DI2SDiff++: Activity Style Decomposition and Diffusion-Based Fusion for Cross-Person Generalization in Activity Recognition

Junru Zhang, Cheng Peng, Zhidan Liu, Lang Feng, Yuhan Wu, Yabo Dong, Duanqing Xu

Abstract—Existing domain generalization (DG) methods for cross-person sensor-based activity recognition tasks often struggle to capture both intra- and inter-domain style diversity, leading to significant domain gaps with the target domain. In this study, we explore a novel perspective to tackle this problem, a process conceptualized as domain padding. This proposal aims to enrich the domain diversity by synthesizing intra- and inter-domain style data while maintaining robustness to class labels. We instantiate this concept using a conditional diffusion model and introduce a style-fused sampling strategy to enhance data generation diversity, termed Diversified Intra- and Inter-domain distributions via activity Style-fused Diffusion modeling (DI2SDiff). In contrast to traditional condition-guided sampling, our style-fused sampling strategy allows for the flexible use of one or more random style representations from the same class to guide data synthesis. This feature presents a notable advancement: it allows for the maximum utilization of possible combinations among existing styles to generate a broad spectrum of new style instances. We further extend DI2SDiff into DI2SDiff++ by enhancing the diversity of style guidance. Specifically, DI2SDiff++ integrates a multi-head style conditioner to provide multiple distinct, decomposed substyles and introduces a substyle-fused sampling strategy that allows cross-class substyle fusion for broader guidance. Empirical evaluations on a wide range of datasets demonstrate that our generated data achieves remarkable diversity within the domain space. Both intra- and inter-domain generated data have been proven significant and valuable, enabling DI2SDiff and DI2SDiff++ to surpass state-of-the-art DG methods in various cross-person activity recognition tasks.

Index Terms—Wearable sensor, human activity recognition, domain generalization, diffusion model

1 Introduction

Human activity recognition (HAR) is essential for a wide range of applications, including healthcare [1], assisted living [1], [2], and smart home systems [3]–[5]. By leveraging wearable devices and smartphones equipped with inertial measurement units (IMUs) to collect time-series data, HAR enables the accurate classification of various human activities, such as walking and sleeping. With the advancement of deep learning (DL) techniques [6]–[9] in time series classification (TSC) for HAR tasks, the deployment of trained models directly on edge devices has become increasingly feasible [10], [11]. However, a common assumption underpinning these models is that training and test data distributions are identically and independently distributed (i.i.d.) [12], a condition that does not often hold up in real life due to *individual differences in activity styles* influenced by factors such as age and gender [13], [14]. These discrepancies significantly hinder the cross-person generalization performance of standard DL models.

Domain generalization (DG) seeks to address this issue [12]. Approaches such as domain-invariant [13], [15]–[18] and domain-specific [19], [20] methods are designed to extract robust intra-domain and inter-domain features that can withstand data distribution shifts across various domains. However, their effectiveness is reliant on the diversity and breadth of the training data [21]. The challenge arises in HAR tasks, where the collected training data is often small-scale and lacks the necessary diversity due to resource constraints on edge devices [14], [22]. This inherent *scarce diversity in source domain training data* can lead to overfitting to local and narrow intra- or inter-domain features, resulting in poor generalization to new, unseen domains. As shown in Fig. 1(a) and (b), the learned features lack required intra- or inter-domain feature robustness, thereby impeding their generalization to target domains (red circles).

One promising solution is to enrich training distributions by data augmentation [12]. Recent research [14] has focused on enhancing training data richness through standard data augmentation like rotation and scaling; however, it primarily enhances *intra-domain diversity and falls short of addressing inter-domain variability*. As shown in Fig. 1(c), the augmented data (stars) for source domains (orange and blue circles) tends to cluster tightly, yet fails to generate the necessary inter-domain data. The target domain (red circles) thus cannot be comprehensively represented.

In this work, we focus on generating highly diverse data

Manuscript received xxx. This work was supported in part by the Zhejiang Science and Technology Plan Project (No.2025C02154), Key Scientific Research Base for Digital Conservation of Cave Temples (Zhejiang University), National Natural Science Foundations of China (No.62172284), the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No.2023B1212010007). (Corresponding author: Yabo Dong)

Junru Zhang, Cheng Peng, Yuhan Wu, Yabo Dong, and Duanqing Xu are with the College of Computer Science and Technology, Zhejiang University, Zhejiang 310027, China (e-mail: {junruzhang, chengchng, wuyuhan, dongyb, xdq}@zju.edu.cn).

Zhidan Liu is with the Intelligent Transportation Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangdong 511442, China (e-mail: zhidanliu@hkust-gz.edu.cn).

Lang Feng is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: lang.feng@ntu.edu.sg).

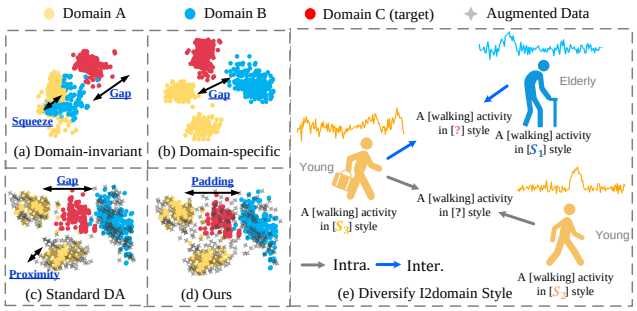


Fig. 1. T-SNE visualization of time-series activity features extracted by various methods across three domains in sensor-based HAR. Existing representation learning methods result in domain gaps as in both (a) and (b), covering a small portion of the target domain (red circles). Standard data augmentation (DA) leads to augmented data (stars), with source domains (orange/blue circles) remaining in close proximity to each other and failing to fill gaps. Our method (d) creates a comprehensive feature space by padding domain gaps via the idea of (e).

distributions to address the issue of limited domain diversity in sensor-based HAR. We explore a novel perspective to tackle this problem. As depicted in Fig. 1(d), the core idea involves enabling the synthetic data (stars) to fill the empty spaces within and across source domains while maintaining robustness to class labels, a process we conceptualize as "domain padding." For instance, as illustrated in Fig. 1(e), we decompose various walking styles from training data, and then we can not only combine the walking styles of an elderly man and a young man to create a novel inter-domain style but also merge multiple walking styles of different young men to generate a new intra-domain style. Compared to existing DG methods, our domain padding holds great potential to generate a more extensive range of unknown style distributions. This enables HAR models to comprehensively explore a wide array of intra- and inter-domain variations, contributing to enhanced generalization in HAR scenarios.

We propose **D**iversified **I**ntra- and **I**nter-domain distributions via activity **S**tyle-fused **Diff**usion modeling (DI2SDiff), to implement our concept using conditional diffusion probabilistic models [23], [24]. To generate samples with instance-level diversity, we first design a contrastive learning pipeline [25], aimed at extracting single-activity style representations from each available instance in the source domains while maintaining robustness for classification tasks. The extracted style representation, denoted as S_i, can be interpreted as "*an [activity class] performed in the [S_i] style*." We then propose a novel style-fused sampling strategy for the diffusion model to achieve domain padding requirements. This involves randomly combining one or multiple style representations of training samples within the same class. Styles in each combination are then utilized to jointly guide the diffusion to generate novel activity samples that fuse the styles. This innovation presents a notable advancement: the randomness of the combination (whether originating from the same or different domains) ensures diversity in both intra-domain and inter-domain, thereby achieving the domain padding, as shown in Fig. 1(d) and (e). Additionally, it maximizes the use of possible combinations among existing styles, generating a broad spectrum of new style instances.

Despite the promise of DI2SDiff in addressing cross-person activity recognition, its reliance on an entangled contrastive learning pipeline as a style conditioner introduces some limitations. Specifically, this conditioner is designed to capture a single-style representation that summarizes the overall characteristics of an activity instance. While effective for maintaining class-preserving guidance, this approach inherently lacks the diversity required to uncover complex guidance patterns in time-series activity data. The strict focus on a single style not only constrains DI2SDiff's capacity to capture nuanced and intricate variations but also exacerbates redundancy by enforcing intra-class style constraints, thereby overlooking a wealth of valuable style variants. This limitation is particularly pronounced in low-data classes, where the potential for exploring diverse patterns is critically hindered.

To address these challenges, we make several improvements to DI2SDiff. Firstly, we propose a novel multi-head style conditioner that extracts K distinct substyle features from each input instance. This is achieved through a hybrid training approach that integrates a supervised primary task with a self-supervised auxiliary task, enabling the conditioner to learn both meaningful patterns and diverse variations. By decomposing instances into multiple substyles, the conditioner provides a fine-grained representation, capturing both class-specific and class-agnostic features. Each instance is thus described as *"an [activity class] performed in the [$\{S_i^{(k)}\}_{k=1}^{K}$] style."* Secondly, thanks to these decomposed substyles, we introduce a substyle-fused sampling strategy that allows for precise and independent control of each substyle during synthesis. This approach enables flexible combinations of multiple class-specific substyles within the same class and the integration of diverse class-agnostic substyles across different classes. These two advancements culminate in a significantly improved model, termed DI2SDiff++, which demonstrates superior sample diversity and generation capabilities while overcoming the limitations of its predecessor.

Our main contributions to the field of generalizable HAR are as follows.

- We explore a pivotal challenge hampering the effectiveness of current DG methods in HAR: poor diversity of source domain features. In response, we propose that DI2SDiff implement the concept of "domain padding," enhancing domain diversity and improving the performance of DG models, and further advance this approach into DI2SDiff++ to effectively tackle more extreme scenarios with complex distribution shifts.
- We propose DI2SDiff, which uses activity style features as conditions to guide the diffusion process, extending the information available at the instance level beyond mere class labels. Meanwhile, DI2SDiff leverages a style-fused sampling strategy, which can flexibly fuse one or more style conditions from the same class to generate new, unseen samples. This strategy guarantees data synthesis diversity both within and across domains, enabling DI2SDiff to instantiate the concept of domain padding.

- To overcome the rigidity of single-style representation guidance, we further present DI2SDiff++ by introducing a multi-head style conditioner, which decomposes each instance into multiple substyles. This innovation captures multi-view semantic representations of activities, delivering a comprehensive and nuanced characterization of the data while preserving intricate pattern guidance.
- DI2SDiff++ introduces an advanced substyle-fused sampling strategy that dynamically blends intra-class substyles while seamlessly incorporating inter-class substyles. This method transcends traditional intra-class limitations, unlocking significant style diversity and enhancing the synthesis of high-quality, varied data.
- We conduct extensive empirical evaluations of DI2SDiff and DI2SDiff++ across a board of HAR tasks. Our findings reveal that these methods markedly diversify the intra- and inter-domain distribution without introducing class label noise. Leveraging these high-quality samples, DI2SDiff and DI2SDiff++ outperform existing solutions, achieving state-of-the-art results across various cross-person activity recognition tasks.

This paper extends our previous work [26] by introducing multi-substyle-guided generation techniques to enhance data diversity and improve time-series guidance in cross-person activity recognition. First, we propose a multi-head style conditioner that decomposes activity instances into multiple substyles, addressing the limitations of single-style representations and uncovering diverse style variations for more nuanced data characterization. Second, we introduce a substyle-fused sampling strategy that overcomes intra-class constraints, enabling flexible substyle fusion across the entire sample space to effectively enhance diversity. Finally, we conduct extensive evaluations, including visualizations, ablation studies, sensitivity analyses, and cross-architecture tests, showcasing the robustness and scalability of DI2SDiff++ as a new DG method for diversity and generalization in activity recognition.

2 Related Work

2.1 Sensor-Based Human Activity Recognition

Human Activity Recognition (HAR) using wearable sensors has achieved transformative progress across diverse applications, from healthcare to human-computer interaction [1], [9], [27]. Recent review literature [28]–[30] underscores the potential of deep learning to capture intricate activity patterns directly from raw IMU data, effectively bypassing the need for manual feature engineering. Techniques such as convolutional neural networks (CNN) [31], [32], recurrent neural networks (RNN) [33], and newer methods like generative adversarial networks (GAN) [34] and deep reinforcement learning (DRL) [35] have been applied to enhance both the accuracy and robustness of HAR. For example, CNN-SVM [31] leverages CNNs for spatial feature extraction, while [33] based on RNNs capture temporal dependencies in activity sequences. Additionally, hybrid models combining CNN and LSTM architectures are commonly utilized, harnessing CNN's spatial extraction capabilities and LSTM's strength in modeling temporal dynamics. For example, DeepConvLSTM [36] incorporates convolutional and LSTM units for multimodal wearable sensors. [37] proposed a hybrid model integrating CNN with bidirectional long short-term memory (BiLSTM), achieving impressive accuracy in HAR. [38] employed CNN for feature extraction, coupled with a reinforced selective attention model, to further refine performance. These advancements show great promise for algorithms deployed in real-world sensor systems. However, they often experience a significant performance drop when the distribution shifts due to changes in the subject, sensor device, or environment (i.e., domain). Such issues hinder the deployment of these advancements in real-world applications.

2.2 Domain Generalization

Domain Generalization (DG), a key technique in transfer learning [12], addresses distribution shifts by training models on multiple source domains to generalize to unseen target domains without requiring target data during training [39]. A common DG strategy is to learn domain-invariant features shared across domains [40], [41]. For example, SCA [42] uses multi-task autoencoders to emphasize shared features, but overlooks domain-specific traits that may be discriminative. To overcome this, recent methods [20], [43] disentangle domain-specific from invariant features. For instance, mDSDI [20] employs meta-learning to enhance adaptability to unseen domains.

To further enhance model robustness, augmentation-based approaches generate diverse inputs to improve DG performance. Domain randomization [44] is a widely adopted technique, often utilizing style transfer methods like AdaIN [45] to generate synthetic data by altering textures and other style attributes. For example, WildNet [46] leverages style transfer to alter low-level visual features, albeit limited by auxiliary domain dependence. These approaches rely on additional real data, which can be resource-intensive or impractical. Style augmentation [47]–[49] addresses additional data dependency issues by leveraging generative models (e.g., GANs or diffusion models) to vary image styles. For instance, StyleGAN-NADA [47] uses text prompts that describe diverse styles to train a GAN, enabling the synthesis of various images. Mixup [50] is another line of data generation technique, creating new samples by linear interpolation across domains to encourage generalization. Although DG has been extensively studied in the field of computer vision, its application to HAR tasks remains in its early stages. Furthermore, due to the unique characteristics of time-series data, most existing DG methods struggle to adapt to HAR signals, limiting their overall effectiveness.

