition to those two operators, the identity operator is also incorporated. Within the first stage, the count of conservative operations N_{1st} is set to 1. Thus, the size of the categorical distribution for this stage is $\boldsymbol{p}^{mixing} \in \mathbb{R}^3$. Modality-wise optimization is not adopted, because the mixing approach does not merely amalgamate data values, but also integrates their corresponding labels, which would become semantically ambiguous.

4.3.2 Random Transformation Operators. Given that random transformations run the risk of corrupting the original semantic information of the data, in total 14 such candidate operators are employed in the second stage of **AutoAugHAR**.

Adding Noise to data samples, also referred to as jittering, is specifically designed to simulate sensor noise. To implement this method, controlled amounts of random noise are added to the raw data, producing a new representation denoted as $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon}$ symbolizing the random drawn noise vector from a Gaussian distribution $\mathcal{N}\left(0, \sigma^2\right)$. Unlike prior studies, we introduced multiple distinct values of σ, namely **0.05**, **0.1**, and **0.15**, with each corresponding to different noise intensities.

Moving Average involves calculating the average of a sliding window of sensor data over time. It is effective in mitigating the impact of outliers and noise present in the sensor data, but might introduce lag, especially with larger window sizes (ws). The moving average's inherent smoothing effect can dampen abrupt changes. Therefore, the selection of an appropriate ws becomes critical. We have incorporated three ws: **3**, **5**, and **7**.

Magnitude Scaling is a technique that alters the magnitude of a signal by applying a stochastic scaling factor to simulate variations in the intensity of physical activity observed in real-world scenarios, defined as $\tilde{\boldsymbol{x}} = \boldsymbol{x} \times sf$, where the scaling factor sf is a stochastic variable drawn from a Gaussian distribution $\mathcal{N}\left(1, \sigma^2\right)$.We have introduced two scaling ranges, $\sigma = 0.1$ and $\sigma = 0.2$.

Magnitude Warping [6]. In contrast to magnitude scaling, which uniformly scales all values within the signal using the same factor, magnitude warping involves distorting the magnitude of the signal by applying a smoothed curve generated through cubic spline interpolation with K knots. As a result, Magnitude Warping can produce more realistic variations in the intensity of time-series data. To implement magnitude warping, K knots (reference points) are selected along the time-series data, dividing the time-series data into $K-1$ equal segments. For each knot, a random scaling factor is sampled from a Gaussian distribution $\mathcal{N}\left(1, \sigma^2\right)$. These knots, along with their corresponding scaling factors, serve as control points for the subsequent cubic spline interpolation. The original time-series data is then warped by applying the values of the interpolated curve s_t at corresponding time points t, $\tilde{x}_t = x_t \times s_t$. We defined two scaling ranges, $\sigma = 0.2$ and $\sigma = 0.4$ to avoid unrealistic distortions.

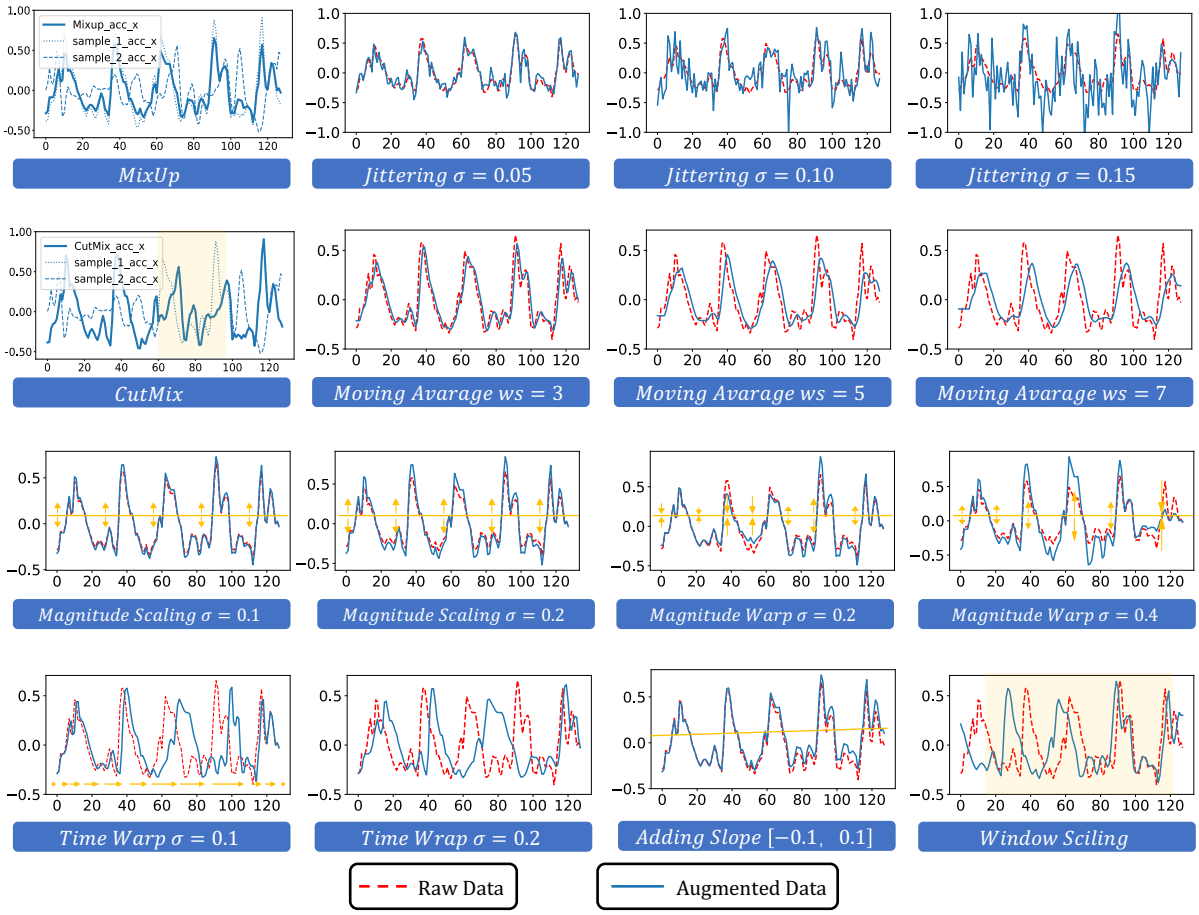
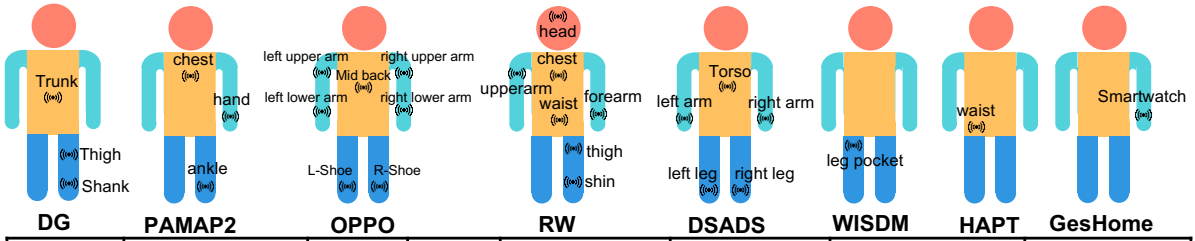


Fig. 4. Examples of candidate augmentation operators on the HAPT dataset.

Window Slicing [37], also known as cropping, involves randomly selecting a segments of random length from the original signal. Given a data sample $\boldsymbol{x}$ with T time steps, mathematically, window slicing can be expressed as $\tilde{\boldsymbol{x}} = \boldsymbol{x}[t : t + \lambda T]$ at the starting point t with a random segment length λT. We chose to sample segments between 70% and 90% of the original length of the input signal, $\lambda \in [0.7, 0.9]$.

