2024 33rd International Conference on Computer Communications and Networks (ICCCN) | 979-8-3503-8461-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCCN61486.2024.10637566

Data Augmentation for Human Activity Recognition via Condition Space Interpolation within a Generative Model

Tianshi Wang, Yizhuo Chen, Qikai Yang, Dachun Sun, Ruijie Wang,
Jinyang Li, Tomoyoshi Kimura, Tarek Abdelzaher
The Department of Computer Science
University of Illinois Urbana-Champaign
Urbana, IL, USA
{tianshi3, yizhuoc, qikaiy2, dsun18, ruijiew2, jinyang7, tkimura4, zaher}@illinois.edu

***Abstract*—This paper presents a generative data augmentation approach for human activity recognition (HAR) to close the distribution gap between laboratory training and real-world deployment. Despite the recent success of deep learning methods in wearable sensor-based HAR tasks, performance degradation occurs during real-world deployment due to training data scarcity and the vast variability in human activities. In light of this, we aim to enhance the diversity of training datasets by generating new data points within the vicinity of existing samples, as informed by domain expertise. Unlike the commonly utilized methods that augment data by interpolating in *data space* or *feature space*, we innovate by applying interpolation in the *condition space* of a conditional generative model to augment HAR datasets. We use domain-specific knowledge to extract statistical metrics from sensor data, which serve as conditions to direct the generation process. We demonstrate how a conditional generative diffusion model, steered by interpolated conditions, can synthesize realistic new data with various high-level features that benefit the robustness of the downstream HAR models. Our methodology advances the use of interpolation in data augmentation by exploring the capability of a state-of-the-art generative model, offering novel perspectives for bolstering the robustness and generalizability of HAR systems. Experimental results demonstrate that condition space interpolation outperforms the conventional interpolation-based and generative model-based augmentation methods across various datasets and downstream classifier combinations.**

***Index Terms*—human activity recognition, data augmentation, generative model, diffusion model**

I. Introduction

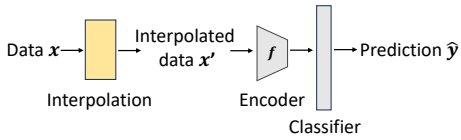
Wearable sensor-based Human Activity Recognition (HAR) has emerged as a critical area of research, driven by its significant potential applications in health monitoring, physical performance analytics, and virtual reality. Wearable sensor signals capture distinct patterns as records of the physical movements of a human body, enabling end-to-end activity recognition by state-of-the-art deep learning methods. However, their effectiveness is often compromised due to variations in the deployment scenario not represented in the training data. This limitation poses a significant challenge for the practical application of HAR technologies.

A considerable amount of research has been dedicated to enhancing the robustness of HAR models against variability in deployment conditions. Among various strategies, data augmentation stands out as a key technique. It involves defining a legitimate vicinity around data samples and generating synthetic samples within this distribution. Such an approach introduces plausible real-world variations to the original (training) dataset, thereby improving the robustness of the downstream HAR models in practical scenarios, when trained on the augmented data.

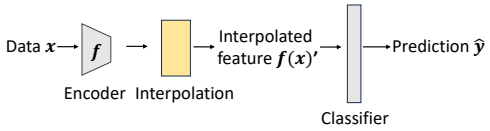
Extensive research has underscored the efficacy of interpolation between samples within the same class as an important data augmentation technique [1]–[4]. Successful interpolation is predicated on the assumption that data samples within the same class lie in a "neighborhood" where interpolating between samples can generate new samples that are still representative of that class. The intuition is that if the model can correctly classify not only the original samples but also these in-between points, it is likely to perform better on unseen data that fall within the same general manifold.

Previous research has explored interpolation within the *data space* [1]. Illustrated in Figure 1a, this process occurs before the original data sample x is fed into the downstream neural network, which consists of both a feature extraction encoder and a classifier. Another approach, *feature space* interpolation [1]–[4], is applied after x has been processed by an encoder that projects it into the latent feature space, as depicted in Figure 1b. The underlying rationale is that interpolating (training data) at a higher level of abstraction effectively enlarges the volume of feasible data points in the feature space that the downstream model is trained with, thereby reducing the proportion of space that the model has not seen [2], [5].

