AUGMENTATION ROBUST SELF-SUPERVISED LEARNING FOR HUMAN ACTIVITY RECOGNITION

Cong Xu[⋆] *Yuhang Li*[†] *Dae Lee*[⋆] *Dae Hoon Park*[⋆]
Hongda Mao[⋆] *Huyen Do*[⋆] *Jonathan Chung* [⋆] *Dinesh Nair*[⋆]

[⋆] Amazon Inc.
[†]Yale University

ABSTRACT

Human Activity Recognition (HAR) is widely applied on wearable devices in our daily lives. However, acquiring high-quality wearable sensor data set with ground-truths is challenging due to the high cost in collecting data and necessity of domain experts. In order to achieve generalization from limited data, we study augmentation-based Self-Supervised Learning (SSL) for data from wearable devices. However, there is an issue in one of the most popular SSL approaches, contrastive learning: it is sensitive to the choice of data augmentations. To resolve this, we first propose to combine contrastive learning with generative learning, which is robust to augmentations. Second, we propose an automatic augmentation policy search method to discover the most promising augmentation policy. We empirically verify our approaches on three public HAR datasets. Experimental results show that our proposed SSL approach is robust to augmentations, and delivers higher accuracy than contrastive learning. Additionally, with the searched augmentation policy we are able to further improve the accuracy of HAR task.

***Index Terms*—** Human Activity Recognition, Self-supervised learning

1. INTRODUCTION

Deep learning has rapidly fostered a wide range of intelligent applications, such as computer vision and natural language processing. As a result, the massive deployments of deep learning based applications span from cloud servers to edge devices, *e.g.*, wearables. For example, Human Activity Recognition (HAR) tasks leverage the signal from wearable sensors such as accelerometer and gyroscope to predict daily activities of users.

When building an HAR model with wearables data, a popular approach is to train a multi-class classification model based on the deep neural network [1, 2]. The success of this training method relies on the enormous accurate label annotations of training data. However, annotating the labels of wearable sensor data is laborious and expensive, thereby limiting the scalability of supervised learning. Furthermore, the label annotation is subject to human bias and may create incorrect labels. These factors limit HAR to create a large-scale generalizable model.

In an effort to mitigate the aforementioned problems, Self-Supervised Learning (SSL) algorithms [3, 4, 5] have been applied to HAR tasks. SSL algorithm defines a pretext task and optimizes the model on this pretext task. Based on the sequential characteristics of the wearable sensor data, the pretext task can be defined in many ways, such as data reconstruction [6], data transformation predictions [7], and contrastive comparison [8]. Learning parameters on the pretext task can produce general-purpose latent representations, hence benefiting the fine-tuning on the downstream tasks.

Recently, contrastive learning has been applied to SSL on wearables. Qian et al. [9] conducted a systematic study on the framework of contrastive learning for HAR tasks. They showed the wearable datasets were extremely sensitive to the choice of augmentations in contrastive learning. For example, inferior augmentations could only obtain around 51% downstream task accuracies, while good augmentations can achieve greater than 90% accuracies. This reveals that the augmentations of contrastive learning are not robust for SSL of wearable data.

In this work, we focus on robust SSL for HAR tasks. In particular, our SSL framework contains two parts. First, we combine the idea of contrastive and generative learning [10] by adopting contrastive learning on augmented signals, and reconstructing the original signals from augmented signals like generative learning. The framework enjoys the discriminative power of contrastive learning, as well as the reconstructing power provided by generative learning, which leads to a robust and effective model. Second, we employ an automatic search algorithm to search a sequence of augmentations in our proposed SSL framework to further improve the accuracies in HAR tasks. We empirically verify our method on three HAR datasets and demonstrate its effectiveness.

2. METHODOLOGY

2.1. Preliminaries

We focus on the scenario of self-supervised learning for HAR tasks. Denote the training dataset as $\{\mathbf{x}_i\}_{i=1}^N$, each input signal is a $L \times C$ tensor, where L is the time duration of the signal and C is the dimension from wearable sensors. For example, the accelerometer has $C = 3$ standing for x, y, z axis. For self-supervised learning, we are interested in training a feature extractor $f(\cdot)$ with a pretext task, then using the pre-trained latent representation $f(\mathbf{x})$ to optimize the linear classifier $h(\cdot)$ with ground-truth labels $\{\mathbf{y}_i\}_{i=1}^N$.