2.3 Domain Generalization for Sensor-Based HAR.

DG can be extended to cross-person activity recognition by treating each user as a separate domain. CoDATS [51] proposes an adversarial approach aimed at learning domain-invariant features, which facilitates better generalization across different users. However, this method relies on the availability of labeled data in the target domain during

training, which may not always be practical in real-world applications. To address this, GILE [13] introduces an improved variational autoencoder (VAE) framework [52] that automatically disentangles domain-agnostic and domain-specific features. This approach allows for a more effective separation of generalizable and user-specific characteristics in HAR, yet it still requires domain labels. CrossHAR [53] and MobHAR [54] adopt two-stage pipelines consisting of an initial pre-training phase followed by fine-tuning. Although MobHAR [54] achieves improved performance by leveraging unlabeled target domain data during the fine-tuning stage, this strategy is less practical in real-world scenarios, where the goal is to deploy pre-trained models directly on data from new users without requiring any additional adaptation. DDLearn [14] and ContrastSense [55] both focus on enhancing feature diversity through different augmented views. However, they reliance on standard augmentation techniques primarily enriches intra-domain features, leaving it less effective at capturing complex inter-domain variations essential for robust cross-person activity recognition. DI2SDiff [26] represents one of the most recent DG methods for cross-person activity recognition, introducing a new diffusion-based generation framework for synthesizing diverse data. While promising, it still has certain limitations, which are detailed in Sec. 5 and effectively addressed in DI2SDiff++.

2.4 Diffusion Models

Diffusion models have shown strong capabilities in generating diverse, high-quality samples across domains such as computer vision [56]. Classifier-free guidance [57] further extends their effectiveness in multimodal tasks like text-to-image [58] and text-to-motion synthesis [59]. Given the non-stationary nature of time-series data [60], we explore the use of diffusion models to generate diverse samples for HAR, aiming to improve generalization. While diffusion models have seen success in time-series domains such as audio [61] and healthcare [62], their application to HAR remains underexplored [63]. Our work pioneers the use of diffusion models in time-series HAR, introducing a framework that guides the generation process toward diverse and representative samples. This contributes a novel solution to domain generalization in HAR and opens new directions for future research.

3 Preliminaries

3.1 Problem Statement

Following the definition of cross-person activity recognition [13], [14], we define the training dataset from source domains as: $D^s = \{(\mathbf{X}_i, y_i)\}_{i=1}^{n^s}$, where n^s is the number of instances. Each instance $\mathbf{X}_i \in \mathbb{R}^{D\times L}$ represents IMU data in the form of timestamped multivariate sequences collected from sensors, where D denotes the feature dimensionality and L represents the temporal length. The corresponding activity label $y_i \in \{1, \cdots, C\}$ indicates the class of the activity, with C denoting the total number of activity classes. The domain is characterized by a joint distribution $P(\mathbf{X}, y)$ across the time-series space $\mathcal{X}$ and activity label space $\mathcal{Y}$. Our goal is to learn a generalized model from D^s to predict the unseen target domain: $D^t = \{(\mathbf{X}_i, y_i)\}_{i=1}^{n^t}$, where $P^s(\mathbf{X}_i, y_i) \neq P^t(\mathbf{X}_i, y_i)$, $\mathcal{X}^s = \mathcal{X}^t$ and $\mathcal{Y}^s = \mathcal{Y}^t$. We aim to minimize the risk on D^t: $\min_f E_{(\mathbf{X}_i,y_i)\sim P^t}[f(\mathbf{X}_i) \neq y_i]$. Notably, to simulate real-world applications, the domain identifier is unavailable in our DG setting. Additionally, we consider a severe small-scale scenario where n^s is smaller than in typical DG setups.

3.2 Diffusion Probabilistic Model

The diffusion model [23] trains a distribution $p_\theta(x)$ to approximate the target distribution $q(x)$ using a Markov chain of Gaussian transitions:

$$p_\theta(x_0) = \int p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) dx_{1:T},$$

where $x_1, \ldots, x_T$ are latent variables with the same dimensionality as the original data x_0, and $p_\theta(x_T) \sim \mathcal{N}(0, \mathbf{I})$ is the Gaussian prior. The reverse process is defined as:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t)). \tag{1}$$

The forward process adds Gaussian noise to x_0 over T steps:

$$q(x_t|x_{t-1}) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}\right), \tag{2}$$

where $\beta_t \in (0, 1)$ controls the noise variance.

Training procedure. The diffusion model's parameters θ are optimized by maximizing the evidence lower bound of the log-likelihood $\log p_\theta(x_0)$, simplified to a surrogate loss [24]:

$$\mathcal{L}(\theta) := \mathbb{E}_{x_0,t\sim\mathcal{U},\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[||\epsilon - \epsilon_\theta(x_t, t)||^2\right], \tag{3}$$

where $\mathcal{U}$ is the uniform distribution and the noise predictor $\epsilon_\theta(x_t, t)$, parameterized with a deep neural network, aims to estimate the noise ϵ at time t given x_t. As $\mu_\theta(x_t, t)$ is determined by $\epsilon_\theta(x_t, t)$, the target $p_\theta(x_{t-1}|x_t)$ can be consequently derived.

Sampling procedure. Given a trained p_θ, data generation starts with Gaussian noise $x_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively denoises x_t for $t = T, \ldots, 1$ using $p_\theta(x_{t-1}|x_t)$, resulting in the generated data x_0.

4 Domain Padding

4.1 Intuitive Insight – Two Key Criteria

In HAR, the pivotal obstacle to domain generalization lies in the limited data diversity of the source domain, which hinders representation learning methods from conquering distribution shifts. Nevertheless, existing data augmentation methods [14] still expose the issue of insufficient data richness in the intra-domain and even inter-domain distribution. To this end, this work achieves effective data generation to enrich the diversity of training distributions. We propose a novel perspective termed "domain padding," whose core intention is to improve the coverage of the trainable domain space by "padding" the distributional gaps within and across source domains, as shown in Fig. 1(d). To ensure the generation of high-quality and diverse data, domain padding adheres to two key criteria:

- **First Criterion: Class-Preserved Generation**. The generated data should maintain class alignment with the original data, ensuring semantic consistency. This criterion ensures class consistency of generated data without compromising semantic stability.
- **Second Criterion: Intra- and Inter-Domain Diversity**. The generated data should not only boost the intra-domain diversity within an individual distribution but also enrich the inter-domain diversity across distinct distributions. This criterion guarantees a wide range of augmented variations for a more robust model training.

4.2 Feasible Solution – A Diffusion-based Framework

To meet the above two criteria, we implement domain padding into a conditional diffusion paradigm [24], [64], due to its excellent generative and flexible conditioning capabilities. Specifically, given an original instance $x \sim \mathcal{X}^s$, we can leverage its conditional information $s \in \mathcal{X}^{\text{cond}}$ to guide the generation of a new instance $\tilde{x}_0 \sim \tilde{\mathcal{X}}^s$. After the repeat conditional generation based on the dataset D^s, we collect a synthetic dataset $\tilde{D}^s = \{(\tilde{\mathbf{X}}_i, y_i^{\text{act}})\}_{i=1}^{\tilde{n}^s}$, where $\tilde{n}^s$ denotes the number of generated samples. The generation objective is to estimate the conditional data distribution $q(\tilde{x}|s)$ by using the specific constraint s to guide the synthesis of a new sample $\tilde{x}_0$. The conditional diffusion process can be formulated as:

$$q(\tilde{x}_t|\tilde{x}_{t-1}, s), \quad p_\theta(\tilde{x}_{t-1}|\tilde{x}_t, s). \tag{4}$$

Sequentially, performing p_θ enables the generation of new samples to capture the attributes of s.

Subsequently, we coordinate the conditional diffusion technique into a coherent framework, which encapsulates two key components. First, we devise a style conditioner to extract activity style features, which serve as conditions of a classifier-free guidance [57]. Thus, leveraging one explicit style condition effectively ensures that the generated samples meet the first criterion. The extracted instance-level styles also provide a guarantee for the construction of a diverse style-combination condition space $\mathcal{X}^{\text{cond}}$. Second, we introduce a style-fused sampling strategy to generate highly diverse intra- and inter-domain data, meeting the second criterion. In DI2SDiff++, the two components can further evolve into a multi-head style conditioner and a substyle-fused sampling strategy, respectively, which facilitate the nuanced pattern acquisition and diverse data generation from a cross-class aspect.

5 DI2SDiff Framework

To satisfy the first criterion, DI2SDiff employs a contrastive learning pipeline to extract robust, instance-level style representations, termed "*styles*," which fuse both distinctive instance characteristics and discriminative class semantics. Specifically, as shown in Fig. 2, a CNN-Transfomer-based model is used to encode each instance $\mathbf{X}_i$ into a style vector $S_i = f_{\text{style}}(\mathbf{X}_i) \in \mathbb{R}^H$, where H is the vector length. The vector S_i is learned by a contrastive learning approach [25]. The contrastive objective is to maximize the similarity between different augmented views of the same instance, while minimizing the similarity between different instances. This encourages the model to capture the instance's unique style while preserving class information of y_i. By aggregating style vectors from n^s instances, we form a set $\mathcal{S} = \{S_i\}_{i=1}^{n^s}$, which can be further partitioned into C class-specific subsets $\mathcal{S}^1, \mathcal{S}^2, \ldots, \mathcal{S}^C$. Then, DI2SDiff utilizes the style vector $s \in \mathcal{S}$ within the classifier-free guidance framework [57] to condition the sampling process, ensuring that each generated sample adheres to a specific style while preserving class semantics, thereby meeting the first criterion.

To address the second criterion, we propose a style-fused sampling strategy to guide the diffusion process conditioned on the style combination, i.e., multiple styles rather than a single style. Specifically, for each class c, we randomly combine one or more style features in the subset $\mathcal{S}^c$, obtaining a power set-like collection $\mathcal{P}^*(\mathcal{S}^c) = \mathcal{P}(\mathcal{S}^c) \setminus \{\emptyset\}$ with all $2^{|\mathcal{S}^c|} - 1$ possible style combinations. Repeatedly, we form a comprehensive set $\mathcal{D} = \bigcup_{c=1}^{C} \mathcal{P}^*(\mathcal{S}^c)$ across all C classes. The diffusion process is modified to condition on specific style combinations $\mathcal{D}_j \subseteq \mathcal{S}^c$. By applying this sampling strategy with various $\mathcal{D}_j$, the model generates new samples with diverse domain distributions, enabling the synthesis of novel domains and enhancing both intra- and inter-domain diversity. This generation process requires considering two hyperparameters κ and o. κ denotes the proportion of synthetic to original training samples, effectively controlling the expansion volume of the synthetic dataset. o denotes the maximum number of style features that can be integrated into each style combination $\mathcal{D}_j$.

Despite the promise of DI2SDiff on two domain-padding criteria, several crucial limitations are still exposed. (1) Using the single-style representation as the sampling condition will inevitably lead to a lack of diversity in generated data, thereby losing the ability to excavate complex guidance of time-series activity data. (2) The strict constraint, i.e. intra-class style combination, not only makes diffusion myopic and exacerbates the style redundancy within each class, but also natively overlooks a mass of potential valuable style variants, thereby limiting the exploration of data diversity, especially for low-data classes.

6 DI2SDiff++ Model

To address these limitations, Sec. 6.1 introduces a multi-head activity style conditioner that decomposes each time-series activity instance into multiple substyles, capturing richer semantics and nuanced variations. In Sec. 6.2, these substyles are then used to condition the generation process via classifier-free guidance framework [57], meeting the first criterion. Sec. 6.3 presents a substyle-fused sampling strategy that leverages the flexibility of the decomposed substyle space for diverse generations, overcoming previous intra-class fusion constraints and addressing both domain-padding criteria. The complete workflow of DI2SDiff++ is outlined in Sec. 6.4.

6.1 Multi-Head Activity Style Conditioner

To unlock diversity beyond single-style representations (the first limitation of DI2SDiff), we further decompose an ac-

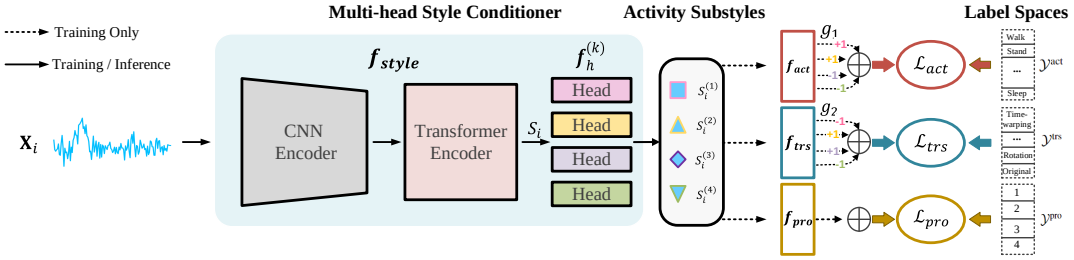


Fig. 2. The multi-head style conditioner of DI2SDiff++. This style conditioner integrates a combination of a CNN and a Transformer encoder (f_{style}), followed by four heads $f_{\text{h}}^{(k)}$. 1) During training, given an activity instance $\mathbf{X}_i$ (which can be original data or transformed data), four heads of the style conditioner decompose it into distinct substyle features (different shapes). These substyles are then processed through three fully connected layers (f_{act}, f_{trs}, f_{pro}) to execute separate multi-task learning objectives. Each task is optimized via its respective loss function: $\mathcal{L}_{\text{act}}$, $\mathcal{L}_{\text{trs}}$, and $\mathcal{L}_{\text{pro}}$. 2) During inference, only the style conditioner module is used to decompose the original training data into four distinct substyles. The first and second substyles, $S_i^{(1)}$ and $S_i^{(2)}$, capture class-specific semantics, while the third and fourth substyles, $S_i^{(3)}$ and $S_i^{(4)}$, represent class-agnostic features.

tivity instance into multiple representations, termed "substyles," to form a multi-view semantic expression of the sample. To ensure effectiveness and interpretability, we make these decoupled substyles complementarily capture information according to two different task spaces, i.e., the original activity class space $\mathcal{Y}^{\text{act}}$ with an attached transformation type space $\mathcal{Y}^{\text{trs}}$. Specifically, the former constrains each substyle to extract information about its relevant activity label, while the latter forces each substyle to focus on the augmented perturbation brought by the transformation. To obtain a reasonable transformation space, we apply five distinct time-series augmentations [25], [65], i.e., time-warping, permutation, jitter-and-scale, permutation-and-jitter, and rotation, on the original input data to introduce various real-world shifts. Thus, these transformed data construct a new data space $\mathcal{X}'$, which can be combined with the original data space to form an enriched input data space as $\mathcal{X}^{\text{in}} = \mathcal{X}^{s} \cup \mathcal{X}'$.