Time Warping [28] involves warping the time steps based on a cubic spline interpolation with K knots, that are first selected along the time-series data, dividing the time-series data into $K - 1$ equal segments. These knots are randomly perturbed by multiplying them by a random factor which is sampled from a Gaussian distribution $\mathcal{N}(1, \sigma^2)$, where sigma controls the strength of perturbations (we use $\sigma = 0.1$ and $\sigma = 0.2$). The original time steps are replaced with warped time steps from the new spline. Values corresponding to new time steps are obtained through interpolation.

Incorporating a Slope-Like Trend involves adding a linear trend to the time series to represent patterns in specific scenarios, such as a drift in accellerometer data. The slope is selected randomly from a predetermined range, $[-0.1, 0.1]$.



Datasets	Sampling Frequency	# Subjects	# Classes	# Sensor Channels	Sensor Types	Sliding Window Size	# Modalities
DG	64 Hz	10	2	9	Acc	1 s	2
PAMAP2	100 Hz	9	18	18	Acc, Gyro	2 s	3
OPPO	30 Hz	4	18	42	Acc, Gyro	1 s	3
RW	50 Hz	15	8	21	Acc	2.56 s	4
DSADS	25 Hz	8	19	30	Acc, Gyro	5 s	3
WISDM	20 Hz	36	6	3	Acc	5.12 s	1
HAPT	50 Hz	30	12	6	Acc, Gyro	2.56 s	1
GesHome	25 Hz	20	18	9	Acc, Gyro, Mag	2 s	1

Fig. 5. Statistical summary of selected datasets.

If operators change the length of the original time series, we interpolate the transformed time series back to the original length. In addition, the identity operator is also incorporated in this step.

In this second stage of **AutoAugHAR**, we set the conservative operation count N_{2nd} to 2. Therefore, there are in total $P^2_{14+1} - 14 - 3 \times 2 - 3 \times 2 - 2 - 2 - 2 = 178$ augmentation sub-policies. Among all possible augmentation sub-policies, we eliminate identical ones. For example, 'identity + jittering' is the same as 'jittering + identity'. We also removed sub-policies with the same operators, such as 'jittering $\sigma = 0.05$ + jittering $\sigma = 0.10$'. As a result, each modality's categorical distribution size is as follows: $\boldsymbol{p}^{head} \in \mathbb{R}^{178}$, $\boldsymbol{p}^{upperlimb} \in \mathbb{R}^{178}$, $\boldsymbol{p}^{lowerlimb} \in \mathbb{R}^{178}$, $\boldsymbol{p}^{torso} \in \mathbb{R}^{178}$. In this setting, the total number of conservative operations $N = 3 = N_{1st} + N_{2nd}$.

5 EXPERIMENTS AND DISCUSSIONS

We hypothesize that AutoAugHAR is more effective than manually selecting existing DA approaches without additional expert-based optimizations. Thus, AutoAugHAR must not perform worse than any of the existing techniques given a specific sensor-based HAR task using deep learning techniques. To validate the effectiveness and universality of the proposed **AutoAugHAR**, we conducted extensive evaluations.

5.1 Experiment Setup

5.1.1 Datasets. Eight datasets are selected to represent a broad spectrum of sensing modalities, sampling frequencies, and activity classifications. These datasets include: Human Activities and Postural Transitions (**HAPT**) [55], PAMAP2 Physical Activity Monitoring Data Set (**PAMAP2**) [54], Opportunity (**OPPO**) [11], RealWorld HAR (**RW**) [64], Daily and Sports Activities Data Set (**DSADS**) [7], Wireless Sensor Data Mining (**WISDM**) [35], Daphnet Gait (**DG**) [6] and **GesHome** [49]. Table 5 provides a statistical summary of each dataset, covering aspects such as sampling frequency, number of classes, sensor type, and number of channels. Sensors are grouped into different modalities based on their mounting locations, as depicted in the figure at the top of the aforementioned table. Moreover, we maintained the sizes of the sliding windows for each data set consistent

with the configurations of previous studies [23, 43, 58, 76]. During training, data are split using a sliding window with an overlap of 50%. For test data, we shift the window forward by just one time step [31].

Data preprocessing consists of two steps: the separation of movement and gravity components and the normalization of the data. To separate the acceleration signal into body acceleration and gravity components, the methodology delineated in [5] is followed. Specifically, a Butterworth low-pass filter with a cutoff frequency of 0.3 Hz is employed [68]. Subsequent to this separation, each signal undergoes the z-score normalization before entering the DA transformation.

5.1.2 HAR Models. In order to demonstrate the generalizability, we considered five diverse HAR models in the experiments. Multi-branch Convolutional Neural Network (**MCNN**) [47] applies a sensor-based late fusion technique. It employs individual convolutional networks to extract features from each sensor modality and then uses a fully connected layer to fuse the learned features across all modalities. To improve the capacity to extract temporal information, the hybrid model DeepConvLSTM (**DCL**) [51] incorporates LSTM layers into the individual convolution network. In order to address the 'forgetting' limitations of LSTMs, DeepConvLSTM-Attention (**DCL-A**) [48] augments DeepConvLSTM with temporal attention, facilitating global temporal information exchange in a single forward pass. Attend-Discriminate (**Attend**) [1] further improves performance on multiple HAR datasets by specifically learning the interactions between channels at each time-step through an attention mechanism. A lightweight model, **TinyHAR** is carefully designed for local information extraction, cross-channel information interaction, cross-channel information fusion, and global temporal information extraction and fusion. The structures of all the aforementioned models align with their original descriptions in the cited papers. Owing to their distinct architectures, these models exhibit different capabilities in capturing local context information, long-time dynamics, and cross-modality information.

5.1.3 Compared Data Augmentation Approaches. We compare the proposed **AutoAugHAR** against the following seven DA techniques in addition to a baseline (training without DA): These techniques can be categorized into three groups: traditional DA techniques (MixUp, CutMix), generative models (SFN [45], ActivityGAN, SF-DM), and DA optimization framework (w-augment). The brief description of each technique is as follows.

SDA [30] introduces a DA pipeline that incorporates time-warping and data masking, drawing inspiration from the SpecAugment method [52] used in language processing. Unlike the original SpecAugment approach, SDA proposes different data masking strategies. Based on their experimental findings, we implemented the random masking strategy.

MixUp [1] and **CutMix** [2] are also included in the candidate operators, please refer to section 4.3.1 for more details.

SFN [45] draws inspiration from sample-pair-based augmentation strategies. Instead of using hand crafted techniques like MixUp and CutMix, SFN generates virtual samples using a 4-layer LSTM autoencoder (AE). This LSTM AE is cascaded with the HAR model through a MixUp fusion style to form a unified network that can be trained end-to-end.

ActivityGAN [39]: This method presents a GAN-based framework for generating synthetic sensor-based data. The framework consists of a generator model and a discriminator model. The generator model uses a stack of 1D-convolution and 1D-transposed convolution layers to generate synthetic sensor data, and the discriminator model employs 2D-convolution networks to distinguish between real and synthetic data. After training this GAN-based framework, the generator model is unitized for DA. In the experiments, the configuration of ActivityGAN aligns with the original study's design as described in [39].

SF-DM [81] proposes an unsupervised statistical feature-guided diffusion model for sensor-based HAR. By conditioning the diffusion model on statistical information, SF-DM can generate diverse and representative synthetic sensor data. The structure of the diffusion model and training setup are consist with the original paper.

***w*-augment** [19] is very similar to our proposed framework: both are specifically designed to learn the optimal weight of each DA sub-policy during the training phase. In *w*-augment, for all DA sub-policies, a weight vector with a dimension equals to the number of sub-policies is initialized with equal weights. During the optimization process, the training loss is utilized to update the weights of each DA sub-policy. *w*-augment aims to prioritize sub-policies by assigning them larger weights. Following the settings in the original paper, only one-step sub-policies are considered. In this experiment, the included one-step sub-policies for *w*-augment are the DA methods summarized in Table 1.

It is worth nothing that for all DA techniques, original samples are incorporated into the training process.