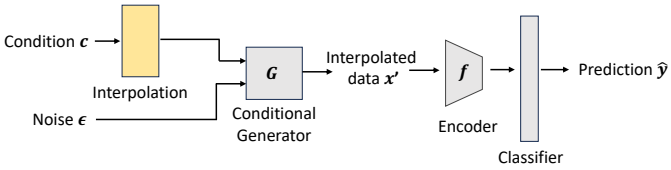
In this work, we explore this concept further by applying interpolation in the *condition space* that steers the generation process of a conditional generative model. Generative neural networks excel in capturing and replicating the complexity of the training data, offering the prospect of creating realistic synthetic data indistinguishable from the real data. Given



(a) Data Space Interpolation



(b) Feature Space Interpolation



(c) Condition Space Interpolation (Ours)

Fig. 1: Concepts of interpolation in different spaces.

specified conditions, the conditional generative model further provides control over the characteristics of the synthesized samples. This controlled approach avoids random sampling from the dataset, ensuring that new samples adhere closely to the desired feature distributions.

We illustrate our concept of condition space interpolation in Figure 1c. Rather than performing interpolation directly on the data sample x or its latent features, our approach applies interpolation to the condition c. This interpolated condition serves as the input for the conditional generator G, which in turn produces the interpolated data x' utilized in training the downstream model.

To construct the condition c, we extract statistical metrics from each data sample, based our expertise in HAR sensor signals. These metrics, such as maximum, minimum, deviation, entropy, and centroid, capture the high-level features of the data. Within a given activity type, such features cover the variance in sensor signals shaped by physical aspects like intensity, speed, and frequency, akin to distinct "styles" [6]. Our novel approach involves linearly interpolating these (condition) metrics between pairs of samples from the same activity class, enabling us to guide the generation of new samples in a specific, interpolated style. We choose a *conditional diffusion model* as the generative model due to its training stability and superior generation quality. We summarize our method with three key advantages:

1) **Domain-Specific Insights**: By grounding our conditions in the statistical metrics of HAR data, we achieve strong control over the style variance of the generated data.
2) **Vicinity Exploration**: Through the interpolation of the statistical properties, we leverage the inherent notion of vicinity defined within each activity type to create plausible high-level features (not present in the original training data).
3) **Realistic Synthesis**: the power of the diffusion model ensures that the newly created samples closely align with the conditions specified while demonstrating realism in the semantics of spectrograms.

In the rest of the paper, we first review the related work in Section II. Next, we introduce our proposed methodology in detail in Section III. In Section IV, the evaluation results are presented. Finally, the paper concludes in Section V.

II. Related Work

A. Human Activity Recognition

Wearable sensor-based human activity recognition (HAR) has been a popular research topic since the late '90s, when feature extraction (FE) methodologies were initially adopted [7]. These approaches involve manually selecting statistical features from the raw data, building a standard profile from the training set, and classifying the testing samples by quantitatively comparing the features with the standard profile. Even though the well-designed statistical features can be representative of a limited dataset, they are less generalizable to a more diverse population and set of activities. Machine learning models were widely introduced to solve HAR tasks in recent decades [8], where classic models such as decision trees, Support Vector Machines (SVM), Bayesian models, and ensemble models were adopted. With advances in neural networks, deep learning-based models have dominated the HAR domain in the last decade [9], [10]. Deep learning models can effectively extract features from vast amounts of data, thus achieving superior performance compared to conventional machine learning models. Due to the vast variability in human activities and the high cost of data collection, HAR datasets might under-represent the real-world variance [11], [12]. In such situations, prior research has demonstrated that deep learning models can potentially overfit the training datasets [13], leading to poor generalizability to deployment conditions. To avoid overfitting, much recent work considered different techniques for (training) data augmentation [14], [15]. In this paper, we further improve model robustness (in various deployment environments) by introducing a better data augmentation algorithm.

B. Data Augmentation

Data augmentation is an effective approach to improve model robustness in HAR tasks. Traditional techniques involve manually crafted transformations to generate variants from existing samples. These include operations like cropping, flipping, and jittering, applicable across time, frequency, and time-frequency domains [16], [17]. However, their reliance on an empirically determined vicinity may not generalize well across diverse scenarios and can introduce synthetic artifacts that compromise the realism of generated data. With the evolution of generative artificial intelligence, generative neural networks have been extensively utilized for data augmentation, such as Generative Adversarial Networks (GANs) [14], Variational Autoencoders (VAEs) [18], and diffusion models [15],

[19]. They learn the data distribution of the training dataset conditioned by the corresponding activity labels. During the generation stage, the trained generative models are adopted to introduce new variations in the generated samples by taking the target activity labels as inputs. While these generative models are adept at producing outputs that closely resemble real data, the generation process primarily depends on random sampling within the designated activity category, resulting in a lack of precise control over the desired attributes of the generated samples. Our approach mitigates this challenge by interpolating in the condition space of the conditional generative model.