To further enhance the semantic diversity of features within these two task spaces, we decompose each into task-specific and task-agnostic components. This is supported by prior research [66], which shows that such factorization improves representation expressiveness. Consequently, we obtain *four* orthogonal semantic subspaces for substyle learning. To implement this, we propose a multi-head style conditioner within a multi-task learning framework, ensuring that the four resulting substyles capture explicit and complementary semantics. As illustrated in Fig. 2, our multi-head style conditioner is a CNN–Transformer-based module f_{style}, which follows a standard time-series backbone architecture [25]. Here, the CNN encoder is responsible for extracting local features, while the Transformer encoder captures long-range dependencies to summarize global contextual information. This enables the style conditioner to integrate both low-level and high-level feature representations. Once f_{style} generates a comprehensive style representation S_i for the input $\mathbf{X}_i \in \mathcal{X}^{\text{in}}$, we decouple S_i into $K(=4)$ distinct substyle components, $\{S_i^{(k)} = f_{\text{h}}^{(k)}(S_i) \in \mathbb{R}^H\}_{k=1}^K$, using K specialized projection heads. Then, to ensure that each projection head $f_{\text{h}}^{(k)}$ captures distinct semantics, we impose a specific multi-task learning objective $\mathcal{L}_{\text{mt}}^{(k)}$ on its corresponding mapped substyle $S_i^{(k)}$. Specifically, the objective $\mathcal{L}_{\text{mt}}^{(k)}$ incorporates three separate loss functions, enabling comprehensive learning for each head. Below, we provide a detailed breakdown of the three loss components.

Primary task - activity recognition. The primary task aims to identify the activity class for each instance. The activity label space, denoted as $\mathcal{Y}^{\text{act}}$, assigns each instance a label $y_i \in \{1, 2, \ldots, C\}$. The activity recognition loss for each projection head k is defined as:

$$\mathcal{L}_{\text{act}}^{(k)} = \mathbb{E}_{(\mathbf{X}_i \in \mathcal{X}^{\text{in}}, y_i \in \mathcal{Y}^{\text{act}})} \left[\ell(f_{\text{act}}(f_{\text{h}}^{(k)}(f_{\text{style}}(\mathbf{X}_i))), y_i) \right], \quad (5)$$

where ℓ denotes the cross-entropy loss function, and f_{act} is a fully-connected classifier to predict activity labels. This task primarily functions to constrain the substyles within the scope of activity semantics. Minimizing this loss enforces the alignment of substyles with activity semantics, thus fulfilling the first domain padding criterion, i.e., class-preserved generation.

Auxiliary task - transformation classification. To enrich fine-grained and intricate patterns, we introduce an auxiliary task that works in collaboration with the primary task. This auxiliary task is to identify the specific transformation applied to each input. The five transformations, along with the original format, form six distinct classes in the label space $\mathcal{Y}^{\text{trs}}$. Each instance is labeled as $y_i^{\text{trs}} \in \{1, 2, \ldots, 6\}$, representing the transformation types. The fully-connected classifier f_{trs} predicts the transformation type for each input instance. The corresponding loss for each projection head k is given by:

$$\mathcal{L}_{\text{trs}}^{(k)} = \mathbb{E}_{(\mathbf{X}_i \in \mathcal{X}^{\text{in}}, y_i^{\text{trs}} \in \mathcal{Y}^{\text{trs}})} \left[\ell(f_{\text{trs}}(f_{\text{h}}^{(k)}(f_{\text{style}}(\mathbf{X}_i))), y_i^{\text{trs}}) \right], \quad (6)$$

where ℓ represents the cross-entropy loss function. The optimization of $\mathcal{L}_{\text{trs}}^{(k)}$, in collaboration with the primary task of activity classification, enriches the style representation S_i.

Additional task - projection rectification. To ensure that each decoupled substyle $S_i^{(k)}$ is precisely mapped to its corresponding orthogonal subspace k, thereby allowing different substyles to remain independent and capture complementary semantics, we introduce an additional rectification loss for each projection head. The corresponding label space $\mathcal{Y}^{\text{pro}}$ is defined as the set of projection space indices, where each projection head k is assigned a label $y_i^{\text{pro}} \in \{1, 2, \ldots, K\}$, corresponding to the index of its respective subspace. The projection rectification loss for each head k is formulated as:

$$\mathcal{L}_{\text{pro}}^{(k)} = \mathbb{E}_{(\mathbf{X}_i \in \mathcal{X}^{\text{in}}, y_i^{\text{pro}} \in \mathcal{Y}^{\text{pro}})} \left[\ell(f_{\text{pro}}(f_{\text{h}}^{(k)}(f_{\text{style}}(\mathbf{X}_i))), y_i^{\text{pro}}) \right], \tag{7}$$

where f_{pro} is a fully-connected classifier that maps each output to its respective projection space index k, and ℓ is the cross-entropy loss function. This loss encourages each projection head to specialize in learning features aligned with its designated projection space, thus promoting the independence and diversity of the extracted components.

Overall loss. The overall learning objective of our multi-head style conditioner is formulated as:

$$\mathcal{L}_{\text{mt}} = \sum_{k=1}^{K} \mathcal{L}_{\text{mt}}^{(k)} = \sum_{k=1}^{K} (g_1^{(k)} \times \mathcal{L}_{\text{act}}^{(k)} + g_2^{(k)} \times \mathcal{L}_{\text{trs}}^{(k)} + \mathcal{L}_{\text{pro}}^{(k)}), \tag{8}$$

where $g_1^{(k)}$ and $g_2^{(k)}$ dynamically adjust the relative contributions of activity recognition and transformation classification losses, which promise task-specific flexibility during training. By systematically exploring all combinations of -1 and $+1$ for these coefficients[1], each projection head is optimized under a distinct multi-task objective. This design enables a more comprehensive and diverse depiction of $\mathbf{X}_i$ through the outputs of the projection heads, addressing the limitations of conventional single-style representations.

6.2 Decomposed Activity Substyles for Guidance

We now elaborate on the concept of decomposed substyles and their respective roles in generation guidance. In particular, the trained multi-head style conditioner decomposes each input sample $\mathbf{X}_i \in \mathcal{X}^s$ into 4 substyles: (1) class-specific, transformation-invariant features ($S_i^{(1)}$), which capture core class traits while ignoring transformations; (2) class-specific, transformation-aware features ($S_i^{(2)}$), which identify subtle intra-class variations and complex patterns introduced by transformations; (3) class-agnostic, transformation-aware features ($S_i^{(3)}$), which generalize transformation patterns across classes, aiding adaptation to distribution shifts; and (4) class-agnostic, transformation-invariant features ($S_i^{(4)}$), which provide stable, universal patterns across contexts. Consequently, each activity can be interpreted as *"a [y_i] activity performed in the [$\{S_i^{(k)}\}_{k=1}^K$] style,"* where y_i denotes the class of the original data.

This decomposition operation represents a significant step toward satisfying the first criterion of domain padding, as it preserves the class semantics through the class-specific substyles $S_i^{(1)}$ and $S_i^{(2)}$. The aggregation of these two substyle types from n^s training instances forms the sets $\mathcal{S}^{(1)} = \{S_i^{(1)}\}_{i=1}^{n^s}$ and $\mathcal{S}^{(2)} = \{S_i^{(2)}\}_{i=1}^{n^s}$, which can be further partitioned into C class-specific subsets, corresponding to the C activity classes. Each subset contains style vectors specific to a particular class, expressed as: $\mathcal{S}^{(1)} = \{\mathcal{S}^{(1),1} \cup \mathcal{S}^{(1),2} \cup \ldots \cup \mathcal{S}^{(1),C}\}$, $\mathcal{S}^{(2)} = \{\mathcal{S}^{(2),1} \cup \mathcal{S}^{(2),2} \cup \ldots \cup \mathcal{S}^{(2),C}\}$. In contrast, $\mathcal{S}^{(3)}$ and $\mathcal{S}^{(4)}$ capture class-agnostic features, providing a wider range of style variations not tied to specific activity classes. As such, these sets remain unpartitioned: $\mathcal{S}^{(3)} = \{S_i^{(3)}\}_{i=1}^{n^s}$, $\mathcal{S}^{(4)} = \{S_i^{(4)}\}_{i=1}^{n^s}$.

[1] $(g_1^{(k)}, g_2^{(k)}) = (+1,-1), (+1,+1), (-1,+1), (-1,-1)$ for $k \in \{1,2,3,4\}$

To control the generation of time-series activity samples, we leverage the style in $\mathcal{S}^{(k)}$ where $k \in \{1,2,3,4\}$ to guide the conditional sampling process $p_\theta(\tilde{x}_{t-1}|\tilde{x}_t, s)$ as presented in Eq. (4). To achieve this, we adopt classifier-free guidance [57], which has proven effective for generating data with specific characteristics. In this framework, the training process is adjusted to learn both a conditional $\epsilon_\theta(\tilde{x}_t, t, s)$ and an unconditional $\epsilon_\theta(\tilde{x}_t, t, \emptyset)$, where $\emptyset$ represents the absence of the condition s. The loss function is formulated as follows:

$$\mathcal{L}(\theta) := \sum_{k=1}^{K=4} \mathcal{L}^{(k)}(\theta), \tag{9}$$
$$\mathcal{L}^{(k)}(\theta) := \mathbb{E}_{x_0 \sim \mathcal{X}^s, \epsilon \sim \mathcal{N}(0,\mathbf{I}), t \sim \mathcal{U}, s \sim \mathcal{S}^{(k)}} \left[\|\epsilon - \epsilon_\theta(\tilde{x}_t, t, s)\|^2 \right].$$

Here, the condition s is a style feature in $\mathcal{S}^{(k)}$ where $k \in \{1,2,3,4\}$, derived from the pre-trained conditioner. This condition is randomly dropped during training to facilitate effective learning.

During the sampling phase, a sequence of samples $\tilde{x}_T, \ldots, \tilde{x}_0$ is generated starting from $\tilde{x}_T \sim \mathcal{N}(0, \mathbf{I})$. For each timestep t, the model refines the process of denoising $\tilde{x}_{t-1}$ based on $\tilde{x}_t$ through the following operation:

$$\hat{\epsilon}_\theta = \epsilon_\theta(\tilde{x}_t, t, \emptyset) + \omega\big(\epsilon_\theta(\tilde{x}_t, t, s) - \epsilon_\theta(\tilde{x}_t, t, \emptyset)\big), \tag{10}$$

where ω is a scalar hyperparameter controlling the alignment between the guidance signal and the sample [57], the iterative application of Eq. (10) enables the diffusion model to generate time-series activity data that precisely conform to specific styles $s \in \mathcal{S}^{(k)}$. This iterative refinement process empowers the diffusion model to synthesize activity samples that faithfully adhere to the prescribed substyles. By embedding class-specific semantic conditions within $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$, the generation process ensures the fulfillment of the first criterion of domain padding: each generated sample is associated with a known class under the guidance of a single condition. Meanwhile, the class-agnostic substyles $\mathcal{S}^{(3)}$ and $\mathcal{S}^{(4)}$ introduce nuanced yet powerful variations, enriching the guidance and dramatically enhancing the expressiveness and diversity of the generated data.

6.3 Beyond Class-Wise Activity Style Combination

So far, our approach has not fully met the second criterion of domain padding. Samples conditioned on a single style $s \in \mathcal{S}^{(k)}$ (where $k \in \{1,2,3,4\}$) exhibit limited variation within the intra-domain space, resulting in a narrow semantic scope. To address this limitation, we propose a substyle-fused sampling strategy to enhance diversity further. This strategy allows the diffusion process to generate data conditioned on any combination of substyles, rather than a single style. By blending diverse intra- and inter-domain styles, this method fulfills the second criterion of domain padding, enriching the generative process and enabling greater variation.