5.1.4 Training&Evaluation Protocol. During the evaluation process, the performance of all models across various datasets is assessed utilizing the Leave-One-Subject-Out (LOSO) Cross-Validation (CV) approach. Specifically, in each CV iteration, the data corresponding to one subject is designated as the test set, while the data pertaining to all other subjects forms the training/validation set. The ratio of the training set to the validation set is fixed at 9:1, and the DA policy is applied exclusively to the training set. We use macro average F1-score ($F1_M$) as evaluation metric, the mean $F1_M$ is further computed across all subjects. We repeat the process five times with random seeds ranging from 1 to 5. The mean and standard deviation $F1_M$ are used to compare the models. We use Adam optimizer [34] with default parameters and an initial learning rate of $\xi_w = 10^{-4}$ to optimize the model weights $\boldsymbol{w}$. The learning rate decays by 0.9 after a 10-epoch patience threshold is reached. The maximum training epoch is set to $epoch_{maximum} = 200$, and the batch size remains fixed at 256.

To train the proposed **AutoAugHAR**, we use an additional Adam optimizer for the optimization of the policy parameters $\boldsymbol{\alpha}$ with the learning rate $\xi_\alpha = 5 \times 10^{-3}$, momentum $\beta = (0.5, 0.999)$ and weight decay 10^{-3}. Parameters $\boldsymbol{\alpha}$ are initialized to 10^{-3}. The temperature τ is initially set to $\tau_0 = 5.0$, and its decay equation is as follows, $\tau = \tau_0 \times exp\,(-0.05 \times epoch)$. The 200 training epochs are divided into two phases by $epoch_{search} = 130$. The first 130 epochs are used for the DA sub-policies optimization and model training. During this phase, the ratio of the training set to the validation set is 1:1. Specifically, 50% of the available data is utilized to train the model weights, and the remaining 50% is allocated for updating the weights of the augmentation sub-policies. After 130 epochs, the second phase commences, focusing exclusively on the optimization of the model weights. For this latter phase, 80% of the prior phase's validation set is incorporated into the training set. At the same time, 20% of the prior phase's validation set (equivalent to 10% of the available data) is reserved for the selection of the final best model. Note that the model weights are not reinitialized at the beginning of the second phase. Throughout this latter phase, the $\boldsymbol{\alpha}$ parameters for the sub-policies are fixed and no longer subject to optimization. The data will be transformed in accordance with the optimized DA sub-policy distributions.

5.2 Comparison to State-of-the-art

The results presented in Figures 6 and 7 provide an exhaustive comparison of various DA techniques across multiple datasets and models. The bars represent the mean of $F1_M$ and its standard deviation. Each row in the figure corresponds to the performance on a specific dataset. Furthermore, each row is divided into five groups, each representing the performance of one HAR model under different DA algorithms. To determine the statistical difference in performance between the two DA algorithms, we employed the Mann-Whitney U test [71]. Bold items mark the statistically significant best result with a p-value less than 0.05.

Across all datasets and models, applying DA techniques leads to an improvement of performance compared to the baseline that does not incorporate DA. Among the DA techniques examined, AutoAugHAR stood out, achieving the best results in 38 out of 40 comparison experiments. SFN also exhibits commendable performance, followed by SF-DM, MixUp and CutMix. SFN, MixUp and CutMix are sample-pair-based approaches. They generate virtual samples while preserving label semantic information, underscoring the importance of label-preserving DA strategies in HAR tasks.

Fig. 6. Classification performance on five datasets, each containing multiple modalities.

In contrast, *w*-augment shows inconsistent results in all datasets, performing particularly poorly on the DG and OPPO datasets. While *w*-augment and AutoAugHAR both aim to allocate more weight to beneficial augmentation sub-policies during training, they diverge in their augmentation sub-policy weight update procedures. Specifically, *w*-augment employs training loss for augmentation sub-policy weight optimization, while the proposed AutoAugHAR leverages validation loss. This difference makes *w*-augment more prone to over-fitting, potentially favoring "easy-to-learn" augmentation sub-policies. "Easy-to-learn" augmentation sub-policies might significantly

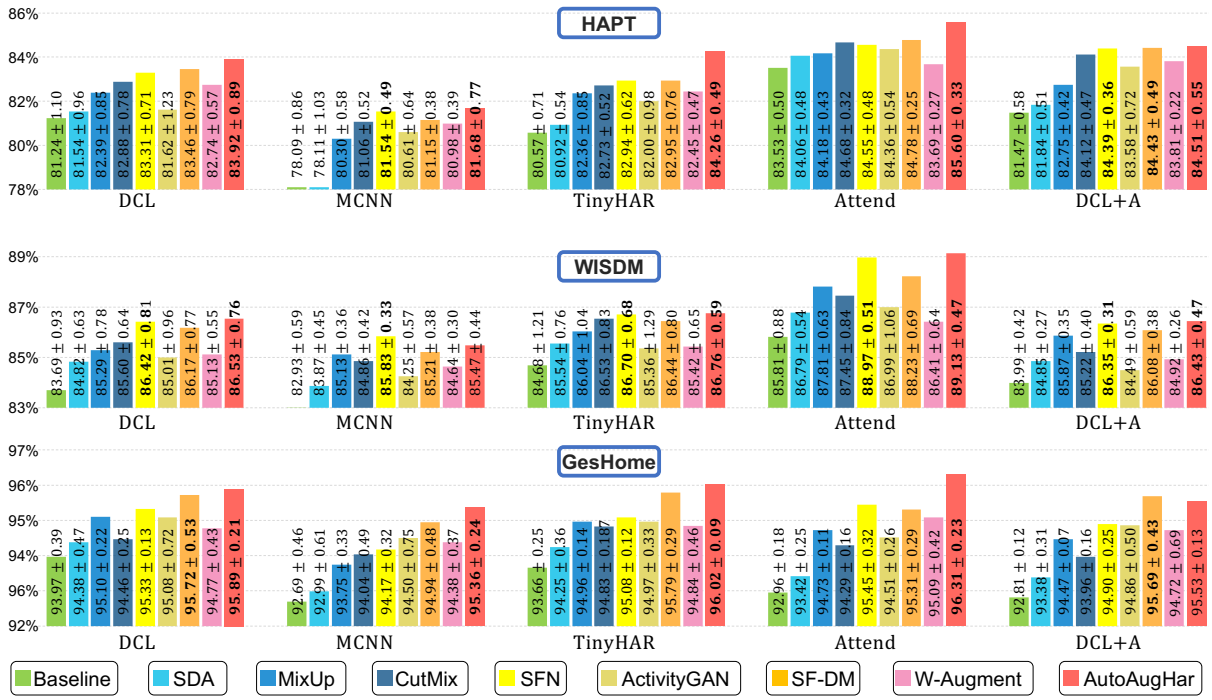


Fig. 7. Classification performance on three datasets, each containing a single modality.

reduce training loss, but they often fail to bridge the distribution gap between training and testing datasets. This shortcoming is especially evident on the OPPO and DG datasets, characterized by challenges such as limited subjects and intra-class variability.

ActivityGAN improves the performance of the model, but it does not reach the extent achieved by AutoAugHAR, SFN, SF-DM, MixUp or CutMix. This can be attributed to the separate training procedures of the generator and the HAR model, potentially leading to sub-optimal solutions. The generator within ActivityGAN is trained to produce virtual samples that match the original data distribution, often neglecting the need to bridge the gap with data not previously encountered. These virtual samples may not align optimally with the downstream HAR model's task requirements. In contrast, the standout performance of AutoAugHAR and SFN can be attributed to their end-to-end training approach. This ensures that the virtual data both mirrors the original distribution and improves the model's performance on unfamiliar data.