Contrastive Learning (CL). The pretext task in CL is to discriminate between two different input signals. As done in [8], CL can use *positive pairs* and *negative pairs* in its objective. For positive pairs, the same input signal is transformed by two different augmentations, $T'(\mathbf{x_i})$ and $T''(\mathbf{x_i})$. Let $\mathbf{z_i'}$ and $\mathbf{z_i''}$ be the latent representation of the corresponding augmented instances, and the feature extractor is expected to minimize the representation distance between $\mathbf{z_i'}$ and $\mathbf{z_i''}$. At the same time, the feature extractor will maximize the distance between any pairs of different input signals in the mini batch, which is assumed to be negative pairs. To this end, the InfoNCE loss [11]

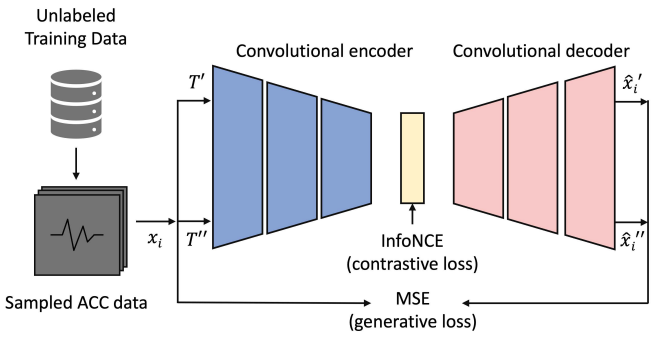


Fig. 1: Our framework of incorporating the contrastive and generative learning.

used in CL can be defined as

$$\ell_i^{\text{con}} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i', \mathbf{z}_i''))}{\exp(\text{sim}(\mathbf{z}_i', \mathbf{z}_i'')) + \sum_{j=1}^{n} \mathbb{1}_{[j \neq i]} \exp(\text{sim}(\mathbf{z}_i^{\dagger}, \mathbf{z}_j^{\dagger}))}, \tag{1}$$

where sim is a similarity function such as cosine similarity, n is the batch size, $\mathbb{1}_{[\cdot]}$ is an indicator function for identifying different examples, and $\mathbf{z}_i^{\dagger}$ and $\mathbf{z}_j^{\dagger}$ denote latent representations of different inputs, $\mathbf{x}_i$ and $\mathbf{x}_j$, augmented by either T' or T''. One can optionally add a temperature hyperparameter to smoothen the loss function.

Generative Learning (GL). Different from CL, GL employs an encoder-decoder structure. In doing so, the original input signal $\mathbf{x}_i$ is first augmented to $T(\mathbf{x}_i)$, extracted to a latent representation $\mathbf{z}_i$, and then decoded as $\hat{\mathbf{x}}_i$ to reproduce the augmented input signal. The loss function in the GL is used to minimize the distance between the augmented input and the decoded output. For example, the L_2 loss function is defined as

$$\ell_i^{\text{gen}} = ||T(\mathbf{x}_i) - \hat{\mathbf{x}}_i||_F^2 \tag{2}$$

where $||\cdot||_F$ denotes the Frobenius norm. GL generally extracts the latent representation that contains essential information to reconstruct the input.

2.2. Proposed Framework

As mentioned previously, the major pretext tasks of SSL can be classified into contrastive-based and generative-based tasks. Both of them can train a feature extractor for latent representation learning, but they both have some flaws. For HAR tasks, as mentioned in [9], the contrastive learning algorithm is extremely sensitive to the choice of data augmentation T and other training hyper-parameters. On the other hand, the learning process of generative learning is more stable, but it fails to learn discriminative power in the latent representation, thus the task performance of GL is usually limited.