Random Substyle Combination. The random substyle combination method involves fusing four substyle features within a unified class to define a new diffusion sampling condition. By flexibly replacing substyle components, we move beyond the traditional constraints of intra-class sampling: class-specific substyles are selected from samples within the same class, while class-agnostic substyles are

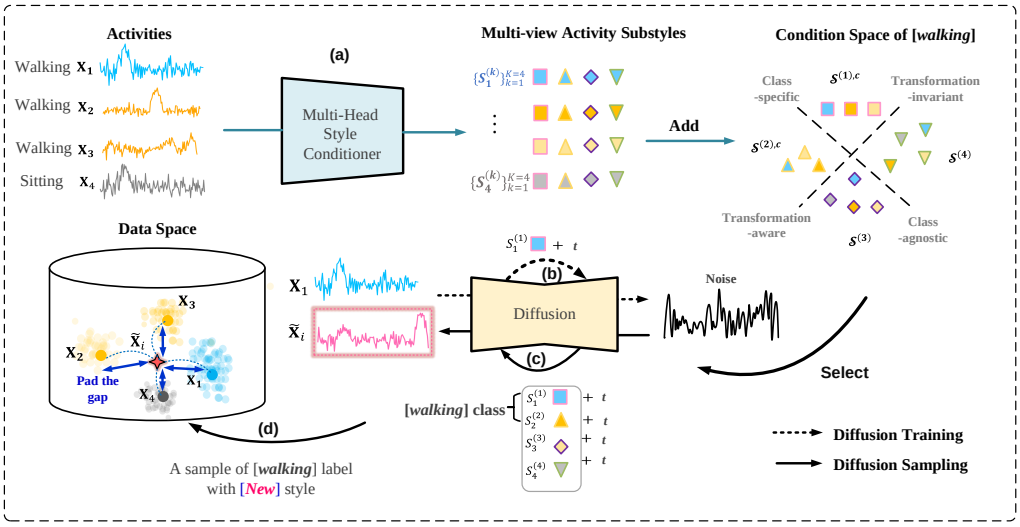


Fig. 3. Illustration of the diffusion within DI2SDiff++. Suppose we have three original walking instances $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$, and a sitting instance $\mathbf{X}_4$. Here, $\mathbf{X}_1$ and $\mathbf{X}_4$ are from different domains, while $\mathbf{X}_2$ and $\mathbf{X}_3$ belong to the same domain. (a) The multi-head style conditioner processes these samples to extract four substyle features, resulting in features such as $\{S_1^{(k)}\}_{k=1}^4$ for $\mathbf{X}_1$. The class-specific substyle features are grouped into $\mathcal{S}^{(1),c}$ and $\mathcal{S}^{(2),c}$, while the class-agnostic substyle features are stored in $\mathcal{S}^{(3)}$ and $\mathcal{S}^{(4)}$. (b) During training, the diffusion retrieves each data sample (e.g., $\mathbf{X}_1$) along with one substyle component (e.g., $S_1^{(1)}$) for the forward process. (c) To generate a walking instance $\tilde{\mathbf{X}}_i$, style components $S_1^{(1)}, S_2^{(2)}, S_3^{(3)}, S_4^{(4)}$ are selected, originating from the samples $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, and $\mathbf{X}_4$, respectively, to form a new substyle combination. (d) During sampling, the diffusion takes in noise and the substyle combination for the reverse process. (e) The generated sample $\tilde{\mathbf{X}}_i$ enriches the data space by combining four distinct substyles from different samples and domains.

drawn from across all classes. This thus expands the diversity exploration range, maximizing the utilization of the entire sample space.

Specifically, for each instance $\mathbf{X}_i$ labeled as class c with an original fused substyle combination $\{S_i^{(k)}\}_{k=1}^{K=4}$, a new substyle combination $\mathcal{C}_j$ is generated by randomly replacing one or more of the original substyle components with new components sampled from the corresponding substyle sets: $\mathcal{S}^{(1),c}$, $\mathcal{S}^{(2),c}$, $\mathcal{S}^{(3)}$, and $\mathcal{S}^{(4)}$. Formally, the set of all possible new substyle combinations $\mathcal{C}_j$ for a synthetic instance $\tilde{\mathbf{X}}_j$ with class label c is defined as:

$$\mathcal{C}_j \in \left\{ \left\{ S_j^{(1)}, S_j^{(2)}, S_j^{(3)}, S_j^{(4)} \right\} \mid \begin{array}{l} S_j^{(1)} \in \mathcal{S}^{(1),c}, \\ S_j^{(2)} \in \mathcal{S}^{(2),c}, \\ S_j^{(3)} \in \mathcal{S}^{(3)}, \\ S_j^{(4)} \in \mathcal{S}^{(4)} \end{array} \right\}. \quad (11)$$

This process is applied iteratively across all instances, with randomness in selecting substyles that may come from different samples and domains for combination. This enables the fusion of the four substyle features into novel, unseen forms within a unified class. The resulting fused combinations define various new diffusion sampling conditions, significantly expanding the generative condition space $\mathcal{X}^{\text{cond}}$.

Substyle-Fused Sampling. To empower the diffusion model to fuse substyles in $\mathcal{C}_j$ during the data generation, we perform sampling from the composed data distribution $q(\tilde{x}_0|\mathcal{C}_j)$ for any given substyle combination $\mathcal{C}_j$ is achieved through the following substyle-fused sampling strategy:

$$\hat{\epsilon}_\theta = \epsilon_\theta(\tilde{x}_t, t, \emptyset) + \omega \sum_{s \in \mathcal{C}_j} \Big(\epsilon_\theta(\tilde{x}_t, t, s) - \epsilon_\theta(\tilde{x}_t, t, \emptyset) \Big). \quad (12)$$

The derivation of Eq. (12) is detailed in the appendix. This equation extends the diffusion training process from single-style to multi-style combinations during sampling. For instance, as illustrated in Fig. 3(c), consider an original substyle combination $\{S_1^{(1)}, S_1^{(2)}, S_1^{(3)}, S_1^{(4)}\}$ from $\mathbf{X}_1$ labeled as class c. The first substyle component $S_1^{(1)}$ remains unchanged, preserving core class-specific traits. The second class-specific component is replaced with a sample from $\mathbf{X}_2$, which also belongs to class c. Meanwhile, the third and fourth class-agnostic components are substituted with samples from $\mathbf{X}_3$ and $\mathbf{X}_4$, which may originate from any class. This replacement produces a new substyle combination $\mathcal{C}_j = \{S_1^{(1)}, S_2^{(2)}, S_3^{(3)}, S_4^{(4)}\}$. Eq. (12) ensures that generated samples labeled as c exhibit unique characteristics derived from the fusion of the substyles in $\mathcal{C}_j$. This approach is crucial for achieving inter- and intra-domain diversity in domain padding, as it allows the diffusion model to flexibly incorporate instance-level substyles from either the same or different domains. Consequently, the diffusion model generates novel samples that exhibit a rich variety of previously unseen domain distributions, thereby satisfying the second criterion of domain padding: inter- and intra-domain diversity. Moreover, given the existence of sub-domains within each domain, our diffusion model is capable of synthesizing novel domains, even from sampling instances within the same domain (we verify this later in the experiments).

6.4 Workflow of DI2SDiff++

Finally, we present the comprehensive workflow of our methods. Both DI2SDiff and DI2SDiff++ include two processes: 1) generating new synthetic data via a diffusion

model guided by substyles, and 2) training a HAR classifier using both the synthetic and original data. During inference, only the trained HAR classifier is used.

6.4.1 Data Synthesis

Architectural Design. The diffsuion model $\epsilon_\theta : \tilde{\mathcal{X}}^s \times \mathbb{N} \times \mathcal{X}^{\text{cond}} \rightarrow \tilde{\mathcal{X}}^s$ is built upon a UNet architecture [57] with repeated convolutional residual blocks. To accommodate the characteristics of time series input, we adapt 2D convolution to 1D temporal convolution. The model incorporates a timestep embedding module and a condition embedding module, each of which is a multi-layer perceptron (MLP). The condition embedding module is used to encode each activity style $s \in \mathcal{S}$, and in the unconditional case $s = \emptyset$, we zero out the entries of s. These embeddings are then concatenated and fed into each block of the UNet.

Training. As shown in Fig. 3(a) and (b), the multi-head style conditioner in DI2SDiff++ extracts four distinct types of substyle features from each training instance. Each data instance $\mathbf{X}_i$, paired with its substyle component $S_i^{(k)} (k \in [1, \ldots, 4])$ and a randomly sampled timestep $t \sim \mathcal{U}$, forms a tripartite input $(\mathbf{X}_i, t, S_i^{(k)})$. This setup optimizes the model using a refined loss function defined in Eq. (9).

Sampling. During sampling, DI2SDiff++ constructs substyle combination sets $\mathcal{C}$ as described by Eq. (11). Each specific substyle combination $\mathcal{C}_j$ guides the diffusion process, creating a new sample that fuses diverse substyles, as depicted in Fig. 3(c).

Enhanced Domain Space Diversity. The iterative sampling procedure generates a broad range of new, unseen samples to form the synthetic dataset $\tilde{D}^s$ for expanding the data space, as shown in Fig. 3(d). DI2SDiff++ employs two hyperparameters: κ, which denotes the expansion proportion of synthetic samples, and $\zeta \in [1, 4]$, which specifies the number of replaced substyle components in new combinations. Notably, DI2SDiff++ eliminates the need for an extensive search to determine the optimal number of styles in a combination (i.e., the hyperparameter o in DI2SDiff), simplifying the process of achieving effective domain diversity. By using ζ to control the number of substyle replacements per instance, DI2SDiff++ ensures enhanced diversity in a fine-grained and controlled manner.

6.4.2 Training the HAR Classifier

Finally, we train a HAR classifier on the augmented dataset $\{\tilde{D}^s \cup D^s\} = \{(\mathbf{X}_i, y_i)\}_{i=1}^{n^s+\tilde{n}^s}$ and optimize it using a standard cross-entropy loss function to ensure accurate classification. The trained HAR model is then used to perform inference on the target data. By enhancing the expressiveness of the generated samples, DI2SDiff++ provides more effective benefits for HAR models, for example, achieving better generalization performance with fewer synthetic samples. We also provide detailed pseudocode for DI2SDiff++ in the appendix.

7 EXPERIMENTS

In this section, we conduct a comprehensive evaluation of DI2SDiff and DI2SDiff++ across various cross-person activity recognition tasks to demonstrate the following: **(1)** their capability to achieve domain padding, significantly diversifying the domain space; **(2)** their overall performance in domain generalization; **(3)** detailed ablation and sensitivity analyses; **(4)** a case study involving class-wise performance and efficiency analysis; **(5)** the computational complexity of our models; **(6)** a comparison of training time; **(7)** the feasibility of deployment on mobile devices; and **(8)** their extensibility in enhancing existing DG baselines and maintaining robustness across varying model backbones.

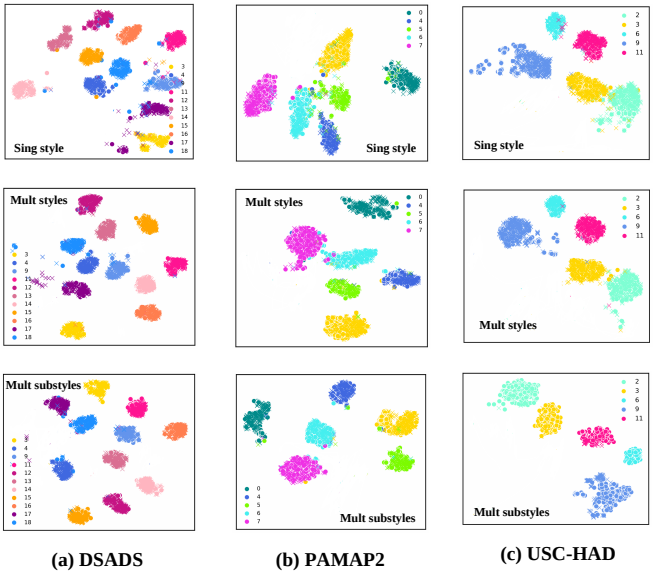
7.1 Experimental Setup

Datasets and domain split. We evaluate our method on three widely used wearable sensor-based HAR datasets: the UCI Daily and Sports Activities (DSADS) dataset [67], the PAMAP2 dataset [68], and the USC-HAD dataset [69]. We follow the same settings in [14] that provided a generalizable cross-person scenario. Specifically, the subjects are organized into separate groups for leave-one-out validation. We assign the data of one group as the target domain and utilize the remaining subjects' data as the source domain. Each subject is treated as an independent task.

Baselines. We compare our DI2SDiff [26] and DI2SDiff++ with a wide range of closely related, strong baselines adapted to sensor-based activity recognition tasks. We begin by including `Mixup` [70], `RSC` [71], `SimCLR` [72], `Fish` [73], and `DDLearn` [14], all of which have demonstrated strong performance in recent studies [14]. We further incorporate three more recently proposed cross-domain activity recognition approaches: `CrossHAR` [53], `ContrastSense` [55], and `MobHAR` [54]. We also include `TS-TCC` [25] for its remarkable generalization performance in self-supervised learning. Additionally, we incorporate `DANN` [74] and `mDSDI` [20], which are designed to address domain-invariant and domain-specific feature learning, respectively. In our analysis, the standard data augmentation (DA) techniques [65] are identical to those employed in [14], such as scaling and jittering.

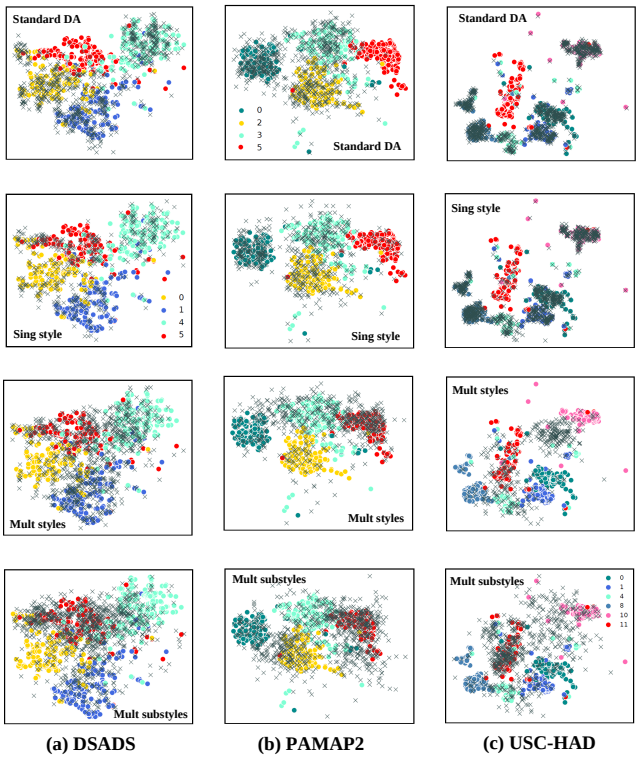
Architecture. For fairness, we adopt the same HAR classifier (except CrossHAR, MobHAR and TS-TCC) as described in [14], which consists of 2 convolutional blocks for DSADS and PAMAP2, and 3 convolutional blocks for USC-HAD. Each block includes a convolution layer, a pooling layer, and a batch normalization layer.

Implementation. For the diffusion model setting, both methods use the default diffusion configuration [24]: the forward process variances are set to constants increasing linearly from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$, with the number of diffusion steps $T = 100$ to ensure a fair comparison. In the generation setting, DI2SDiff adjusts the synthetic-to-original sample ratio κ between 1 and 5 to ensure effective performance. The parameter o, which defines the maximum number of style features used per combination, is set between 5 and 10. DI2SDiff++ sets $\kappa = 1$ and the number of replaced substyle components in each new combination to $\zeta = 2$ by default. In each experiment, we report the average performance and standard deviation over three random seeds. Additional experimental details, including information about datasets, architecture, and training settings, are provided in the appendix.