Compared to ActivityGAN, another generative model, SF-DM has shown superior performance, especially notable in its outperformance over SFN across three datasets: DSADS, HAPT, and GesHome. However, it still lags behind the proposed model, AutoAugHar. In datasets collected from a relatively larger number of subjects, SF-DM demonstrates its impressive ability to generate diverse and complex data. However, it encounters challenges with certain datasets, particularly the OPPO dataset. This could be attributed to two primary factors. First, the OPPO dataset, which consists of data from only four subjects, exhibits a highly varied data distribution. Similar to ActivityGAN, the separate training of the data generator and the HAR model hinders the use of validation loss as guidance for data generation. Second, the simple denoising model employed by SF-DM fails to deal with datasets such as OPPO, which feature 42 channels and 18 classes. The structure of the model does not adequately

address the characteristics of HAR datasets. Literatures [16, 50] suggest that the success of a diffusion-based approach depends greatly on the model's design and its denoising configuration. Therefore, there is considerable room for improvement in the performance of diffusion-based techniques.

Handcrafted algorithms, MixUp and CutMix, produce consistent results across all datasets. However, their performance varies among different datasets and models. For example, MixUp outperforms CutMix on the RW and PAMAP2 datasets, whereas CutMix excels over MixUp on the OPPO and DSADS datasets. This variability highlights the importance and need for automated DA techniques, such as AutoAugHAR, which autonomously determine the best techniques or combine them.

The effectiveness of the SDN technique, is generally less impressive compared to MixUp and CutMix. SDN achieves only modest improvements on most datasets and can sometimes even reduce model performance. This underperformance is mainly due to SDN's dependence on a predefined DA policy, which might not be suitable for all scenarios.

AutoAugHAR surpasses SFN on most datasets and models, though it shows comparable performance on the WISDM dataset. We attribute this observation to two main reasons: 1) These though MixUp-style generated samples may appear completely different from the original sequences and become meaningless from a human perspective [2]. As past comparisons between CutMix and MixUp indicate, each sample-pair-based technique has its unique advantages. The adoption of the MixUp-style fusion approach in SFN could potentially reduce its efficacy in specific scenarios. 2) Sample-pair-based algorithms are constrained in their capacity to explore more diverse data domains, being entirely reliant on data sample fusion. In contrast, the proposed AutoAugHAR can effectively select from various sample-pair-based methods and combine them with random transformation methods to produce more diverse data.

In the context of the WIDSM and HAPT datasets, AutoAugHAR and SFN exhibit comparable performance. On both datasets, AutoAugHAR and SFN obtained the best performance 6 times without significant differences. These datasets possess shared characteristics, including the utilization of a singular sensor, the categorization of everyday activities, and data sourced from over 30 subjects. These factors make the datasets sufficiently representative for their tasks, eliminating the need for random augmentation to enhance data diversity. Given the singular modality in these datasets, AutoAugHAR can't optimize on a modality basis to boost performance. However, it's worth noting that SFN's integration with an additional LSTM AE encoder into the HAR model enlarges the model's size and necessitates expert knowledge for its design. In contrast, AutoAugHAR doesn't alter the HAR model's structure or increase its size.

5.3 Ablation Study

In order to assess the contribution of the design underlying the proposed AutoAugHAR, three variants of AutoAugHAR were subjected to comparative analysis: AutoAugHARrandom, AutoAugHARbasic, and AutoAugHARNoModality.

AutoAugHARrandom, as elaborated in section 3.2, does not optimize the categorical distribution for DA sub-policies. It adopts a uniform distribution for the application of these DA sub-policies.

AutoAugHARbasic, detailed in section 4.1, optimizes the categorical distribution for DA sub-policies. However, it doesn't differentiate between sample-pair-based DA methods and random transformation methods, and it overlooks multi-modality considerations.

AutoAugHARNoModality follows a two-stage structure like the proposed AutoAugHAR but doesn't account for multi-modality.

When the total conservative operation count N is set to 3, the total number of augmentation sub-policies for AutoAugHARrandom and AutoAugHARbasic surpasses 3000, even after eliminating redundant ones. Given the expansive search space, the performance for both AutoAugHARrandom and AutoAugHARbasic is poor.

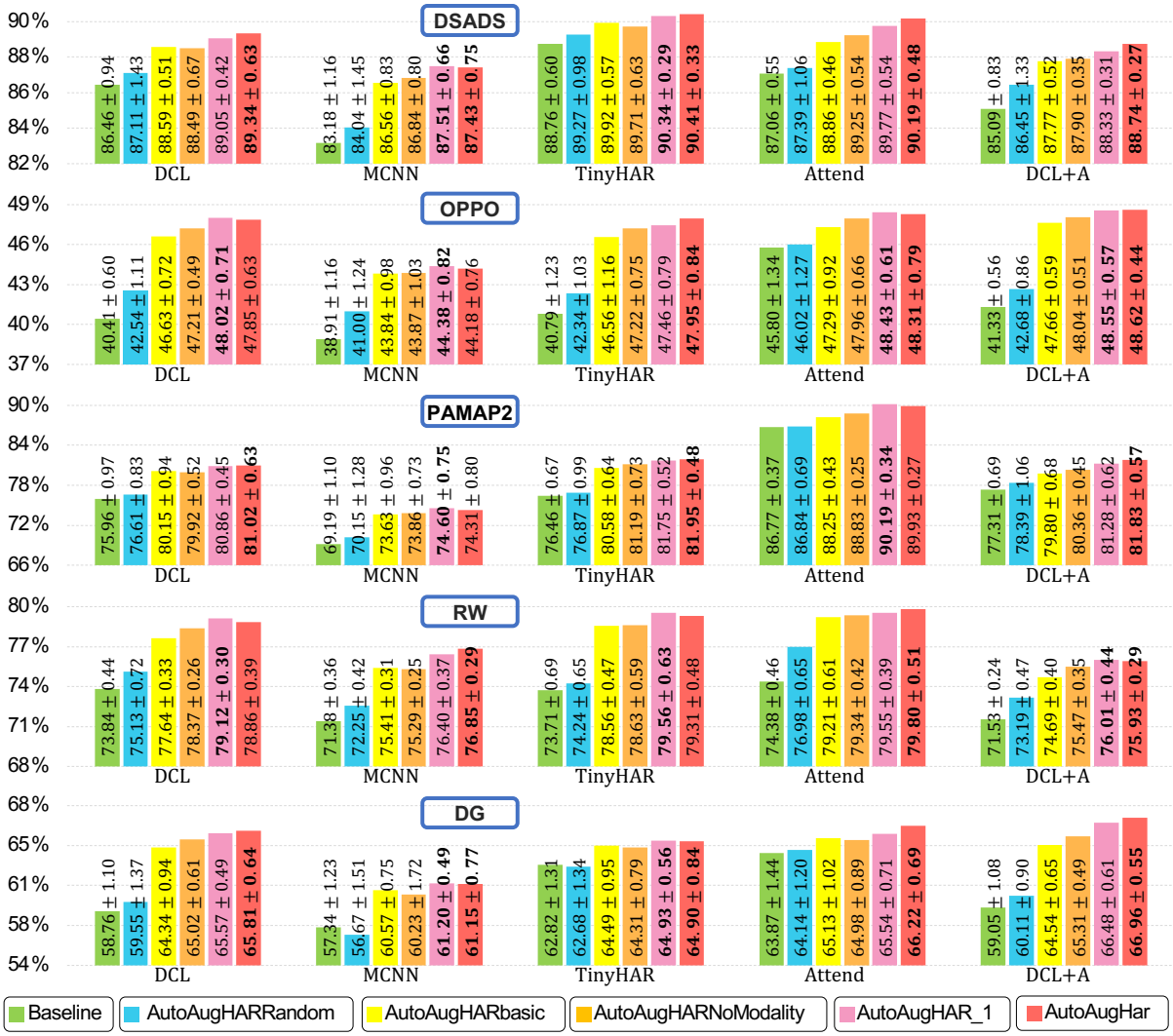


Fig. 8. Classification performance on five datasets, each containing multiple modalities. In this figure, we compare the proposed AutoAugHAR with several of its variants to validate the contributions of its design.