Hence, we propose to incorporate both CL and GL into one framework. As shown in Figure 1, the architecture is designed with an encoder-decoder structure, similar to generative learning. The input signal is augmented via two distinct augmentations, $T'(\mathbf{x}_i)$ and $T''(\mathbf{x}_i)$, and then they are encoded to the latent representations, $\mathbf{z}_i'$ and $\mathbf{z}_i''$. After the decoder structure, we can obtain the output signals $\hat{\mathbf{x}}_i'$ and $\hat{\mathbf{x}}_i''$. Our training objective is defined as

$$\ell_i^{\text{dual\_gen}} = ||\mathbf{x}_i - \hat{\mathbf{x}}_i'||_F^2 + ||\mathbf{x}_i - \hat{\mathbf{x}}_i''||_F^2 \tag{3}$$

Here, we expect the output signal can be made as close as possible to the original input signal $\mathbf{x}_i$. In addition, we add an InfoNCE loss to the latent representation, which restrict the latent representation to discriminate different input signals. Overall, the loss can be computed as following, where the λ is a hyper-parameter.

$$\ell_i = \ell_i^{\text{con}} + \lambda \ell_i^{\text{dual\_gen}} \tag{4}$$

For one thing, our framework enjoys a robust training process from generative learning due to the encoder-decoder structure and the reconstruction objective. For another, we explicitly require the latent embedding to be contrastive as well, playing the role of contrastive learning. Implicitly, since Eq. (3) requires two output signals to reconstruct the same original input signal, the latent representations $\mathbf{z}_i'$ and $\mathbf{z}_i''$, are also required to be made as close as possible, acting like positive pairs. Hence, our framework incorporates the advantages of both contrastive and generative learning.

2.3. Searched Augmentations

Even though our framework allows a robust training process regardless of data augmentation, one still needs to choose the most effective data augmentation policy. As done in [9], conventional methods select optimal augmentations based on manual trials, i.e., traversing each possible augmentation policy and selecting the most performant policy.

However, this method has two disadvantages that impair its practical usage. First, traversing each possible policy or different combinations of multiple augmentations is computationally expensive. As an example, [9] only explores a single augmentation choice in one branch (from 11 potential augmentations) requiring 121 trials, which is too expensive in practice. Second, finding the optimal augmentation policy requires labels of the validation dataset. This is impractical since we cannot access the labels during the pre-training phase.

To this end, we propose an end-to-end method to automatically select the best augmentation choice in the SSL pre-training. Inspired by [12, 13], where the authors show that SSL tasks can be used to evaluate network architectures, we choose the model that has the minimum loss during the pre-training task. This proxy task does not require any labels or fine-tuning on downstream tasks, and it can implicitly reflect the performance on downstream tasks. Since the loss is calculated based on Eq. (4), the latent representations we learned enjoy the advantages of both contrastive and generative learning. As a result, they have discriminative power while exhibiting the advantage of GL where the fine-tuning accuracy is insensitive to data augmentation.

For searching the data augmentations, the conventional approach selects the single choice of data augmentation in each branch. This simplifies the search space a lot but may fail to find the optimal search policy. Following [14], we search a sequence of augmentations for each branch. Denoting $\mathbb{O}$ as the augmentation set, which contains a set of candidate augmentations, we define a subpolicy as $\mathcal{O}$, containing 4 augmentations from $\mathbb{O}$. Each augmentation in $\mathcal{O}$ has a probability of being sampled or not, and an intensity parameter for controlling the strength of the augmentations.

We first train the encoder-decoder using pretext task, and search a sequence of augmentations as well as their strengths. Once promising augmentations and their strengths are found through self-supervised pre-training, we remove the decoder and fine-tune a classifier for the HAR task. Therefore, the objective for searching the augmentations can be defined as:

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} \mathcal{L}(\theta_{SSL} | \mathcal{T}(\mathcal{D}_{VAL})) \tag{5}$$

where θ_{SSL} is the parameter trained with SSL pretext task. $\mathcal{L}(\cdot|\cdot)$ evaluates the performance of pretext task given certain network parameters with the non-labeled validation dataset. In other words, the above minimization finds the policy that has the lowest loss on validation dataset $\mathcal{D}_{VAL}$.

3. EXPERIMENTS

We implement all experiments with PyTorch [16] package. We select three popular human activity recognition datasets collected on smartphones and smartwatches: SHAR [15], UCI-HAR [17] and HHAR [18]. We use a 3-layer convolutional neural network (CNN) as our encoder architecture. Each convolutional layer is followed by a BatchNorm layer and a ReLU layer. The decoder is also built with 3 transposed convolutional layers. We use the same candidate augmentations mentioned in [9], including: noise, scale, negate, perm, shuffle, t_flipped, t_warp, resample, rotation, perm_jit, jit_scal.