C. Diffusion Model

A diffusion model is a probabilistic generative model that simulates the process of information or signal propagation through a medium by iteratively refining random noise into coherent structures resembling the target data distribution [20]. In the previous literature, diffusion models have been employed to generate images [21], [22], human speech [23], [24] and music [25]. It has been shown that diffusion models are versatile in handling various data types and more stable during training compared to Generative Adversarial Networks (GANs) while generating comparable (or even better) quality results. Moreover, their exceptional flexibility and control in conditional generation are particularly advantageous for our proposed interpolation in the condition space, which relies on various statistical metrics for conditioning.

Diffusion models are based on the concept of adding noise step by step and learning to gradually denoise the data. To formally define diffusion, we first describe the forward process, which is a Markov process that adds noise to the data over a series of time steps:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon \tag{1}$$

where x_t is the data at time step t, α_t is the predefined variance schedule that defines how much noise to be added, and ϵ represents the noise sampled from a standard Gaussian distribution. The forward process gradually transforms the data into a Gaussian distribution.

In the backward process, a neural network learns the reverse of the forward process, gradually denoising the data to recover the original data distribution from the noise. The reverse process can be parameterized as:

$$p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2}$$

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are the mean and covariance of the distribution, predicted by the model at time t, parameterized by neural networks with parameters θ. The training process involves learning to predict the noise that was added at each step of the forward process, so that it can be reversed.

And the loss function is defined as:

$$\mathcal{L}_{denoise} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t, c)\|^2\right] \tag{3}$$

where c is the condition embedding for guiding the generation process. This loss function encourages to train a neural network that can accurately predict the noise that was added to the data at each step of the diffusion forward process, allowing it to generate samples from the data distribution by starting from noise and progressively denoising it.

To generate a new sample, the diffusion process begins with a sample ϵ drawn from the a standard Gaussian noise distribution. The additional condition c' is provided to the model at each step of the reverse process, guiding the model to generate data that aligns with c'.

III. Data Augmentation Framework

In this section, we present our data augmentation framework. We begin with an overview of the framework, followed by an in-depth introduction to its three major components.

A. Framework Overview

Let x denote a data sample collected by wearable sensors for HAR, $m \in \mathbb{R}^n$ represent the metric vector, consisting of n statistical metric extracted from sensor data x. Let y be the activity type label, encoded as a one-hot embedding vector in $\mathbb{R}^k$, where k is the number of distinct activity types. We define two neural network layers as embedding functions:

- $f_m : \mathbb{R}^n \to \mathbb{R}^{d_m}$ is the embedding function for the metric vector m, mapping it to a d_m-dimensional embedding space.
- $f_y : \mathbb{R}^k \to \mathbb{R}^{d_y}$ is the embedding function for the activity type label y, mapping it to a d_y-dimensional embedding space.

Both f_m and f_y are trained concurrently with the main conditional diffusion network to optimize their ability to represent the inputs in a manner to generating realistic synthetic data.

The condition vector c is then constructed by concatenating the embeddings of m and y, resulting in:

$$c = [f_m(m); f_y(y)] \tag{4}$$

where $[;]$ denotes the concatenation operation, and $c \in \mathbb{R}^{d_m+d_y}$ serves as the condition input to the diffusion model during both the training and generation processes.

Let G be a diffusion model parameterized by θ that outputs the generated sample x' given a noise vector ϵ and condition vector c. The statistical metric of x' is denoted as m'.