(a) DSADS (b) PAMAP2 (c) USC-HAD

Fig. 4. T-SNE visualization of DSADS, PAMAP2, USC-HAD datasets. The original and synthetic data are represented by shapes dots and crosses, and each class is denoted by a color. Best viewed in color and zoom in.



(a) DSADS (b) PAMAP2 (c) USC-HAD

Fig. 5. T-SNE visualization of DSADS, PAMAP2, and USC-HAR datasets. Each domain category is represented by a color, and the target domain is represented by a red dot. The original and synthetic data are represented by shapes dots and crosses, respectively. Best viewed in color and zoom in.

7.2 Domain Padding and Diversity Evaluation

In this part, we demonstrate whether our DI2SDiff and DI2SDiff++ can effectively diversify the domain space and generate diverse samples that meet domain padding criteria. To this end, we adopt T-SNE [75] to visualize the latent feature space in terms of class and domain dimensions.

(1) Class-Preserved Generation. First, we evaluate the class consistency of synthetic data, i.e., the first criterion of domain padding. We employ a class feature extractor, trained with class labels, to map both original and synthetic data into a class-specific space. The results of single-style guidance ($|\mathcal{D}_j| = 1$), multiple-style guidance ($|\mathcal{D}_j| > 1$) and multiple-substyle guideance ($(|\mathcal{C}_j| = 4$) are shown in Fig. 4.

It can be observed that all synthetic samples (crosses) are closely clustered around their corresponding original instances and classes (dots). This clustering indicates that our method effectively maintains class information, avoiding the introduction of class noise; importantly, this holds true under single-style, multiple-style, and multiple-substyle guidance. Moreover, the use of multiple-substyle guidances appears to enhance class discriminability more than both single- and multiple-style guidance in Fig. 4. This improvement is likely due to the introduction of explicit class-specific guidance signals, which provide more robust and precise class semantics, leading to better class alignment in the generated samples.

(2) Intra- and Inter-Domain Diversity. We evaluate the intra- and inter-domain diversity of the synthetic data, i.e., the second criterion of domain padding. We train the domain feature extractor on source domains with domain labels. We then map source and target data into a domain-specific latent space and compare the synthetic data from the standard DA method, single-style guidance ($|\mathcal{D}_j| = 1$), multiple-style guidance ($|\mathcal{D}_j| > 1$) and multiple-substyle guidance ($(|\mathcal{C}_j| = 4$). The results are shown in Fig. 5.

The findings reveal that the standard DA method generates tightly clustered samples (crosses) around the original data (dots), falling short of diversifying the domain space, particularly the inter-domain space. Our single-style guidance method offers a partial solution and generates sparse data between different domains thanks to the diffusion's probabilistic nature. However, relying on a single-style guidance approach has limitations for domain padding. The introduction of our style combinations and substyle combination makes a substantial improvement: the multiple-condition guidance excels in "padding" the distributional gaps both within and across source domains, as shown in Fig. 5. By capitalizing on the distinct characteristics of individual styles within the original data, the synthetic samples (crosses) more closely align with the target domain data (red dots) while exhibiting reduced dependence on the specific traits of the source domains. This underscores the ability of multi-style and multi-substyle fusion, supported by our sampling strategy, to generate a broader and more diverse range of styles, which is essential for achieving robust domain generalization in HAR.

We also observe in Fig. 5(c) that the USC-HAD dataset presents an additional challenge of intra-domain gaps due to its fragmented and sparsely distributed source domains with distinct sub-domains. These gaps contribute to an increased distribution shift, posing difficulties for existing DG methods to perform effectively (We show their results in Tab. 1 later). By random instance-level style/substyle fusion, our DI2SDiff and DI2SDiff++ effectively address this

TABLE 1
Classification accuracy (%) (± standard deviation) on three public datasets, where each task only comprises 20% of training data. The second-best results are underlined, and the best results are in bold. "T0-T4" represent different cross-person activity recognition tasks.

	Tar.	Mixup	RSC	SimCLR	Fish	DANN	mDSDI	TS-TCC	DDLearn	CrossHAR	ContrastSense	MobHAR	DI2SDiff	DI2SDiff++
DSADS	T0	74.77 (±1.76)	54.32 (±2.19)	72.48 (±3.18)	55.06 (±1.60)	72.49 (±3.21)	76.91 (±2.34)	81.47 (±0.53)	87.88 (±1.92)	82.41 (±0.73)	80.07 (±0.81)	81.38 (±0.45)	89.93 (±2.57)	**90.17** (±1.22)
	T1	75.78 (±3.95)	63.62 (±10.56)	76.61 (±2.56)	62.28 (±3.13)	69.61 (±1.96)	76.02 (±1.56)	79.68 (±0.42)	88.80 (±1.11)	82.60 (±1.25)	81.90 (±0.58)	82.17 (±0.95)	90.17 (±0.84)	**91.25** (±0.54)
	T2	74.18 (±4.36)	66.48 (±1.80)	78.25 (±0.92)	68.15 (±1.60)	78.97 (±4.06)	72.71 (±0.98)	84.37 (±1.87)	89.21 (±1.23)	80.77 (±2.45)	82.58 (±2.33)	82.82 (±1.78)	91.39 (±1.31)	**91.82** (±1.22)
	T3	75.85 (±3.45)	64.29 (±3.37)	76.49 (±0.91)	68.83 (±3.83)	78.54 (±2.14)	79.58 (±1.29)	82.09 (±2.51)	85.63 (±1.13)	80.66 (±2.33)	78.23 (±2.56)	80.33 (±1.56)	88.95 (±1.79)	**89.95** (±0.24)
	Avg	75.15 (±2.36)	62.18 (±4.32)	75.96 (±1.25)	63.58 (±0.37)	74.90 (±2.63)	76.31 (±1.56)	81.65 (±1.33)	87.88 (±0.82)	81.61 (±1.69)	80.70 (±1.57)	81.68 (±1.19)	90.11 (±1.63)	**90.80** (±1.55)
PAMAP2	T0	57.81 (±0.55)	55.99 (±1.29)	63.28 (±3.33)	54.04 (±4.31)	54.02 (±3.52)	58.70 (±3.14)	64.08 (±1.98)	75.55 (±0.79)	70.36 (±2.18)	65.45 (±0.68)	63.82 (±0.25)	79.58 (±2.46)	**81.51** (±1.25)
	T1	81.51 (±3.94)	83.08 (±2.42)	81.25 (±1.59)	85.16 (±1.39)	77.21 (±3.79)	83.82 (±1.62)	86.55 (±2.28)	90.07 (±2.40)	90.15 (±1.22)	91.91 (±1.48)	92.28 (±2.05)	94.12 (±1.20)	**95.35** (±1.54)
	T2	77.34 (±3.33)	78.65 (±3.99)	78.65 (±1.87)	79.69 (±4.00)	78.80 (±1.87)	79.15 (±2.72)	80.21 (±0.52)	85.51 (±0.76)	86.22 (±0.88)	81.27 (±0.23)	82.63 (±1.75)	89.57 (±2.48)	**91.51** (±1.25)
	T3	70.31 (±5.64)	68.10 (±6.27)	71.09 (±1.99)	72.53 (±0.49)	61.96 (±2.11)	78.61 (±0.49)	77.32 (±0.47)	80.67 (±1.78)	79.35 (±0.78)	78.81 (±0.79)	81.88 (±1.22)	84.75 (±3.72)	**85.80** (±2.22)
	Avg	71.74 (±1.37)	71.45 (±2.55)	73.57 (±1.21)	72.85 (±0.37)	68.00 (±2.66)	75.07 (±1.99)	77.04 (±1.29)	82.95 (±0.60)	82.02 (±1.27)	79.36 (±0.80)	80.15 (±1.32)	87.01 (±1.94)	**88.54** (±1.53)
USC-HAD	T0	68.66 (±4.67)	75.69 (±4.28)	69.36 (±2.34)	73.70 (±3.97)	57.79 (±4.73)	59.71 (±1.23)	78.96 (±0.79)	79.06 (±2.11)	75.86 (±1.23)	72.15 (±0.88)	77.11 (±1.77)	88.33 (±1.70)	**89.93** (±1.05)
	T1	68.75 (±1.29)	72.40 (±2.88)	66.62 (±1.44)	72.05 (±2.93)	64.95 (±2.68)	67.35 (±2.46)	79.55 (±1.23)	80.15 (±1.11)	77.94 (±1.56)	78.93 (±1.25)	79.46 (±2.18)	81.64 (±0.28)	**84.52** (±0.12)
	T2	71.79 (±0.65)	72.83 (±3.62)	76.04 (±1.61)	69.10 (±2.93)	71.97 (±3.23)	63.89 (±3.69)	78.15 (±2.15)	80.81 (±0.74)	82.83 (±0.58)	80.74 (±0.14)	78.28 (±0.79)	88.37 (±1.46)	**89.75** (±0.23)
	T3	61.29 (±3.90)	63.19 (±5.30)	61.24 (±1.06)	58.51 (±3.66)	45.65 (±2.18)	63.87 (±4.92)	64.35 (±1.58)	70.93 (±1.87)	70.53 (±0.74)	72.77 (±1.69)	65.63 (±1.44)	77.84 (±1.10)	**80.05** (±1.21)
	T4	65.63 (±4.55)	66.75 (±3.25)	62.85 (±2.17)	63.72 (±8.31)	54.94 (±3.56)	55.95 (±6.15)	70.25 (±0.88)	75.87 (±2.99)	70.13 (±3.45)	73.05 (±4.66)	68.76 (±0.68)	83.84 (±0.88)	**86.77** (±0.84)
	Avg	67.22 (±2.41)	70.17 (±3.51)	67.22 (±0.39)	67.42 (±3.91)	59.06 (±2.65)	62.15 (±3.08)	74.25 (±1.16)	77.36 (±0.99)	75.46 (±1.51)	75.53 (±1.51)	73.85 (±1.37)	84.00 (±1.09)	**86.20** (±1.22)
Avg All		71.37	67.93	72.25	67.95	67.32	71.18	77.65	82.73	79.70	78.53	78.56	87.04	**88.51**

TABLE 2
Classification accuracy (%) on three public datasets with varying percentages (%) of used training data. The second-best results are underlined. The best results are in bold.

	Perct.	Mixup	RSC	SimCLR	Fish	DANN	mDSDI	TS-TCC	DDLearn	CrossHAR	ContrastSense	MobHAR	DI2SDiff	DI2SDiff++
DSADS	20%	75.15	62.18	75.96	63.58	74.90	76.31	81.65	87.88	81.61	80.70	81.68	90.11	**90.80**
	40%	82.48	67.70	75.76	65.82	75.45	76.55	82.54	89.71	82.42	81.21	82.45	91.25	**91.88**
	60%	82.70	69.98	75.61	67.65	76.55	77.89	83.78	90.43	82.95	81.45	82.54	92.56	**92.94**
	80%	81.58	75.37	74.69	66.03	76.89	79.25	84.12	90.97	82.12	84.52	85.66	94.58	**94.97**
	100%	83.44	75.58	76.22	69.35	80.52	79.58	86.57	91.95	87.61	86.60	88.76	95.23	**96.86**
PAMAP2	20%	71.74	71.45	73.57	72.85	68.00	75.07	77.04	82.95	82.02	79.36	80.15	87.01	**88.54**
	40%	76.69	73.73	74.25	77.02	69.85	72.55	78.35	84.34	82.55	79.55	81.45	87.66	**88.22**
	60%	77.83	75.72	74.71	76.04	70.88	76.56	80.15	85.03	81.54	80.54	85.12	88.75	**89.54**
	80%	78.00	76.17	74.09	75.13	77.82	77.53	81.78	86.67	82.65	81.15	86.97	89.92	**90.41**
	100%	79.72	77.96	74.25	75.49	79.56	78.83	83.45	86.31	83.63	81.54	86.27	90.96	**92.25**
USC-HAD	20%	67.22	70.17	67.22	67.42	59.06	62.15	74.25	77.36	75.46	75.53	73.85	84.00	**86.20**
	40%	75.30	77.31	69.16	73.54	61.52	68.85	75.32	80.72	75.81	75.65	74.52	84.97	**85.72**
	60%	78.14	77.59	71.38	76.09	68.71	76.75	77.84	80.88	76.45	76.45	76.84	87.53	**89.53**
	80%	79.76	78.65	71.99	77.21	68.52	77.72	78.91	82.49	78.84	78.88	80.12	89.25	**90.79**
	100%	81.27	79.41	72.14	78.92	72.05	78.59	79.15	82.51	81.22	80.62	83.43	91.13	**92.95**

sub-domain challenge, enabling the synthesis of new data distribution within sub-domains. As a result, our methods can yield exceptional performance on complex tasks like USC-HAD.

More importantly, the multi-substyle guidance within DI2SDiff++ proves highly effective in simulating complex target data distributions and addressing substantial distributional shifts. For example, in the USC-HAD dataset, the target data shows minimal overlap with the source domain, representing an extreme case of distributional divergence. As shown in Fig. 5(c), our multi-substyle fusion aligns well with the true target distribution, enabling the synthesized samples to effectively cover the target domains. This validates the ability of DI2SDiff++ to unlock meaningful distributional diversity, significantly improving the model's generalization across complex domains, as Tab. 1 shows.

7.3 Generalization Performance

Now we conduct a series of experiments to evaluate the generalization performance of DI2SDiff and DI2SDiff++ against other strong DG baselines.