Unless otherwise specified, we use following hyper-parameters in our experiments. The coefficient λ in the loss function is by default set to 1, consequently the contrastive and generative learning losses contribute equally to the loss function. Contrastive learning loss is calculated using InfoNCE loss, while generative learning loss is obtained by mean squared error. Adam optimizer [19] is adopted to minimize the loss with learning rate of 0.003 and weight decay of 10^{-4}. We train the model with batch size 256 for 120 epochs. The accelerometer signal sliding window lengths of SHAR, UCI-HAR, and HHAR datasets are 151, 128, and 100, respectively.

Table 1: Comparison among different self-supervised learning methods on three HAR datasets. Experiments run on 121 different combinations of data augmentations. "Best"=highest accuracy, "Median"=median accuracy, "Avg±Stddev"=mean accuracy and the standard deviation among all 121 trials. The best measure for each metric (including StdDev) of each data set is in bold.

Dataset	Method	Best	Median	Avg±StdDev
SHAR [15]	Contrastive	92.9	89.2	87.2±6.0
	Generative	91.7	89.1	89.0±**1.3**
	Ours	**94.0**	**91.8**	**91.5**±**1.3**
UCI-HAR [17]	Contrastive	96.7	94.1	92.1±5.6
	Generative	96.0	**94.9**	93.8±**3.5**
	Ours	**97.0**	94.0	**93.9**±4.0
HHAR [18]	Contrastive	92.3	90.8	90.0±2.0
	Generative	93.6	92.0	91.8±**0.8**
	Ours	**94.0**	**92.6**	**92.4**±0.9

3.1. Comparison with Existing SSL

We first verify the robustness of each algorithm against data augmentations. In particular, we compare our framework with contrastive learning and generative learning methods under different data augmentations. All three algorithms require two data augmentations (one per branch), since we have 11 candidate augmentations for each branch, there are totally 121 different combinations. Therefore, we can obtain a heat map, where each entry represents the accuracy for a combination of two data augmentations.

For contrastive learning, we select SimCLR [8] framework to minimize the distance between two augmentations from the same input, and maximize the distance between different input signals. For generative learning, to accommodate the data augmentations in the autoencoder, in each branch we let it reconstruct the augmented input signal, whereas in our framework the goal switches to reconstruct the original input data.

Fig. 2 summarizes the results of fine-tuning accuracy heat maps on the SHAR dataset. It can be seen that the contrastive learning results in Fig. 2a show high variance, i.e., some data augmentation pairs yield low (<70%) accuracies while the others can achieve very good (>90%) accuracies. The highest accuracy of CL is 92.9%, yet the lowest accuracy is only 62.1%. Fig. 2b demonstrates the results of generative learning, where the variance of fine-tuning accuracy has been largely reduced. All entries are in similar green colors, meaning that their accuracies are relatively insensitive to data augmentations. However, the best accuracy of GL is lower than that of CL. For example, the highest entry in Fig. 2a reaches 91.7%, while the highest entry in Fig. 2b achieves 92.9%.

Lastly, Fig. 2c shows the results of our framework. It can be seen that our framework exhibits the advantage of GL where the fine-tuning accuracies are insensitive to data augmentations. Additionally, our framework exhibits the advantage of CL, *i.e.*, the latent representations have the discriminative power. Consequently, our framework is able to achieve the highest accuracy 94.0%. Note that the performance from best augmentation pairs is the most important measure in reality as this is the augmentation pair one would use for the final model.

The same experiments are also conducted on the other two datasets. Here we summarize key results from the heat maps in Table 1. We find that our framework consistently achieves the highest accuracy when compared to contrastive or generative learning. Furthermore, our method achieves significantly lower variance than the contrastive learning. In terms of median and mean accuracy, our method also outperforms the other counterparts, demonstrating the effectiveness of combining contrastive and generative methods.