We illustrate the training and generation stage of G in Figure 2. During the training stage of G, the optimization goal can then be written as:

$$\theta^* = \arg\min_\theta \Bigg(\mathbb{E}_{(x,c)\sim p_{\text{data}}(x,c),\epsilon\sim p(\epsilon)}\left[\mathcal{L}_{denoise}(G(\epsilon, c; \theta), x)\right] + \lambda \mathbb{E}_{(x,x')\sim p_{\text{model}}(x,x')}\left[\mathcal{L}_{metric}(m, m')\right] \Bigg) \tag{5}$$

where p_{data} is the empirical distribution of the real data and its conditions, and p_{model} is the model distribution of the data, implicitly defined by G and the process of generating x' from ϵ

and c. $\mathcal{L}_{denoise}$ is the denoising loss from the diffusion model as defined in Equation 3. $\mathcal{L}_{denoise}$ is the mean squared error loss for calculating the distance between the metric values of the original and generated data.

Upon training completion, the diffusion model G generates new samples by taking a random noise vector ϵ sampled from a noise distribution $p(\epsilon)$ (a standard normal distribution in our implementation), and an interpolated condition vector c'. The interpolated metric vector m' is obtained through a linear interpolation between two metric vectors m_1 and m_2, associated with the same activity class y':

$$m' = \lambda m_1 + (1 - \lambda) m_2 \tag{6}$$

where λ is a weight parameter randomly sampled between 0 and 1 for each interpolation.

The corresponding condition vector c' is defined as:

$$c' = [f_m(m'); f_y(y')] \tag{7}$$

The diffusion model G then synthesizes the new data sample x' by:

$$x' = G(\epsilon, c'; \theta^*) \tag{8}$$

Let X' denote the set of new samples created from the generation process. The augmented dataset X_{aug} is then defined as the union of the original dataset X and the generated samples:

$$X_{aug} = X \cup X' \tag{9}$$

This augmented dataset X_{aug} is used to train downstream HAR model, H, parameterized by ϕ, with the goal of minimizing the activity type classification loss function $\mathcal{L}_{class}$ on a separate validation dataset V:

$$\phi^* = \arg\min_{\phi} \mathcal{L}_{class}(H(X_{\text{aug}}; \phi), V) \tag{10}$$

The goal is that training H on X_{aug} will lead to improved performance on V compared to training solely on the original dataset X, due to the increased diversity and coverage of human activity space within X_{aug}, enhancing the generalization and robustness of H to unseen data.

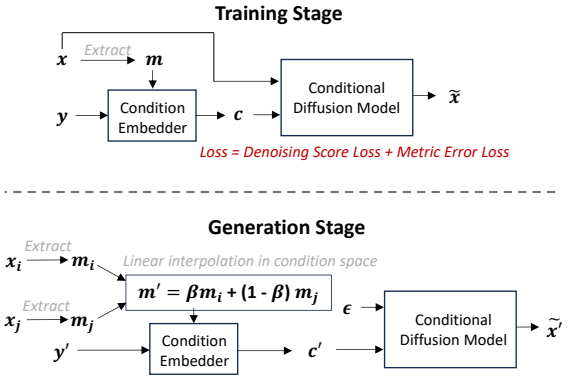


Fig. 2: Framework Overview.

In Section III-B, III-C, and III-D, we will explain the three essential components of the framework in depth.

B. Statistical Metric Extraction

Statistical metrics have been widely used as features for activity classification in numerous early HAR studies. While these features provide an overview of statistical data distributions, they may lack the granularity required for constructing a high-performance activity type classifier. However, high-level features can capture the variability in how each activity is performed, reflecting differences in speed, intensity, and frequency among individuals. This performance variability is the major discrepancy between training datasets and real-world deployment scenarios. Consequently, the statistical metrics can serve as effective proxies for the "styles" in which the same activity is performed, supporting a control space for feature interpolation.

To identify statistical metrics that effectively reflect the variance, we focus on metrics within the frequency and time-frequency domain. This emphasis is due to the periodic nature of human activities, which exhibit regular, identifiable patterns in the frequency and time-frequency domain. For each raw time series data, we use short-time Fourier transform (STFT) to convert it to 2D spectrograms.

We then empirically propose a set of statistical metrics that characterize features in these domains. To assess their effectiveness, we construct a random forest classifier for distinguishing between human subjects based on these metrics and evaluate the impurity-based importance score of each metric. The underlying hypothesis is that metrics capable of differentiating the stylistic variations associated with individual human subjects can serve as potent indicators of high-level features. Employing the importance score evaluation on RealWorld HAR dataset, we select the top nine metrics as our final statistical metrics for conditioning, as shown in Table I.