In order to conduct a effective ablation study, we reduced N to 2. With $N = 2$, after removing redundant and identical augmentation sub-policies, the number of augmentation sub-policies for AutoAugHARrandom and AutoAugHARbasic becomes $P_{17}^{16} - 16 - 3 \times 2 - 3 \times 2 - 2 - 2 - 2 = 238$. For a fair comparison, we set N_{1st} and N_{2nd} to 1 for each stage in AutoAugHARNoModality. Furthermore, we also re-trained AutoAugHAR with $N_{2nd} = 1$ in the second stage, denoted as AutoAugHAR_1. Experiments were specifically conducted on datasets characterized by the presence of multiple modalities, thereby validating the contribution of multi-modality optimization in the proposed AutoAugHAR. The results are visually represented in Figure 8.

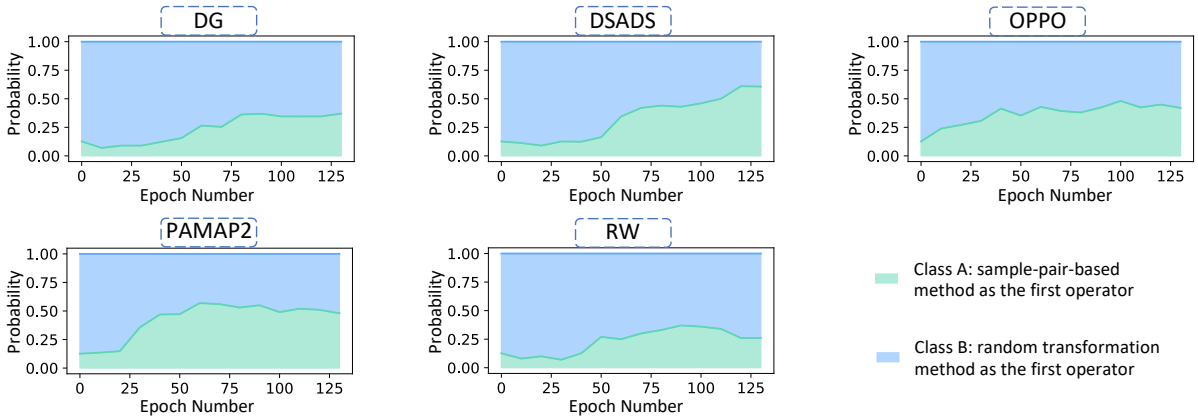


Fig. 9. Evolution of augmentation sub-policies probabilities with training epochs. These examples are derived from training the Attend model.

It can be observed that AutoAugHARRandom's performance is unstable. For instance, on the DG dataset, it had a detrimental impact on the MCNN model's performance. Furthermore, its improvements are marginal compared to the performance of CutMix and MixUp, as illustrated in Figures 6 and 7. Although the stochastic application of DA techniques like AutoAugHARRandom is a conventional procedure in computer vision tasks, it proves unsuitable for HAR tasks. The inherent nature of HAR data, which are more sensitive to perturbations compared to image data, can lead to distortions in label semantic information due to excessive perturbations.

By employing the AutoAugHARbasic algorithm, after the optimization of weights for augmentation sub-policies, a significant improvement in performance compared to AutoAugHARRandom was noted. This observation validates the benefits of automatic optimization of DA sub-policies.

When compared to AutoAugHARbasic, AutoAugHARNoModality mostly demonstrated marginally superior performance. Although the forced ordering of these two operator categories might limit the diversity of augmentation sub-policies, it effectively reduces the search space, which helps optimization. We believe that augmentation sub-policies, when arranged in this specific order, garner more attention during training. To validate this hypothesis, we further examined the evolution of the probability distribution during the AutoAugHARbasic optimization process. Figure 9 illustrates this evolution while training the Attend model on five datasets. For clarity, the 238 augmentation sub-policies were grouped into two classes. Class A consists of sub-policies where sample-pair-based methods are applied as the first operator, totaling 30 sub-policies. Class B includes sub-policies where random augmentation is the initial operator, with 208 sub-policies in this class. The probability assigned to each class is the sum of its constituent probabilities. Given the limited number of sub-policies in Class A, its initial sampling probability is relatively low. However, as optimization continues, there's a noticeable increase in this class's probability. Even though Class A accounts for a small portion of the total sub-policies, its probability exceeds 40% by the end of the optimization process. Remarkably, for the DSADS dataset, this probability reaches 60%. We observed that sub-policies that prioritize the sample-pair-based method as the primary operation tend to receive higher weights, further affirming the rationale behind AutoAugHAR's design.

Compared to AutoAugHARNoModality, AutoAugHAR_1 consistently outperformed AutoAugHARNoModality, highlighting the benefits of considering the multi-modal nature of HAR task. This finding underscores the contribution of optimizing each modality separately.

To understand the influence of the number of operations N on performance, we included the results of AutoAugHAR (with $N_{2nd} = 2$ in the second stage) from previous experiments in Figure 8. Our analysis revealed that AutoAugHAR achieved optimal results in 18 instances, while AutoAugHAR_1 did so in 13 instances. Although AutoAugHAR often had the edge over AutoAugHAR_1, the difference in performance between the two was marginal. An increase in the number of operations indeed offers a more diverse augmentation policy. However, this benefit is offset by the challenges of a larger search space and the potential for excessive transformations due to the increased operational steps.

In summary, the proposed AutoAugHAR demonstrates excellent performance for devices driven by IMUs, evidenced by its adaptability across eight diverse IMU datasets, versatility in enhancing various HAR models, superiority over existing DA algorithms, and effectiveness in handling the multi-modality nature of IMU data. The contribution of each design element within the framework is further validated through an ablation study on five IMU-based datasets. Despite the broad spectrum of IMU-based applications, the core task remains multivariate time series classification, making AutoAugHAR apt for any IMU-based HAR application. It is important to note that the proposed modality-specific approach retains its potential when applied to IMUs. Beyond sensor placement, by considering different sensor types in IMUs as separate modalities, the AutoAugHAR framework can potentially further enhance performance by optimizing DA policies for each sensor type. Moreover, as demonstrated in section 5.5, the framework exhibits flexible extensibility. This advantage facilitates the modification and addition of candidate operators to accommodate a variety of IMU-based application scenarios.