Overall performance. Tab. 1 presents a comparative analysis of the classification accuracies achieved by all DG methods across three datasets, each task of which comprises 20% of the training data. As we can see, representation learning baselines that focus solely on learning domain-invariant features, such as DANN [74], exhibit suboptimal performance due to the limited diversity of the training data in HAR. The method mDSDI [20], on the other hand, achieves improved performance by additionally learning domain-specific features. However, it does not match the performance of ContrastSense [55] and DDLearn [14], which utilize data augmentation, underscoring the importance of training data diversity in enhancing generalization in HAR. Furthermore, CrossHAR [53] and MobHAR [54] introduce external fine-tuning procedures, which contribute to performance gains. In contrast, DI2SDiff and DI2SDiff++ consistently outperform all baselines by directly synthesizing diverse intra- and inter-domain activity data, achieving superior generalization without requiring any additional adaptation. Fig. 6 (a–c) further illustrates the F1-scores of representative DG methods across the three datasets. Consistent with the accuracy results in Tab. 1, DI2SDiff and DI2SDiff++ achieve the highest F1-scores across all tasks, affirming their capacity to generate meaningful activity data that encapsulates domain variability.

In addition, we observe that all baselines, including DDLearn, demonstrate poor performance on the USC-HAD dataset. As illustrated in Fig. 5(c), the observed decline is attributed to the presence of sub-domains within the source domain, posing a significant challenge for DG. DI2SDiff and DI2SDiff++ effectively address this issue by leveraging instance-level style fusion to synthesize new data distributions that bridge these sub-domains. Consequently, DI2SDiff and DI2SDiff++ achieve remarkable performance, surpassing the third-best method (DDLearn) by significant margins of 6.64% and 8.84%, respectively, on the USC-HAD dataset. The superior performance of DI2SDiff++ over DI2SDiff is particularly evident in its ability to tackle the pronounced distributional shifts between the source and target domains, as illustrated in Fig. 5(c). By integrating multi-view, fine-grained guidance (such as transformation-aware substyles) into the synthetic data generation process, DI2SDiff++ achieves a marked enhancement in distributional diversity. This methodology allows the model to encapsulate and represent the intricate nuances of the target domain more effectively. As a result, DI2SDiff++ further advances the robust second-best method (DI2SDiff), delivering even greater improvements across all cross-person tasks, particularly on the USC-HAD dataset.

Data proportion analysis. In Tab. 2 and Fig. 6 (d-f), we evaluate the average accuracy and F1-score of DI2SDiff and DI2SDiff++ across varying proportions of training data, ranging from 20% to 100%. The results consistently demonstrate the superiority of our DI2SDiff and DI2SDiff++ over baseline approaches, regardless of the amount of available data. This underscores the efficiency of our approach in gen-

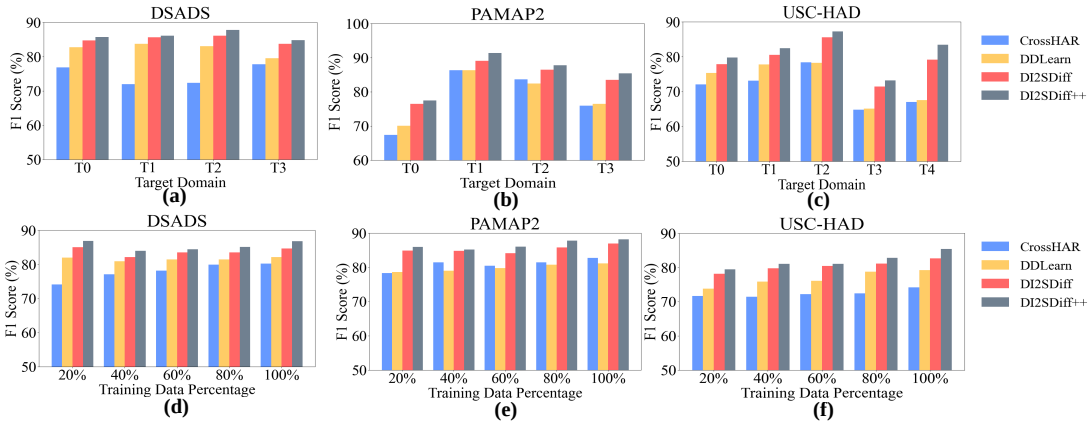


Fig. 6. F1 score (%) comparison of four competitive methods (CrossHAR, DDLearn, DI2SDiff, and DI2SDiff++) on three datasets. Subfigures (a–c) show the results with 20% training data across different target domains; subfigures (d–f) show performance under varying training data percentages.

TABLE 3
The results of the ablation study on three datasets and each task is averaged for an overall assessment.

Line No.	Variants	DSADS		PAMAP2		USC-HAD	
		20%	100%	20%	100%	20%	100%
1	Standard DA	75.58	82.57	70.31	86.41	69.14	75.45
2	Class Label Guidance	76.25	84.67	72.78	88.52	70.85	76.26
3	Single Style Sampling	86.98	91.12	83.07	89.64	75.27	83.57
4	Substyle Sampling	87.15	92.13	84.25	89.75	78.78	84.54
5	Style-Fused Sampling	90.11	95.23	87.01	90.96	84.00	91.13
6	Substyle-Fused Sampling	**90.80**	**96.86**	**88.54**	**92.25**	**86.20**	**92.95**
7	Substyle-Fused Sampling w/o $\mathcal{S}^{(1)}$	83.42	85.12	79.25	84.27	76.25	82.78
8	Substyle-Fused Sampling w/o $\mathcal{S}^{(2)}$	87.52	89.56	83.25	87.17	81.17	86.63
9	Substyle-Fused Sampling w/o $\mathcal{S}^{(3)}$	88.41	93.35	85.21	89.54	81.40	87.23
10	Substyle-Fused Sampling w/o $\mathcal{S}^{(4)}$	89.11	94.03	86.45	90.11	82.54	89.58
11	Substyle-Fused Sampling	**90.80**	**96.86**	**88.54**	**92.25**	**86.20**	**92.95**
12	DI2SDiff++ w/o CNN	84.54	86.45	80.41	80.69	75.12	83.78
13	DI2SDiff++ w/o Transformer	79.54	82.77	78.66	78.29	73.45	81.78
14	DI2SDiff++	**90.80**	**96.86**	**88.54**	**92.25**	**86.20**	**92.95**

erating informative synthetic samples and effectively leveraging them for learning. Meanwhile, ignoring fine-grained style decomposition and broad interactions across styles limits the fidelity and diversity of the generated data, constraining DI2SDiff's performance. By addressing these limitations, DI2SDiff++ achieves enhanced performance across all cross-person activity recognition tasks. Interestingly, as the size of the training sample increases, the advantage of DI2SDiff++ becomes even more pronounced. For example, on the USC-HAD dataset, increasing the training data from 20% to 100% leads to an increase in accuracy improvement from 8.84% to 10.44%, and in F1-score improvement from 5.62% to 6.22%, compared to the third-best baseline (DDLearn). This is attributable to the larger number of possible substyle combinations generated when more training data are available. Thus, enlarging the training dataset not only enhances the diversity of synthesized data but also significantly boosts the model's generalization ability, resulting in more robust performance across a wide range of training volumes.

7.4 Ablation and Sensitivity Analysis

In this section, we perform an ablation study that focuses on the main step of DI2SDiff and DI2SDiff++, i.e., generating diverse time-series activity data via diffusion for data augmentation. We keep the number of synthetic samples and the training strategy of HAR models the same for all variants. Additionally, we perform a sensitivity analysis to examine the impact of critical hyperparameters in both DI2SDiff and DI2SDiff++. First, for DI2SDiff, the focus is on two hyperparameters: o, which defines the maximum number of style features in each style combination, and κ, which regulates the volume of synthetic data. Second, for DI2SDiff++, we analyze the impact of ζ, which specifies the number of replaced substyle features in new combinations, and κ, which similarly controls the volume of synthetic data.

Ablation on diffusion model. Our findings, as outlined in Tab. 3, underscore the limitations of standard data augmentation (DA) (Line 1) and class label guidance (Line 2). The latter, which directly uses class labels as diffusion conditions without leveraging style features, fails to capture instance-specific nuances, resulting in underwhelming performance. This highlights the inadequacy of class labels alone in generating high-quality, diverse data, as they lack the granularity required for robust generalization. In contrast, conditioning the diffusion model on single-style features (Line 3) or substyle features without replaceme nt (Line 4) yields noticeable performance improvements, demonstrating that instance-level representation features are pivotal for enhancing the fidelity and utility of synthesized data. This reinforces the importance of utilizing more granular style representations over static class labels. The introduction of random style and substyle combinations further elevates performance, as evidenced by the results for style-fused sampling in DI2SDiff (Line 5) and substyle-fused sampling in DI2SDiff++ (Line 6). These strategies enable the generation of diverse and novel feature combinations, with substyle-fused sampling consistently outperforming its style-fused counterpart. This finding underscores the critical advantage of incorporating finer-grained, multi-view representations to achieve superior generalization.

To further investigate the role of substyles, we examined the impact of removing various substyle components from DI2SDiff++ (Lines 7–11). The results reveal that excluding any substyle leads to a significant decline in performance, underscoring the integral role of each substyle in facilitating robust generalization. Notably, class-specific substyles ($\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$) are particularly influential, as they encode

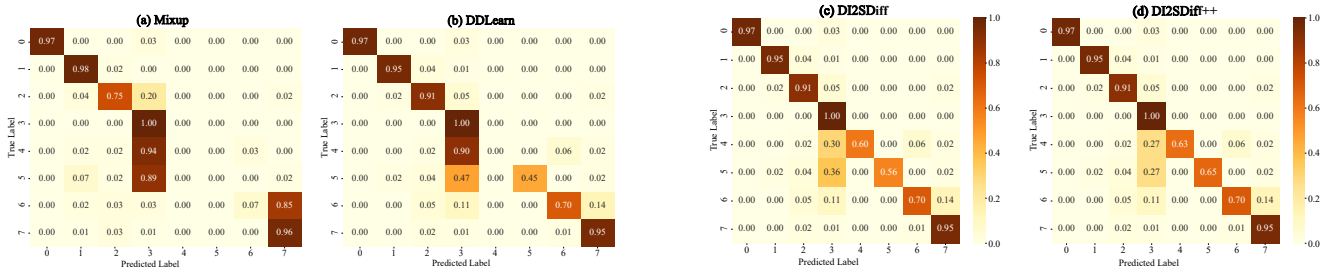


Fig. 7. The confusion matrices for the first task of the PAMAP2 dataset with 20% training data. Labels 0–7 denote the activities: lying, sitting, standing, walking, ascending stairs, descending stairs, vacuum cleaning, and ironing.

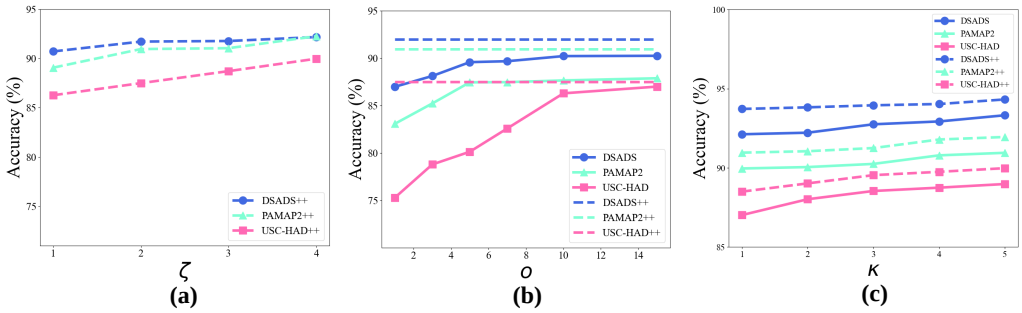


Fig. 8. Hyperparameter sensitivity analysis for DI2SDiff (solid line) and DI2SDiff++ (dashed line): (a) ζ (b) o (c) κ.

the semantic essence of each activity class, ensuring clear differentiation between classes. Moreover, transformation-aware substyles (e.g., $\mathcal{S}^{(3)}$) exhibit a pronounced influence, especially on datasets like USC-HAD, which are marked by significant distributional shifts. These substyles introduce nuanced, transformation-sensitive variations that adeptly capture and adapt to intricate distributional changes, enabling the model to reconstruct complex patterns within the target domain. This capability highlights the nuanced strength of DI2SDiff++ in addressing the multifaceted challenges of cross-domain data synthesis.

Ablation on multi-head style conditioner. Table 3 (Lines 12–14) presents the ablation results of DI2SDiff++ under different encoder configurations. Specifically, we evaluate the performance impact of removing either the CNN or Transformer component from the multi-head style conditioner. When the CNN component is removed (Line 12), we observe a consistent and significant performance drop across all datasets. This demonstrates that local feature extraction is critical, as its absence impairs the model's ability to capture fine-grained style features necessary for distinguishing subtle intra-class variations. Similarly, removing the Transformer component (Line 13) results in a noticeable performance degradation. This suggests that, while CNNs effectively capture rich local patterns, the Transformer complements them by modeling long-range dependencies and infusing global semantic context into the features, resulting in more meaningful substyles. When both components are jointly leveraged (Line 14), the model achieves the best average performance across all tasks, highlighting the complementary strengths of the CNN and Transformer in capturing both local and global style semantics. Overall, these results validate the effectiveness of our hybrid encoder design.

Hyperparameter Sensitivity Analysis. We evaluate the sensitivity of hyperparameters by varying one parameter while keeping others constant. The results, as shown in Fig. 8, offer critical insights into the performance and efficiency of DI2SDiff and DI2SDiff++ under different configurations. Specifically, Fig. 8(a) highlights the impact of the number of replaced style components (ζ) in DI2SDiff++. Increasing ζ tends to improve generalization by introducing a broader range of style combinations. $\zeta = 2$ achieves a reasonable balance, yielding generally stable and robust results. Fig. 8(b) focuses on DI2SDiff, demonstrating that the optimal number of style combinations (o) varies by dataset complexity. For simpler distributions like DSADS, $o = 5$ proves sufficient. However, for the more complex USC-HAD dataset, $o = 10$ is necessary to achieve the diversity required for robust generalization. By contrast, DI2SDiff++ circumvents the need for o altogether, leveraging its multi-head style conditioner to extract four random substyle conditions for each instance. This mechanism inherently captures diverse and independent aspects of activity instances, eliminating reliance on extensive hyperparameter tuning while enhancing efficiency and accuracy.