TABLE I: Selected Statistical Metrics through Impurity-based Importance Score Evaluation

Group	Metrics
Time-frequency domain	Mean, max, min, standard deviation
Frequency domain	Centroid, kurtosis, entropy, contrast

Define the STFT spectrogram as $S(f, t)$ where f represents the frequency bin and t represents the time frame. For a given time-frequency domain metric function $g_{\text{time\_freq}}$, we simply calculate the metric value $v_{\text{time\_freq}}$ by

$$v_{\text{time\_freq}} = g_{\text{time\_freq}}(S(f, t)) \tag{11}$$

For the frequency domain metrics, we average across the time axis of the STFT spectrogram, and then calculate the metric over the entire spectrum.

Let $\overline{S(f)}$ define the averaged spectrogram across the time axis:

$$\overline{S(f)} = \frac{1}{T} \sum_{t=1}^{T} S(f, t) \tag{12}$$

Given the frequency domain metric function g_{freq}, the value of a frequency domain metric v_{freq} is calculated by:

$$v_{\text{freq}} = g_{\text{freq}}(\overline{S(f)}) \tag{13}$$

The metric vector m includes all the time-frequency and frequency domain metric values:

$$\begin{aligned} m = &[v^1_{\text{time\_frequency}}, v^2_{\text{time\_frequency}}, \ldots, v^M_{\text{time\_frequency}}, \\ &v^1_{\text{frequency}}, v^2_{\text{frequency}}, \ldots, v^N_{\text{frequency}}] \end{aligned} \tag{14}$$

where M is the number of time-frequency domain metrics, and N is the number of frequency domain metrics.

C. Conditional Diffusion Model

The conditional diffusion model takes data sample x, metric vector m, and activity type label a as inputs.

It first embeds m and a into a condition vector through the condition embedder, which has a linear layer followed by a GELU activation function. The condition embedder concatenates all the metric values and the one-hot embedding of a into one vector, and pass it through the neural network. The output dimension equals the length of the time dimension F. Then it repeats for T times and turns into the same shape as the STFT spectrogram of x. The condition vector c and x are added up, feeding into the diffusion model.

We employ U-Net as the backbone of our diffusion model, leveraging its established utility in 2D data (image) generation tasks [26]. U-Net is characterized by a symmetric encoder-decoder structure, augmented with skip connections that facilitate the flow of information between corresponding layers of the downsampling and upsampling paths. The encoder, or contracting path, employs a series of convolutional layers for feature extraction, which reduce spatial dimensions while deepening feature representation. In our implementation, we use 2D convolutional layers with stride equal two for halving the height and width of the 2D data, while doubling the output feature dimension. At the core of U-Net, the bottleneck bridges the encoder and decoder, concentrating high-level features for subsequent reconstruction.

The decoder mirrors the encoder architecture in reverse. It utilizes transposed convolutional layers to progressively restore the 2D spatial dimensions. Skip connections from the encoder reintroduce localized spatial information lost during downsampling, aiding in the precise reconstruction of the output 2D data.

Each skip connection directly concatenates feature maps from the encoder to the decoder, ensuring that both high-level semantic information and detailed spatial information are preserved and utilized. This mechanism is critical for the model's performance, particularly in generating or enhancing 2D data where detail fidelity is paramount. The U-Net backbone outputs data in the same shape as the inputs, enabling the direct denoising score loss calculation in the diffusion process.

To optimize the diffusion model, we consider two losses. One is the regular diffusion denoising loss, as defined in Equation 3. The other is metric error loss, which is the mean squared error between the normalized m and m' regarding each single metric:

$$\mathcal{L}_{metric} = MSE(m_{\text{norm}}, m'_{\text{norm}}) \tag{15}$$

The final loss is the sum of the denoising loss and the metric error loss:

$$\mathcal{L} = \mathcal{L}_{denoise} + \mathcal{L}_{metric} \tag{16}$$

D. Interpolation in Condition Space

In the generation stage, we interpolate on the condition vectors of the training data to create new conditions for data augmentation. Specifically, under the same activity type, we randomly select two samples x_i and x_j under the same activity type, and extract their statistical metrics m_i and m_j. We apply linear interpolation on the two metric vectors and create the interpolated metric vector m' as defined in Equation 6. In our implementation, λ is sampled from a uniform distribution between zero to one. The interpolated metric vector m' and the activity type label y, alongside a random noise ϵ sampled from a standard normal distribution, are fed into the conditional diffusion model to generate the interpolated sample x'.