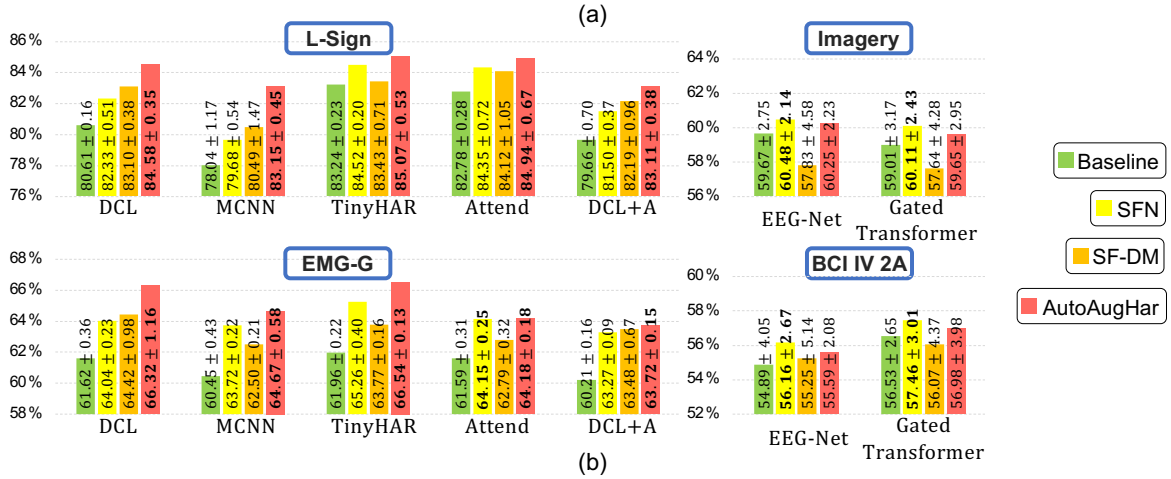
5.4 Generalization on adjacent Domains

The proposed AutoAugHAR framework theoretically holds potential for adaptation to various other multivariate time series classification domains. To assess the framework's applicability in different contexts, we tested four additional datasets, each characterized by distinct data acquisition sensors. The Language-Sign (L-sign) dataset [53] employs a combination of piezoresistive, gyroscopic, and accelerometric sensors to classify the Polish Sign Language alphabet. The Electromyography Gesture (EMG-G) recognition dataset [38] uses data captured via the Myo armband from the upper limb to classify seven grasp gestures. Additionally, the Physionet Imagery (Imagery) [20] and BCI IV 2A [9] datasets utilize electroencephalography (EEG) recordings for motor imagery classification. Figure 10 (a) presents the statistical details of these datasets and outlines the experimental setup. Considering that the imagery data contains data from 109 subjects, we followed the CV setup from the work [69]. For gesture classification tasks, the same HAR models used in previous experiments were employed. For EEG classification tasks, the EEG model [36] and the gated transformer model [66] were implemented. Due to the extensive nature of the experimental work, the comparison was limited to two state of the art DA techniques: the SFN [45] and SF-DM [81], which have proven effective in previous experiments.

Figure 10 (b) illustrates that the improvements in performance achieved by the proposed AutoAugHAR model are particularly notable in the context of the L-sign and EMG-G datasets, surpassing the SFN and SF-DM methods. In contrast, on the Imagery and BCI IV 2A datasets, both the SFN model and AutoAugHAR showed comparatively modest improvements in performance. The diffusion-based SF-DM model actually decreases performance on these two dataset, probably because motor imagery classification benefits from a wider and higher frequency range [10, 26]. The suboptimal performance of the SF-DM may be linked to the Frequency Principle [74], which posits that deep learning models inherently exhibit difficulties in learning and generalizing high-frequency data, displaying a tendency to favor low-frequency information. Furthermore, the denoising process in the SF-DM model, which lacks explicit guidance, might inadvertently remove high-frequency patterns essential for EEG classification. To explore the reasons behind the limited improvement offered by AutoAugHAR on these two datasets, we separately applied the candidate DA operators. We found that the DA operators included in the second stage might not be ideally suited for EEG classification tasks. As mentioned in section 4.3.2, these DA

Dataset	# Sensor Channels	# Subjects	Sampling Frequency	Sliding Window Size	Sensor Type	# Classes	Classes Information
L-Sign	16	16	25	3 s	PRS, Acc, Gyro	36	36 Polish Sign Language alphabet
EMG-G	8	33	200	0.2 s	EMG	7	7 different grasp gestures
Imagery	64	109	160	4 s	EEG	4	4 movement imagery tasks
BCI IV 2A	22	9	250	4.5 s	EEG	4	4 movement imagery tasks

(a)



(b)

Fig. 10. Classification performance across four datasets from adjacent domains.

operators were primarily selected to simulate sensor variations or activity changes in HAR tasks, which may not align well with the specific requirements of EEG classification.

5.5 Extensibility and Flexibility

Our framework is designed to allow for the straightforward substitution of candidate DA operators, making it adaptable to the unique requirements of various domains. Drawing on an extensive review of the prevalent DA operators for EEG classification [57], we have selected four types of DA operators considered to be the most effective. A total of 13 candidate DA operators, each with different parameter ranges, were included in this experiment. These operators are as follows: TimeReverse (Reverse) [56]. Smooth Time Mask (STM) [46], with a smoothing length parameter Δt ranging within $[0\%, 10\%]$, $[10\%, 20\%]$, $[20\%, 30\%]$, $[30\%, 40\%]$ of the input window size. Fourier Transform Surrogate (FTS) [61], which involves modifying the phase of the signal in the Fourier domain. The change of phase $\Delta\varphi$ ranges within $\left[0, \frac{\pi}{2}\right]$, $\left[\frac{\pi}{2}, \pi\right]$, $\left[\pi, \frac{3\pi}{2}\right]$, $\left[\frac{3\pi}{2}, 2\pi\right)$. Channel-Dropout [59], which randomly drops channels from the EEG signal. The channel drop probability p_{drop} ranges within $[0, 0.25]$, $[0.25, 0.5]$, $[0.5, 0.75]$, *and* $[0.75, 1]$. For detailed implementations of these DA operators, please refer to Work [57].

Figure 11 illustrates the significant improvement resulting from the integration of new DA candidate operators into the second stage of AutoAugHAR, denoted as AutoAugHAR(N). It is evident that these operators have varying levels of effectiveness across the two datasets. Specifically, the FTS operator shows outstanding performance on the Imagery dataset compared to the other candidates. However, the STM operator emerges as the most effective on the BCI IV 2A dataset. The AutoAugHAR framework excels at identifying and combining the most efficient DA operators, thus enhancing overall performance. While the SFN model and the diffusion model can also be

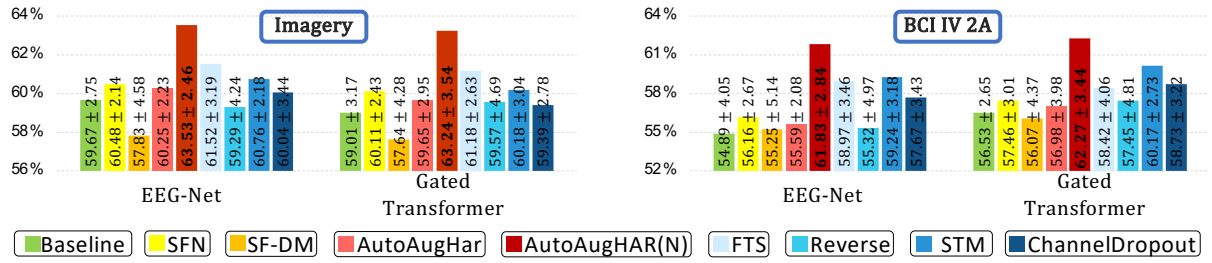


Fig. 11. Classification performance using the new candidate DA operator on EEG datasets.

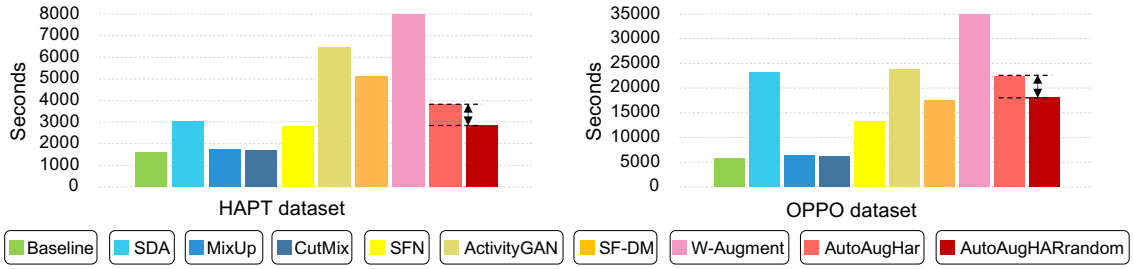


Fig. 12. The training time for one iteration of the LOSO-CV process.

improved, such an enhancement would require specific adjustments to the denoising model's structure and its training methodology.

5.6 Training Overhead

Figure 12 depicts the training time required for a single iteration of the LOSO-CV process on two datasets, primarily differentiated by their number of sensor channels. Among the three traditional DA algorithms, MixUp and CutMix lead to a slight increase in training time. In contrast, the SDA algorithm significantly extends training time due to its default application of the 'time-warp' operation, which requires re-interpolation for each sequence. This impact is more pronounced on the OPPO dataset, where the training time is further extended due to the increased number of channels.

In terms of generative algorithms, SFN is the most time-efficient, followed by ActivityGAN and SF-DM. The longer training durations for ActivityGAN and SF-DM can be attributed to their separate training processes, in which the generator/denoising model is first trained and then incorporated into the dataloader for HAR model training. SF-DM's intelligent design uses unlabeled data as input instead of random noise, resulting in shorter training times compared to ActivityGAN.