Additionally, Fig. 8(c) reveals that increasing the volume of generated data (κ) consistently improves generalization performance. For practical efficiency, we adopt $\kappa = 1$ or 2 for DI2SDiff and $\kappa = 1$ for DI2SDiff++ in Tab. 1 and Tab. 2. The results clearly demonstrate that DI2SDiff++ outperforms all baselines across tasks, achieving superior performance with minimal computational overhead. In contrast, DI2SDiff requires generating a larger volume of data ($\kappa \geq 2$) to attain comparable results, particularly for challenging datasets like USC-HAD, which increases generation costs significantly. These findings underscore the enhanced efficiency and effectiveness of DI2SDiff++. By reducing re-

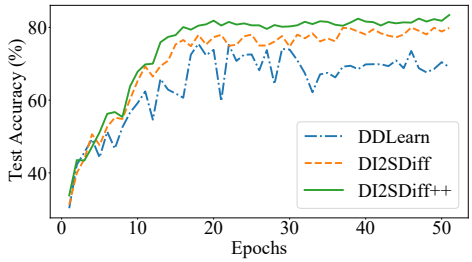


Fig. 9. Trend of test accuracy in comparison as the number of epochs increases on the first tasks of the PAMAP2 dataset with 20% training data.

TABLE 4
Efficiency analysis. Training memory(MB) and storage(MB) include the entire pipeline: the conditioner (if applicable), generator (if applicable), and classifier. Training time(min) reflects the total end-to-end duration. Inference memory(MB), deployment size(MB) and time(ms) are measured for the classifier during label prediction. An asterisk (*) denotes the default setting.

Methods	Training			Inference			Accuracy (%)
	GPU Memory	Storage	Time	GPU Memory	Deploy Size	Time	
TS-TCC	11.3	1.8	2.6	9.6	1.7	1.5	64.4
CrossHAR	25.6	3.8	7.5	12.9	3.8	2.8	70.5
MobHAR	18.5	3.0	10.3	8.7	0.3	1.2	65.6
DDLearn (κ=1)*	12.7	0.2	9.0	2.9	0.1	0.7	70.9
DDLearn (κ=3)	12.7	0.2	13.5	2.9	0.1	0.7	71.5
DDLearn (κ=5)	12.7	0.2	21.0	2.9	0.1	0.7	71.2
DI2SDiff	145.1	147.7	32.5	2.9	0.1	0.7	77.8
DI2SDiff++ (κ=1)*	152.3	148.0	17.9	2.9	0.1	0.7	80.1
DI2SDiff++ (κ=3)	152.3	148.0	34.0	2.9	0.1	0.7	84.6
DI2SDiff++ (κ=5)	152.3	148.0	47.6	2.9	0.1	0.7	86.0

liance on hyperparameter searches and streamlining data generation processes, DI2SDiff++ achieves superior generalization capabilities with lower computational demands.

7.5 Case Study of Class-Wise and Efficiency Analysis

To evaluate classification performance, we conducted a detailed case study using confusion matrices and efficiency metrics. The confusion matrices in Fig. 7 reveal that Mixup struggles to achieve satisfactory results, likely due to its tendency to distort semantic integrity during data augmentation. In contrast, DDLearn achieves moderate improvements by leveraging class-maintained standard data augmentation, which better preserves activity-specific semantics. Both DI2SDiff and DI2SDiff++ significantly enhance the performance of poorly classified activities by introducing greater intra-class diversity. Notably, DI2SDiff++ demonstrates an improvement over DI2SDiff, particularly in reducing misclassifications for dynamic and challenging activities such as "ascending stairs" and "descending stairs." This advancement can be attributed to the multi-view substyles employed in DI2SDiff++, which diversify fine-grained patterns, capturing subtle variations essential for robust class differentiation.

The test accuracy trends depicted in Fig. 9 further underscore the superiority of our approach. Both DI2SDiff and DI2SDiff++ exhibit faster convergence and achieve significantly higher accuracy compared to DDLearn, highlighting the efficiency of diffusion-based data generation in enhancing the learning process of HAR classifiers. Among all methods, DI2SDiff++ emerges as the most effective, consistently delivering the best overall performance.

7.6 Complexity Analysis

We evaluate the training and inference complexity of our method and the baselines following the protocol in [9]. We report GPU memory usage for computation, storage overhead, total training time, inference time, and accuracy. All experiments are conducted on an NVIDIA RTX A5000 GPU using Task 3 of the USC-HAD dataset (a challenging DG task), where the batch size is set to 1.

The results are summarized in Table 4. As shown, DI2SDiff-based models incur higher training overhead in terms of GPU memory (152.3 MB), storage (148.0 MB), and total training time (17.9 minutes) due to the additional multi-head style conditioner and the diffusion process. However, this cost is amortized, as training is performed only once, allowing the model to generalize effectively across unseen domains without the need for retraining. The additional 5% GPU memory overhead and 0.2% storage overhead introduced by the multi-head conditioner, compared to DI2SDiff, can be further optimized by switching from parallel to sequential GPU execution. Notably, the default configuration of DI2SDiff++ ($\kappa = 1$) outperforms DI2SDiff while reducing training time by approximately 45%, owing to shorter generation time.

After training, only the HAR classifier (a 3-layer CNN) trained on the original data and generated data, is used for prediction. As the backbone is shared among DDLearn, DI2SDiff, and DI2SDiff++, the resulting GPU memory usage, deployment size, and inference time remain identical, thereby incurring no additional overhead. The results further demonstrate that DI2SDiff++ consistently achieves the lowest inference memory usage (2.9 MB), a minimal deployment size (0.1 MB), and the shortest inference time (0.7 ms), while surpassing all baseline methods in accuracy. These findings underscore the strong suitability of DI2SDiff++ for real-world edge deployments, offering an excellent balance between performance and efficiency.

To further validate our method's efficiency, we conduct experiments with varying values of κ. DI2SDiff++ offers flexible control over the trade-off between training cost and performance through the data expansion ratio κ: increasing κ leads to longer training times but consistently yields better accuracy, enabling fine-grained adaptability based on task requirements. In contrast, augmentation-based approaches such as DDLearn struggle to provide significant performance gains via traditional augmentation techniques, which often produce redundant samples. This highlights their reliance on manually collected data, which is typically both time-consuming and costly. Overall, despite requiring only modest GPU resources and storage overhead during training, our approach achieves substantial improvements in both accuracy and efficiency, without introducing additional inference cost.

7.7 Training Time Comparison

To support the efficiency of our DI2SDiff++ compared to DI2SDiff, we further report the average training time on each dataset task. The results are summarized in Table 5. The results show that DI2SDiff++ consistently requires less training time across all tasks. In particular, when the training data is limited to only 20%, DI2SDiff++ achieves better

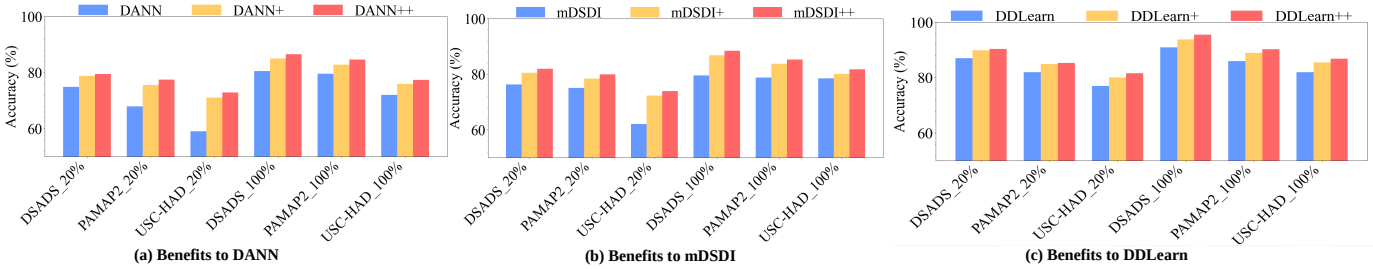


Fig. 10. Enhanced performance of (a) DANN [74], (b) mDSDI [20] and (c) DDLearn [14] with DI2SDiff's data generation (+) and DI2SDiff++'s data generation (++) on 20% and 100% training data in three datasets.

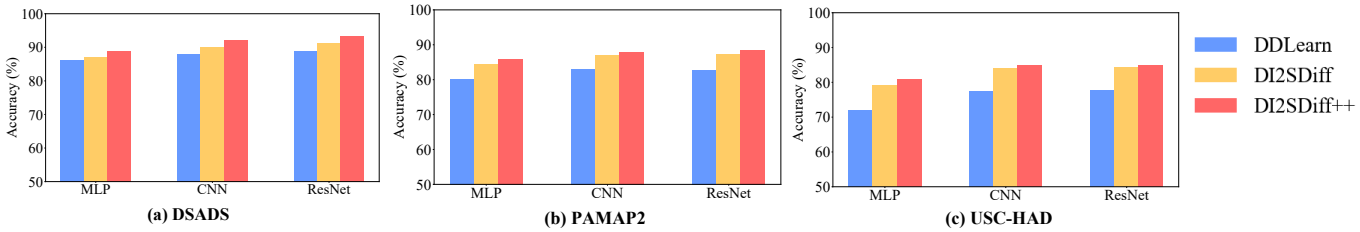


Fig. 11. Results with different backbones on the three datasets.

TABLE 5
Training time comparison (minutes) on RTX A5000 under 20% and 100% data availability for three datasets.

Methods	DSADS		PAMAP2		USC-HAD		**Avg**
	20%	100%	20%	100%	20%	100%	
DI2SDiff	15.9	38.5	25.5	45.2	37.9	58.3	36.9
DI2SDiff++	10.8	21.2	17.5	28.9	18.5	43.6	**23.4**

performance with just a single data expansion (as shown in Tables 1 and 2), whereas DI2SDiff tends to rely on larger expansion ratios to enhance diversity, resulting in significantly longer training times. Thus, DI2SDiff++ achieves up to 37% faster training while maintaining higher accuracy, demonstrating both its effectiveness and efficiency.

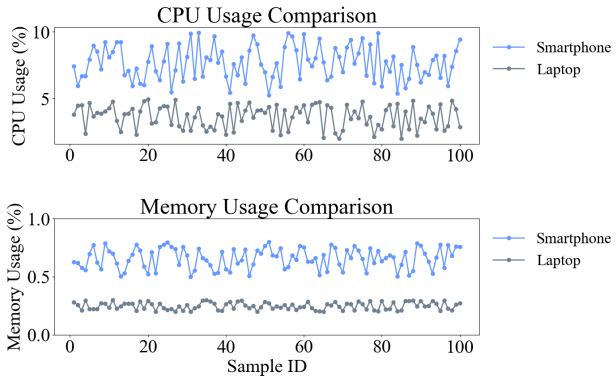


Fig. 12. Performance overhead analysis of DI2SDiff++ on edge devices in terms of CPU usage (%) and memory usage (%) across both smartphone and laptop platforms.

7.8 Deployment on Mobile Devices

We present a practical case study demonstrating the deployment and evaluation of our HAR model, DI2SDiff++, on real-world edge devices. The model was tested on two platforms: a HUAWEI Mate 70 smartphone running HarmonyOS 4.3 with 12GB of RAM and a Kirin 9010 processor, and a laptop equipped with an Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz. We first trained a 3-layer CNN classifier using the DSADS dataset along with synthetic data generated by DI2SDiff++. Following the deployment setting described in [55], the resulting classifier is saved as a '.pt' file for real-time activity recognition. This final model file loaded into the application is lightweight, occupying only 0.1MB.

To evaluate real-world performance, we conducted a case study involving three volunteers aged 25, 27, and 29. Each participant performed four activities: sitting, walking, walking in a parking lot, and sitting. Data was collected for 2 minutes per activity. The model provided predictions every 5 seconds. The activity recognition accuracies for the three participants were 72.5%, 74.4%, and 65.8%, respectively. We observed that walking in a parking lot was particularly challenging to distinguish for the oldest participant (age 29), which contributed to the lower performance. Moreover, the fluctuations in model performance may be attributed to real-world uncertainties, such as sudden car movements or environmental changes like shifting weather conditions. These findings highlight the inherent nature of real-world time-series data: non-stationary and subject to evolving distributional shifts driven by subtle and often unpredictable external factors. These results underscore the importance of domain generalization research and highlight future opportunities for advancement within the HAR community.

Regarding latency, the average inference time per sample on the smartphone was 2.7 ms, with a minimum of 0.5 ms and a maximum of 9.8 ms. We also evaluated the model's inference efficiency on a more advanced edge device,i.e., the laptop, using a batch size of 1. On the laptop, the average in-

ference time per sample was 0.5 ms, with the minimum and maximum being 0.1 ms and 2 ms, respectively. Moreover, we conducted a detailed analysis of CPU usage (%) and memory usage (%), as illustrated in Fig. 12. The on-device inference overhead of our DI2SDiff++ model remains below 10% CPU and 1% memory usage, which falls well within the acceptable range for modern mobile and laptop devices [54]. Overall, the overhead is acceptable when considering the performance gains achieved by DI2SDiff++.

7.9 Extensibility and Varying backbones

We demonstrate the extensibility of DI2SDiff and DI2SDiff++ in boosting the performance of existing DG baselines. The results are shown in Fig. 10. By incorporating our synthetic data into the training datasets of baselines, we consistently observe performance improvements across the board, including DANN [74], mDSDI [20][2], and DDLearn [14][3]. This demonstrates the versatility of integrating our method to provide additional gains, making it a practical solution for immediate application. The diverse synthetic data of DI2SDiff and DI2SDiff++ is thus ready for use, offering a straightforward way to bolster various baselines without necessitating further data generation.

Fig. 11 presents the performance results of DDLearn, DI2SDiff, and DI2SDiff++ across three different backbone architectures (MLP, CNN, and ResNet) on the three datasets. The results confirm that DI2SDiff++ not only generalizes well across different datasets but also effectively boosts the performance of diverse backbone architectures, including lightweight models like MLP. This versatility underscores its potential for deployment in real-world HAR applications with varying computational and architectural constraints.