The number of the interpolated data samples is a tunable hyperparameter. By default, for each activity type, we create interpolated samples in the same number of the real samples under that activity type. More in depth investigation on the influence of the number of interpolated data samples will be discussed in Section IV-E.

The generated samples will be appended to the original dataset to train the downstream HAR classifier.

IV. Evaluation

In this section, we present the evaluation on our proposed augmentation method for HAR. First, we introduce the datasets, baseline methods, and downstream classifiers used in our study. Following this, we showcase the overall performance of our approach. Subsequently, we explore how the augmentation ratio of condition space interpolation affects the performance of the downstream classifier. Lastly, we demonstrate the efficacy of our conditional diffusion model in precisely controlling the statistical metrics of the generated samples.

A. Datasets

1) *RealWorld HAR* [27] The RealWorld HAR dataset includes data collected from 15 participants engaged in five dynamic activities, including walking, running, ascending stairs, descending stairs, and jumping. This dataset encompasses readings from 6 types of sensors located at seven different positions on the body. For our analysis, we specifically utilize data from the accelerometer and gyroscope sensors attached to the upper arm, both of which offer a sampling rate of 50 Hz. Data for each activity spans approximately 10 minutes, with the exception of jumping, which is represented by 1.5 minutes of data. We process the time-series data by segmenting it into 2.5-second intervals for both

generation and classification tasks. We randomly select 8 participants as the training set, and the rest 7 participants as the testing set.

2) *PAMAP2* [28] The PAMAP2 dataset contains data from 9 participants performing 18 distinct physical activities. In our study, we exclude static activities such as standing, sitting, and lying down, as well as activities with insufficient data. The selected activities for our analysis include biking, walking, stair climbing, running, rope jumping, and vacuum cleaning. We utilize the data from accelerometer and gyroscope taken at the wrist, with both sensors recording at a sampling rate of 100 Hz. Similar to the RealWorld HAR dataset, we segment the time-series data into 2.5-second intervals for the generation and classification processes. We randomly select 6 participants as the training set, and 3 participants as the testing set.

B. Baselines

We compare our method with 5 data augmentation baselines.

1) *No augmentation (no-aug)*: Training the downstream HAR classifier without any data augmentation. This naive approach simply ignores the discrepancy between the data distribution between the training set and
2) *Conventional augmentation (convn-aug)*: Conventional time-series augmentation methods proposed in [16], including jittering, scaling, rotation, permutation, magnitude-warping, and time-warping. We randomly apply 2 transformations on each real data sample to generate a synthetic sample.
3) *Data space interpolation (data-interp)*: We linearly interpolate between two random data samples under the same activity type as proposed in [1]. For each interpolation, the weight used to balance the two samples is randomly drawn from a uniform distribution ranging between 0 and 1.
4) *Feature space interpolation (feat-interp)*: As [2] proposed, we first train an autoencoder in a unsupervised manner, and then linearly interpolate the intermediate feature maps from two random samples under the same activity to create synthetic feature maps. The real and synthetic feature maps are then altogether used for training the downstream HAR classifier.
5) *Conditional diffusion model with activity type as condition only (act-only)*: We ablate the component of interpolation in the condition space to assess its benefit to the downstream HAR classifier. Specifically, we only use activity type as the condition to train the conditional diffusion model, and generate the number of synthetic samples equal to the real samples under that activity type. This is the typical usage of a conditional generative model for data augmentation in HAR applications. Compared with our method, this baseline approach provides no explicit control over the high-level features, blindly relying on the variation creation capability of the diffusion model.

Our approach is annotated as *cond-interp* in the following analysis.