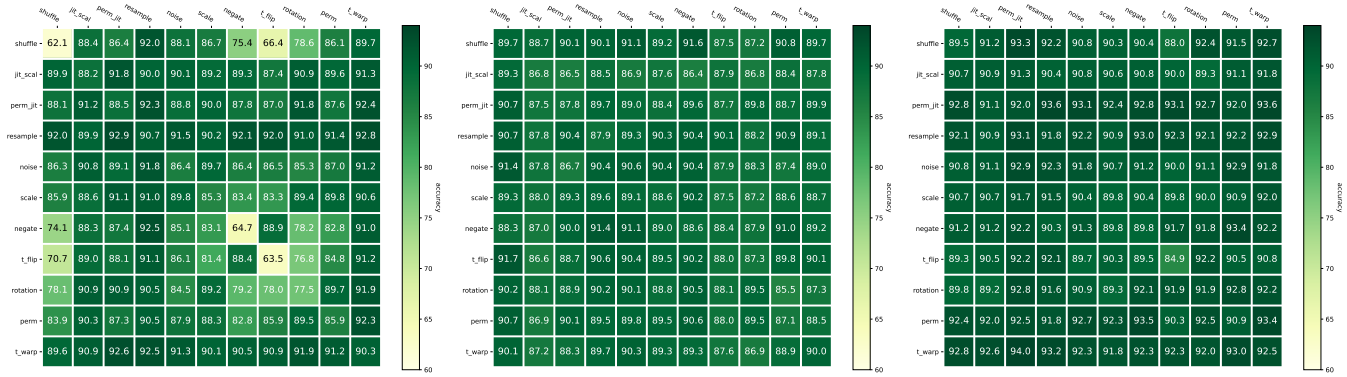
3.2. Comparison of Searched Augmentations

Here, we compare our automatic Data Augmentations (DA) search method with a common rigid DA search method. Note that the rigid DA search method runs every possible combination of the data augmentations, and it is limited to have only 2 data augmentations (one per branch) for each SSL task without adjusting their strengths. In contrast, our automatic DA search method can search a sequence of augmentations for each branch, as well as different strengths of the data augmentations. We search DA on each of UCI-HAR, SHAR, and HHAR datasets, and all experiments are executed under our proposed framework.

The results are summarized in Table 2. Generally, we observe that our automatic DA search method consistently outperforms the rigid DA search on all three datasets. We also visualize the input signal with searched augmentations in Fig. 3, it shows an example of input accelerometer signal (x,y,z axis) and its augmentations found by our method on SHAR dataset, the raw input signal is sequentially processed with t_warp, noise and shuffle augmentations. The details of each augmentation can be easily found in [9].

3.3. Parameter study on λ in Loss Function

In this section, we study different hyper-parameter values λ in the loss function as presented in Eq. (4). Parameter λ controls the weight of generative learning loss in the whole loss function. To have a comprehensive study of the impacts of contrastive and generative learning losses on HAR task accuracies, we also run two experiments



(a) Contrastive Learning (b) Generative Learning (c) Our Framework

Fig. 2: Fine-tuning accuracy on the SHAR [15] dataset with three different self-supervised learning methods.

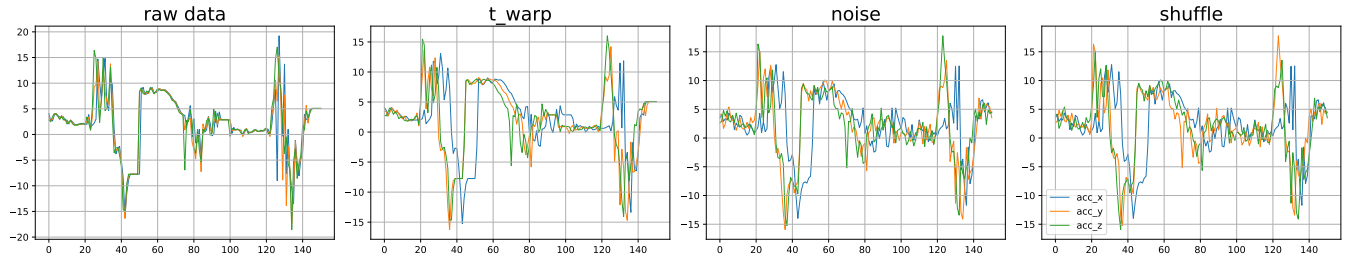


Fig. 3: Example of an input accelerometer signal and its searched augmentations.