8 Conclusion

In this paper, we tackle the key issue of DG in cross-person activity recognition, i.e., the limited diversity in the source domain. We introduce a novel concept called "domain padding" and propose DI2SDiff and DI2SDiff++ to realize this concept. Our approach generates highly diverse intra- and inter-domain data distributions by utilizing random style fusion. Through extensive experimental analyses, we demonstrate that our generated samples effectively pad domain gaps. By leveraging these new samples, our DI2SDiff and DI2SDiff++ outperform advanced DG methods in various HAR tasks. A notable advantage of our work is its efficient generation of diverse data from a limited number of labeled samples. This potential enables DI2SDiff and DI2SDiff++ to provide data-driven solutions to various models, thereby reducing the dependence on costly human data collection.

References

[1] Y. Zhang, X. Wang, Y. Wang, and H. Chen, "Human activity recognition across scenes and categories based on csi," *IEEE Transactions on Mobile Computing*, 2020.

[2] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, "Wifi csi based passive human activity recognition using attention based blstm," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2714–2724, 2018.

[3] Z. Wang, Y. Chen, H. Zheng, M. Liu, and P. Huang, "Body rfid skeleton-based human activity recognition using graph convolution neural network," *IEEE Transactions on Mobile Computing*, 2023.

[4] S. Ishimaru, K. Hoshika, K. Kunze, K. Kise, and A. Dengel, "Towards reading trackers in the wild: Detecting reading activities by eog glasses and deep neural networks," in *Ubicomp*, 2017, pp. 704–711.

[5] S. Inoue, P. Lago, T. Hossain, T. Mairittha, and N. Mairittha, "Integrating activity recognition and nursing care records: The system, deployment, and a verification study," *IMWUT*, vol. 3, no. 3, pp. 1–24, 2019.

[6] S. Shao, Y. Guan, B. Zhai, P. Missier, and T. Plötz, "Convboost: Boosting convnets for sensor-based activity recognition," *IMWUT*, vol. 7, no. 2, pp. 1–21, 2023.

[7] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019.

[8] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018.

[9] S. Xia, L. Chu, L. Pei, J. Yang, W. Yu, and R. C. Qiu, "Timestamp-supervised wearable-based activity segmentation and recognition with contrastive learning and order-preserving optimal transport," *IEEE Transactions on Mobile Computing*, 2024.

[10] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern recognition letters*, vol. 119, pp. 3–11, 2019.

[11] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. 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.

[12] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu, "Generalizing to unseen domains: A survey on domain generalization," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[13] H. Qian, S. J. Pan, and C. Miao, "Latent independent excitation for generalizable sensor-based cross-person activity recognition," in *AAAI*, vol. 35, no. 13, 2021, pp. 11 921–11 929.

[14] X. Qin, J. Wang, S. Ma, W. Lu, Y. Zhu, X. Xie, and Y. Chen, "Generalizable low-resource activity recognition with diverse and discriminative representation learning," *arXiv preprint arXiv:2306.04641*, 2023.

[15] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International conference on machine learning*. PMLR, 2013, pp. 10–18.

[16] S. Erfani, M. Baktashmotlagh, M. Moshtaghi, X. Nguyen, C. Leckie, J. Bailey, and R. Kotagiri, "Robust domain generalisation by enforcing distribution invariance," in *IJCAI-16*. AAAI Press, 2016, pp. 1455–1461.

[17] F. Zhou, Z. Jiang, C. Shui, B. Wang, and B. Chaib-draa, "Domain generalization with optimal transport and metric learning," *arXiv preprint arXiv:2007.10573*, vol. 2, 2020.

[18] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," *arXiv preprint arXiv:1412.4446*, 2014.

[19] Y.-F. Zhang, J. Wang, J. Liang, Z. Zhang, B. Yu, L. Wang, D. Tao, and X. Xie, "Domain-specific risk minimization for domain generalization," in *SIGKDD*, 2023, pp. 3409–3421.

[20] M.-H. Bui, T. Tran, A. Tran, and D. Phung, "Exploiting domain-specific features to enhance domain generalization," *NeurIPS*, vol. 34, pp. 21 189–21 201, 2021.

[21] K. Xu, M. Zhang, J. Li, S. S. Du, K.-i. Kawarabayashi, and S. Jegelka, "How neural networks extrapolate: From feedforward to graph neural networks," *arXiv preprint arXiv:2009.11848*, 2020.

[22] Y. Wang, Y. Xu, J. Yang, Z. Chen, M. Wu, X. Li, and L. Xie, "Sensor alignment for multivariate time-series unsupervised domain adaptation," in *AAAI*, vol. 37, no. 8, 2023, pp. 10 253–10 261.

[23] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[2] In mDSDI, our synthetic data is treated as a new domain.

[3] In DDLearn, our synthetic data is treated as a new augmentation method.

[24] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, vol. 33, pp. 6840–6851, 2020.

[25] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," in *IJCAI*. International Joint Conferences on Artificial Intelligence Organization, 2021.

[26] J. Zhang, L. Feng, Z. Liu, Y. Wu, Y. He, Y. Dong, and D. Xu, "Diverse intra-and inter-domain activity style fusion for cross-person generalization in activity recognition," *arXiv preprint arXiv:2406.04609*, 2024.

[27] Y. Hao, R. Zheng, and B. Wang, "Invariant feature learning for sensor-based human activity recognition," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 4013–4024, 2021.

[28] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition." in *IJCAI*, vol. 15. Buenos Aires, Argentina, 2015, pp. 3995–4001.

[29] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *MM*, 2015, pp. 1307–1310.

[30] S. Zhang, Y. Li, S. Zhang, F. Shahabi, S. Xia, Y. Deng, and N. Alshurafa, "Deep learning in human activity recognition with wearable sensors: A review on advances," *Sensors*, vol. 22, no. 4, p. 1476, 2022.

[31] H. Wu, Q. Huang, D. Wang, and L. Gao, "A cnn-svm combined model for pattern recognition of knee motion using mechanomyography signals," *Journal of Electromyography and Kinesiology*, vol. 42, pp. 136–142, 2018.

[32] S. Matsui, N. Inoue, Y. Akagi, G. Nagino, and K. Shinoda, "User adaptation of convolutional neural network for human activity recognition," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 753–757.

[33] R. Gupta, I. S. Dhindsa, and R. Agarwal, "Continuous angular position estimation of human ankle during unconstrained locomotion," *Biomedical Signal Processing and Control*, vol. 60, p. 101968, 2020.

[34] J. Shi, D. Zuo, and Z. Zhang, "A gan-based data augmentation method for human activity recognition via the caching ability," *Internet technology letters*, vol. 4, no. 5, p. e257, 2021.

[35] W. Seok, Y. Kim, and C. Park, "Pattern recognition of human arm movement using deep reinforcement learning," in *ICOIN*. IEEE, 2018, pp. 917–919.

[36] M. Qiao, S. Yan, X. Tang, and C. Xu, "Deep convolutional and lstm recurrent neural networks for rolling bearing fault diagnosis under strong noises and variable loads," *Ieee Access*, vol. 8, pp. 66 257–66 269, 2020.

[37] S. K. Challa, A. Kumar, and V. B. Semwal, "A multibranch cnn-bilstm model for human activity recognition using wearable sensor data," *The Visual Computer*, vol. 38, no. 12, pp. 4095–4109, 2022.

[38] X. Zhang, L. Yao, X. Wang, W. Zhang, S. Zhang, and Y. Liu, "Know your mind: Adaptive cognitive activity recognition with reinforced cnn," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 896–905.

[39] M. Ragab, Z. Chen, M. Wu, C. S. Foo, C. K. Kwoh, R. Yan, and X. Li, "Contrastive adversarial domain adaptation for machine remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5239–5249, 2020.

[40] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *ICCV*, 2015, pp. 2551–2559.

[41] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5400–5409.

[42] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1414–1430, 2016.

[43] H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo, "Reducing domain gap by reducing style bias," in *CVPR*, 2021, pp. 8690–8699.

[44] J. Huang, D. Guan, A. Xiao, and S. Lu, "Fsdr: Frequency space domain randomization for domain generalization," in *CVPR*, 2021, pp. 6891–6902.

[45] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017, pp. 1501–1510.

[46] S. Lee, H. Seong, S. Lee, and E. Kim, "Wildnet: Learning domain generalized semantic segmentation from the wild," in *CVPR*, 2022, pp. 9936–9946.

[47] R. Gal, O. Patashnik, H. Maron, A. H. Bermano, G. Chechik, and D. Cohen-Or, "Stylegan-nada: Clip-guided domain adaptation of image generators," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–13, 2022.

[48] J. Cho, G. Nam, S. Kim, H. Yang, and S. Kwak, "Promptstyler: Prompt-driven style generation for source-free domain generalization," in *ICCV*, 2023, pp. 15 702–15 712.

[49] R. Gong, M. Danelljan, H. Sun, J. D. Mangas, and L. Van Gool, "Prompting diffusion representations for cross-domain semantic segmentation," *arXiv preprint arXiv:2307.02138*, 2023.

[50] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[51] G. Wilson, J. R. Doppa, and D. J. Cook, "Multi-source deep domain adaptation with weak supervision for time-series sensor data," in *SIGKDD*, 2020, pp. 1768–1778.

[52] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[53] Z. Hong, Z. Li, S. Zhong, W. Lyu, H. Wang, Y. Ding, T. He, and D. Zhang, "Crosshar: Generalizing cross-dataset human activity recognition via hierarchical self-supervised pretraining," *IMWUT*, vol. 8, no. 2, pp. 1–26, 2024.

[54] M. Xue, Y. Zhu, W. Xie, Z. Wang, Y. Chen, K. Jiang, and Q. Zhang, "Mobhar: Source-free knowledge transfer for human activity recognition on mobile devices," *IMWUT*, vol. 9, no. 1, pp. 1–24, 2025.

[55] G. Dai, H. Xu, H. Yoon, M. Li, R. Tan, and S.-J. Lee, "Contrastsense: Domain-invariant contrastive learning for in-the-wild wearable sensing," *IMWUT*, vol. 8, no. 4, pp. 1–32, 2024.

[56] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *CVPR*, 2022, pp. 11 461–11 471.

[57] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[58] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022, pp. 10 684–10 695.

[59] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," *arXiv preprint arXiv:2209.14916*, 2022.

[60] B. Huang, K. Zhang, J. Zhang, J. Ramsey, R. Sanchez-Romero, C. Glymour, and B. Schölkopf, "Causal discovery from heterogeneous/nonstationary data," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 3482–3534, 2020.

[61] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[62] I. Nicholas, H. Kuo, F. Garcia, A. Sonnerborg, M. Bohm, R. Kaiser, M. Zazzi, L. Jorm, and S. Barbieri, "Synthetic health-related longitudinal data with mixed-type variables generated using diffusion models," in *NeurIPS 2023 Workshop on Synthetic Data Generation with Generative AI*, 2023.

[63] Y. Yang, M. Jin, H. Wen, C. Zhang, Y. Liang, L. Ma, Y. Wang, C. Liu, B. Yang, Z. Xu *et al.*, "A survey on diffusion models for time series and spatio-temporal data," *arXiv preprint arXiv:2404.18886*, 2024.

[64] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–39, 2023.

[65] T. T. Um, F. M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in *ICMI*, 2017, pp. 216–220.

[66] P. Chu, X. Bian, S. Liu, and H. Ling, "Feature space augmentation for long-tailed data," in *ECCV*. Springer, 2020, pp. 694–710.

[67] B. Barshan and M. C. Yüksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *The Computer Journal*, vol. 57, no. 11, pp. 1649–1667, 2014.

[68] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th international symposium on wearable computers*. IEEE, 2012, pp. 108–109.

[69] M. Zhang and A. A. Sawchuk, "Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *UbiComp*, 2012, pp. 1036–1043.

[70] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang, "Adversarial domain adaptation with domain mixup," in *AAAI*, vol. 34, no. 04, 2020, pp. 6502–6509.
[71] Z. Huang, H. Wang, E. P. Xing, and D. Huang, "Self-challenging improves cross-domain generalization," in *ECCV*. Springer, 2020, pp. 124–140.
[72] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
[73] Y. Shi, J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve, "Gradient matching for domain generalization," *arXiv preprint arXiv:2104.09937*, 2021.
[74] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.
[75] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

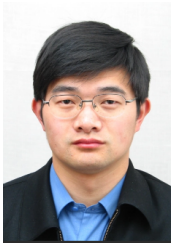
Junru Zhang received her B.S. degree from the Department of Computer Science and Technology at Henan Normal University in 2021. She is currently a Ph.D. candidate in the Department of Computer Science and Technology at Zhejiang University. Her research interests include time series data mining, with a specific focus on the development of self-supervised learning and transfer learning.

Cheng Peng received the BEng degree in software engineering from Dalian University of Technology, in 2020, and is currently working toward a doctoral degree with the Data Intelligence Laboratory, College of Computer Science and Technology, Zhejiang University, Hangzhou, China. His research interest focuses on machine learning and data mining, especially on multi-label learning and multi-modal learning

Zhidan Liu received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2014. After that, he worked as a Research Fellow in Nanyang Technological University, Singapore, and a faculty member with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is currently an Assistant Professor at Intelligent Transportation Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou). His research interests include Artificial Internet of Things, mobile computing, urban computing, and big data analytics. He is a senior member of IEEE and CCF, a member of ACM.

Lang Feng received his B.Eng. (Hons.) in Information Engineering from Southeast University in June 2021 and his M.Eng. degree in Computer Science from Zhejiang University in March 2024. He is currently pursuing the Ph.D. in Computer Science at Nanyang Technological University. His research interests include reinforcement learning, large language models, as well as LLM-based agents.

Yuhan Wu is currently a Ph.D. student in Computer Science and Technology, at Zhejiang University. She received her B.E. degree in College of Information and Electrical Engineering from China Agricultural University in 2019. Her research interests mainly lie in time series data mining, contrastive learning, time series forecasting and classification, and heuristic algorithms.



Yabo Dong received his Ph.D. degree from Zhejiang University in Hangzhou, China, specializing in Computer Application. He is an Associate Professor at the College of Computer Science and Technology, Zhejiang University. His main research interests include Internet of Things technology, time series data analysis, new low-power sensor technologies, and more.

Duanqing Xu received his Ph.D. degree from Zhejiang University, Hangzhou, China. He is currently a Professor at the College of Computer Science and Technology at Zhejiang University. His research interests encompass sensor data mining and analysis, deep learning and Artificial Internet of Things.