The training time required for AutoAugHAR is lower than that of most generative augmentations but higher than SFN. The training time primarily consists of three components: HAR model training, data transformation using DA policies, and DA policy optimization. The extra training time is largely attributable to data transformations using DA policies. As shown in Figure 12, even AutoAugHARrandom without policy optimization significantly increases training time, particularly due to time-intensive operations like 'time-warp' and 'magnitude-warp'. The additional training time required to update the DA policy weights is indicated by the arrows in Figure 12. The reason why the training time for DA policy optimization is acceptable is that the weights for DA policies are

updated using gradients obtained through a backward process in the Adam algorithm. The training time of *w*-augment is substantially higher than AutoAugHAR because all available DA operators are applied simultaneously to each data sample.

6 CONCLUSION

The consistent and superior improvement across a wide range of selected target datasets and tasks suggests that automated DA techniques such as AutoAugHAR have potential as a go-to technique for many HAR applications. This is contrasted by the inconsistent performance of other methods, which emphasises the need for careful selection and experimentation based on the specific task and data at hand. The approach presented in this paper has the advantage that the gradient based two-stage scheme can be efficiently integrated into model training. Unlike many other techniques that address cross-subject generalisation during model training, the DA schemes presented are model agnostic. While it is unlikely that there are one-size-fits-all augmentation techniques for the wide variety of sensor-based HAR tasks, the presented model is easily extensible. Future efforts will be directed towards integration into our user-centric HAR solutions. By open-sourcing the framework under a permissive licence, we also encourage others to confirm the consistent improvements we have seen, but also to report possible adverse effects in applications that we cannot completely rule out. Both the integration of the presented approach into larger, scalable hyperparameter optimisation schemes and its combination with other data synthesis techniques are interesting avenues to pursue in order to further improve the performance and data efficiency of HAR applications.

ACKNOWLEDGEMENT

This work has been partially supported by the German Ministry of Research and Education as part of the SDIL (01IS19030A), the German Ministry for Research and Education as part of SDI-S (01IS22095A) and the Carl-Zeiss-Foundation as part of "stay young with robots" (Jubot) project.

REFERENCES

[1] Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezatofighi, and Damith C Ranasinghe. 2021. Attend and Discriminate: Beyond the State-of-the-art for Human Activity Recognition Using Wearable Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021), 1–22.

[2] Nafees Ahmad and Ho-fung Leung. 2023. ALAE-TAE-CutMix+: Beyond the State-of-the-Art for Human Activity Recognition Using Wearable Sensors. In *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 222–231.

[3] Luay Alawneh, Tamam Alsarhan, Mohammad Al-Zinati, Mahmoud Al-Ayyoub, Yaser Jararweh, and Hongtao Lu. 2021. Enhancing Human Activity Recognition Using Deep Learning and Time Series Augmented Data. *Journal of Ambient Intelligence and Humanized Computing* (2021), 1–16.

[4] G Anandalingam and Terry L Friesz. 1992. Hierarchical optimization: An introduction. *Annals of Operations Research* 34 (1992), 1–11.

[5] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. 2013. A Public Domain Dataset for Human Activity Recognition Using Smartphones.. In *Esann*, Vol. 3. 3.

[6] Marc Bachlin, Daniel Roggen, Gerhard Troster, Meir Plotnik, Noit Inbar, Inbal Meidan, Talia Herman, Marina Brozgol, Eliya Shaviv, Nir Giladi, et al. 2009. Potentials of Enhanced Context Awareness in Wearable Assistants for Parkinson's Disease Patients with the Freezing of Gait Syndrome. In *2009 International Symposium on Wearable Computers*. IEEE, 123–130.

[7] Billur Barshan and Murat Cihan Yüksek. 2014. Recognizing Daily and Sports Activities in Two Open Source Machine Learning Environments Using Body-Worn Sensor Units. *Comput. J.* 57, 11 (2014), 1649–1667.

[8] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).

[9] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. 2008. BCI Competition 2008–Graz data set A. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology* 16 (2008), 1–6.

[10] Timothy J Buschman, Eric L Denovellis, Cinira Diogo, Daniel Bullock, and Earl K Miller. 2012. Synchronous oscillatory neural ensembles for rules in the prefrontal cortex. *Neuron* 76, 4 (2012), 838–846.

[11] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. 2013. The Opportunity Challenge: A Benchmark Database for On-Body Sensor-Based Activity Recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.

[12] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–40.

[13] Seungeun Chung, Jiyoun Lim, Kyoung Ju Noh, Gague Kim, and Hyuntae Jeong. 2019. Sensor Data Acquisition and Multimodal Sensor Fusion for Human Activity Recognition Using Deep Learning. *Sensors* 19, 7 (2019), 1716.

[14] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501* (2018).

[15] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical Automated Data Augmentation with A Reduced Search Space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 702–703.

[16] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

[17] Xuanyi Dong and Yi Yang. 2019. Searching for a robust neural architecture in four gpu hours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1761–1770.

[18] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A Survey of Data Augmentation Approaches for NLP. *arXiv preprint arXiv:2105.03075* (2021).

[19] Elizabeth Fons, Paula Dawson, Xiao-jun Zeng, John Keane, and Alexandros Iosifidis. 2021. Adaptive weighting scheme for automatic time-series data augmentation. *arXiv preprint arXiv:2102.08310* (2021).

[20] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation* 101, 23 (2000), e215–e220.

[21] Maxime Goubeaud, Philipp Joußen, Nicolla Gmyrek, Farzin Ghorban, Lucas Schelkes, and Anton Kummert. 2021. Using Variational Autoencoder to Augment Sparse Time Series Datasets. In *2021 7th International Conference on Optimization and Applications (ICOA)*. IEEE, 1–6.

[22] Emil Julius Gumbel. 1948. *Statistical theory of extreme values and some practical applications: a series of lectures*. Vol. 33. US Government Printing Office.

[23] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. *arXiv preprint arXiv:1604.08880* (2016).

[24] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. 2020. Faster autoaugment: Learning augmentation strategies using backpropagation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*. Springer, 1–16.

[25] Alexander Hoelzemann, Nimish Sorathiya, and Kristof Van Laerhoven. 2021. Data Augmentation Strategies for Human Activity Data Using Generative Adversarial Neural Networks. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 8–13.

[26] Yimin Hou, Lu Zhou, Shuyue Jia, and Xiangmin Lun. 2020. A novel approach of decoding EEG four-class motor imagery tasks via scout ESI and CNN. *Journal of neural engineering* 17, 1 (2020), 016048.

[27] Shuokang Huang, Po-Yu Chen, and Julie McCann. 2023. DiffAR: adaptive conditional diffusion model for temporal-augmented human activity recognition. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. 3812–3820.

[28] Brian Kenji Iwana and Seiichi Uchida. 2021. An Empirical Survey of Data Augmentation for Time Series Classification with Neural Networks. *Plos one* 16, 7 (2021), e0254841.

[29] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[30] Chi Yoon Jeong, Hyung Cheol Shin, and Mooseop Kim. 2021. Sensor-Data Augmentation for Human Activity Recognition with Time-Warping and Data Masking. *Multimedia Tools and Applications* 80 (2021), 20991–21009.

[31] Artur Jordao, Antonio C Nazare Jr, Jessica Sena, and William Robson Schwartz. 2018. Human Activity Recognition Based on Wearable Sensor Data: A Standardization of the State-of-the-art. *arXiv preprint arXiv:1806.05226* (2018).

[32] Gerasimos Kalouris, Evangelia I Zacharaki, and Vasileios Megalooikonomou. 2019. Improving CNN-Based Activity Recognition by Data Augmentation and Transfer Learning. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Vol. 1. IEEE, 1387–1394.

[33] Hua Kang, Qianyi Huang, and Qian Zhang. 2022. Augmented Adversarial Learning for Human Activity Recognition with Partial Sensor Sets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–30.