Table 2: Comparison between our automatic DA search and the conventional rigid DA search method.

Method	SHAR	UCI-HAR	HHAR
Rigid (average)	91.5	93.9	92.4
Rigid (best)	94.0	97.0	94.0
Searched	**94.2**	**97.1**	**95.1**

when we only use contrastive learning loss (Contrastive loss only) or generative learning loss (Generative loss only) in the loss function. Note that these experiments are done with our proposed framework and augmentation search method.

As shown in Table 3, the proposed framework can achieve better accuracies when both contrastive and generative learning losses are adopted in the loss function. This demonstrates that our framework has the advantages of both contrastive and generative learning. It generates the latent embedding that enjoys the discriminative power of contrastive learning, as well as the reconstructing power provided by generative learning.

Additionally, as can be seen from Table 3, for SHAR and HHAR datasets, our framework achieves the best accuracy when λ is 1, which means contrastive and generative learning losses contribute similarly to the whole loss function. While for UCI-HAR dataset, we are able to deliver the best accuracy when λ equals to 100, in this case generative learning is more important in the loss function.

Overall, our optimized framework outperforms contrastive and generative learning on all three HAR datasets (contrastive and generative learning results presented in Table 1). And after tuning coefficient parameter λ in the loss function, we are able to perform better than supervised learning on SHAR and UCI-HAR datasets. This demonstrates the power of our proposed framework.

Table 3: Tune hyper-parameter λ to change the weight of generative learning loss in loss function.

Parameter λ	SHAR	UCI-HAR	HHAR
0 (Contrastive loss only)	93.7	97.3	93.1
0.01	93.6	96.8	93.2
1	**94.2**	97.1	95.1
100	93.7	**98.1**	93.4
Generative loss only	93.7	96.0	93.1
Supervised	92.3	97.7	**97.3**

4. CONCLUSIONS

In this paper, we proposed to incorporate the existing contrastive and generative learning framework to tackle the problem of unstable training regarding different data augmentations. Additionally, we proposed an automatic search method of data augmentations, which could search the optimal sequence of data augmentations with appropriate augmentation strength. Experiments on three HAR datasets were conducted to verify the effectiveness of our framework. Our method improved the robustness against different choices of data augmentations and pushed the limit of SSL on wearable data in HAR tasks. Furthermore, with the searched augmentation policy the accuracy of the proposed framework had been further improved.

5. REFERENCES

[1] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2021.

[2] Charissa Ann Ronao and Sung-Bae Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[3] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, pp. 2, 2020.

[4] Taoran Sheng and Manfred Huber, "Consistency based weakly self-supervised learning for human activity recognition with wearables," *AAAI-22 Workshop on Human-Centric Self-Supervised Learning*, 2022.

[5] Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur, "Collossl: Collaborative self-supervised learning for human activity recognition," *CoRR*, vol. abs/2202.00758, 2022.

[6] Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz, "Masked reconstruction based self-supervision for human activity recognition," in *Proceedings of the 2020 international symposium on wearable computers*, 2020, pp. 45–49.

[7] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien, "Multi-task self-supervised learning for human activity detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–30, 2019.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[9] Hangwei Qian, Tian Tian, and Chunyan Miao, "What makes good contrastive learning on small-scale wearable-based tasks?," *arXiv preprint arXiv:2202.05998*, 2022.

[10] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 857–876, 2023.

[11] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[12] Chenxi Liu, Piotr Dollár, Kaiming He, Ross Girshick, Alan Yuille, and Saining Xie, "Are labels necessary for neural architecture search?," in *European Conference on Computer Vision*. Springer, 2020, pp. 798–813.

[13] Colorado J Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer, "Selfaugment: Automatic augmentation policies for self-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2674–2683.

[14] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim, "Fast autoaugment," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[15] Daniela Micucci, Marco Mobilio, and Paolo Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, pp. 1101, 2017.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[17] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*. Springer, 2012, pp. 216–223.

[18] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen, "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition," in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, 2015, pp. 127–140.

[19] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization.," in *ICLR (Poster)*, Yoshua Bengio and Yann LeCun, Eds., 2015.