[34] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[35] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. 2011. Activity Recognition Using Cell Phone Accelerometers. *ACM SigKDD Explorations Newsletter* 12, 2 (2011), 74–82.

[36] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. 2018. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of neural engineering* 15, 5 (2018), 056013.

[37] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data Augmentation for Time Series Classification Using Convolutional Neural Networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*.

[38] BOREOM LEE. 2023. EMG-EEG dataset for Upper-Limb Gesture Classification. https://doi.org/10.21227/5ztn-4k41

[39] Xi'ang Li, Jinqi Luo, and Rabih Younes. 2020. ActivityGAN: Generative Adversarial Networks for Data Augmentation in Sensor-Based Human Activity Recognition. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*. 249–254.

[40] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. 2020. Dada: Differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780* (2020).

[41] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. 2017. Hierarchical representations for efficient architecture search. *arXiv preprint arXiv:1711.00436* (2017).

[42] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018).

[43] Shengzhong Liu, Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Huajie Shao, and Tarek Abdelzaher. 2020. Giobalfusion: A global attentional deep learning framework for multisensor information fusion. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27.

[44] Chris J Maddison, Daniel Tarlow, and Tom Minka. 2014. A* sampling. *Advances in neural information processing systems* 27 (2014).

[45] Fanyang Meng, Hong Liu, Yongsheng Liang, Juanhui Tu, and Mengyuan Liu. 2019. Sample fusion network: An end-to-end data augmentation network for skeleton-based human action recognition. *IEEE Transactions on Image Processing* 28, 11 (2019), 5281–5295.

[46] Mostafa Neo Mohsenvand, Mohammad Rasool Izadi, and Pattie Maes. 2020. Contrastive representation learning for electroencephalogram classification. In *Machine Learning for Health*. PMLR, 238–253.

[47] Sebastian Münzner, Philip Schmidt, Attila Reiss, Michael Hanselmann, Rainer Stiefelhagen, and Robert Dürichen. 2017. CNN-Based Sensor Fusion Techniques for Multimodal Human Activity Recognition. In *Proceedings of the 2017 ACM international symposium on wearable computers*. 158–165.

[48] Vishvak S Murahari and Thomas Plötz. 2018. On Attention Models for Human Activity Recognition. In *Proceedings of the 2018 ACM international symposium on wearable computers*. 100–103.

[49] Khanh Nguyen-Trong, Hoai Nam Vu, Ngon Nguyen Trung, and Cuong Pham. 2021. Gesture recognition using wearable sensors with bi-long short-term memory convolutional neural networks. *IEEE Sensors Journal* 21, 13 (2021), 15065–15079.

[50] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*. PMLR, 8162–8171.

[51] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* 16, 1 (2016), 115.

[52] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779* (2019).

[53] Jakub Piskozub. 2023. Letters of Polish Sign Language Alphabet. https://doi.org/10.21227/w90m-m764

[54] Attila Reiss and Didier Stricker. 2012. Introducing A New Benchmarked Dataset for Activity Monitoring. In *2012 16th international symposium on wearable computers*. IEEE, 108–109.

[55] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-aware human activity recognition using smartphones. *Neurocomputing* 171 (2016), 754–767.

[56] Cédric Rommel, Thomas Moreau, Joseph Paillard, and Alexandre Gramfort. 2021. CADDA: Class-wise automatic differentiable data augmentation for EEG signals. *arXiv preprint arXiv:2106.13695* (2021).

[57] Cédric Rommel, Joseph Paillard, Thomas Moreau, and Alexandre Gramfort. 2022. Data augmentation for learning predictive models on EEG: a systematic comparison. *Journal of Neural Engineering* 19, 6 (2022), 066020.

[58] Charissa Ann Ronao and Sung-Bae Cho. 2016. Human Activity Recognition with Smartphone Sensors Using Deep Learning Neural Networks. *Expert systems with applications* 59 (2016), 235–244.

[59] Aaqib Saeed, David Grangier, Olivier Pietquin, and Neil Zeghidour. 2021. Learning from heterogeneous eeg signals with differentiable channel reordering. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1255–1259.

[60] Panneer Selvam Santhalingam, Parth Pathak, Huzefa Rangwala, and Jana Kosecka. 2023. Synthetic Smartwatch IMU Data Generation from In-the-wild ASL Videos. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 2 (2023), 1–34.

[61] JTC Schwabedal, JC Snyder, A Cakmak, S Nemati, and GD Clifford. [n. d.]. Addressing Class Imbalance in Classification Problems of Noisy Signals by using Fourier Transform Surrogates. arXiv 2018. *arXiv preprint arXiv:1806.08675* ([n. d.]).

[62] Shuai Shao and Victor Sanchez. 2023. A study on diffusion modelling for sensor-based human activity recognition. In *2023 11th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 1–7.

[63] Connor Shorten and Taghi M Khoshgoftaar. 2019. A Survey on Image Data Augmentation for Deep Learning. *Journal of big data* 6, 1 (2019), 1–48.
[64] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 1–9.
[65] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo. 2021. Selfhar: Improving human activity recognition through self-training with unlabeled data. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 5, 1 (2021), 1–30.
[66] Yunzhe Tao, Tao Sun, Aashiq Muhamed, Sahika Genc, Dylan Jackson, Ali Arsanjani, Suri Yaddanapudi, Liang Li, and Prachi Kumar. 2021. Gated transformer for decoding human brain EEG signals. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 125–130.
[67] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data Augmentation of Wearable Sensor Data for Parkinson's Disease Monitoring Using Convolutional Neural Networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*. 216–220.
[68] Vincent T Van Hees, Lukas Gorzelniak, Emmanuel Carlos Dean León, Martin Eder, Marcelo Pias, Salman Taherian, Ulf Ekelund, Frida Renström, Paul W Franks, Alexander Horsch, et al. 2013. Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity. *PloS one* 8, 4 (2013), e61691.
[69] Xiaying Wang, Michael Hersche, Batuhan Tömekce, Burak Kaya, Michele Magno, and Luca Benini. 2020. An accurate eegnet-based motor-imagery brain–computer interface for low-power edge computing. In *2020 IEEE international symposium on medical measurements and applications (MeMeA)*. IEEE, 1–6.
[70] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2020. Time Series Data Augmentation for Deep Learning: A Survey. *arXiv preprint arXiv:2002.12478* (2020).
[71] Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*. Springer, 196–202.
[72] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. 2019. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10734–10742.
[73] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. 2018. SNAS: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926* (2018).
[74] Zhi-Qin John Xu, Yaoyu Zhang, and Tao Luo. 2022. Overview frequency principle/spectral bias in deep learning. *arXiv preprint arXiv:2201.07395* (2022).
[75] Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. 2019. Understanding and robustifying differentiable architecture search. *arXiv preprint arXiv:1909.09656* (2019).
[76] Ye Zhang, Longguang Wang, Huiling Chen, Aosheng Tian, Shilin Zhou, and Yulan Guo. 2022. IF-ConvTransformer: A framework for human activity recognition using IMU fusion and ConvTransformer. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–26.
[77] Yexu Zhou, Michael Hefenbrock, Yiran Huang, Till Riedel, and Michael Beigl. 2021. Automatic Remaining Useful Life Estimation Framework with Embedded Convolutional LSTM as the Backbone. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part IV*. Springer, 461–477.
[78] Yexu Zhou, Haibin Zhao, Yiran Huang, Till Riedel, Michael Hefenbrock, and Michael Beigl. 2022. Tinyhar: A Lightweight Deep Learning Model Designed for Human Activity Recognition. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers*. 89–93.
[79] Barret Zoph and Quoc V Le. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).
[80] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8697–8710.
[81] Si Zuo, Vitor Fortes, Sungho Suh, Stephan Sigg, and Paul Lukowicz. 2023. Unsupervised Diffusion Model for Sensor-based Human Activity Recognition. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing*. 205–205.