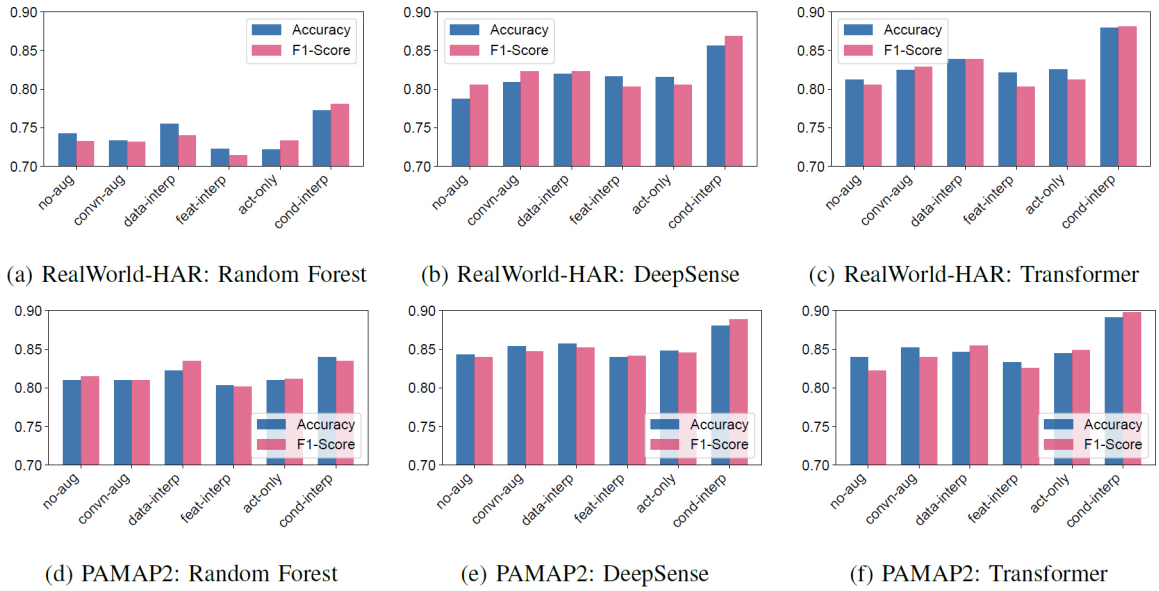
C. Downstream Classifiers

1) *Random forest*: An ensemble-based machine learning technique which employs a collection of decision tree classifiers to gain an improved predicative accuracy. The individual predictions from every tree are aggregated to reduce the risk of overfitting. In our implementation, we take the statistical metrics as the inputs for the random forest. The random forest contains 100 trees. The number of features to consider when looking for the best equals to the square root of the total number of features.
2) *DeepSense* [29]: A deep neural network designed specifically for processing Internet of Things time series data. It learns in the time-frequency domain by converting data by short-time Fourier transform. It first applies multiple 2D convolutional layers on the data to extract spatial features, and then uses Gated Recurrent Unit (GRU) layers to capture temporal dependencies. Subsequently, two linear layers are appended, serving the purpose of refining the feature dimensions and producing the final classification logits. To handle data from multiple sensors, we apply two convolutional layers for each sensor respectively, then concatenate the intermediate feature maps from the two sensors.
3) *Transformer*: A neural network built by stacking multiple Transformer encoder layers, where each incorporates a self-attention layer followed by two linear feedforward layers. In our implementation, given the input data in the shape of *(channel, frequency, time)*, we apply the self-attention along the time dimension. We apply two Transformer encoder layers for data from each sensor. Then the multi-sensor features are fused via concatenation and linear layer processing. A linear layer followed by a softmax acts as the classification head.

D. Overall Performance

In this section, we present a comparative analysis of our proposed data augmentation method, interpolation in the condition space, against the other baselines.

Overall, our proposed interpolation in the condition space outperforms the other baselines in all the test cases. Our primary findings, illustrated in Figure 3, reveal that condition space interpolation consistently beats the baseline methods across all test cases. Our method particularly enhances the performance of complex deep learning models including DeepSense and Transformer. We attribute this result to the stronger learning capability of the deep learning models, which allows them more effectively consume the extra data variations introduced from the augmented data. Conversely, as the random forest only takes the statistical metrics as input features, it gains less from the realism present in the synthetic



(a) RealWorld-HAR: Random Forest (b) RealWorld-HAR: DeepSense (c) RealWorld-HAR: Transformer

(d) PAMAP2: Random Forest (e) PAMAP2: DeepSense (f) PAMAP2: Transformer

Fig. 3: General performance evaluation. Each subfigure illustrates the performance comparison of the data augmentation methods under a dataset-classifier combination.

spectrograms generated from our method. This phenomenon is further illustrated by the marginal superiority of condition space interpolation over data space interpolation in scenarios involving random forest classifiers, as depicted in Figure 3a and 3d. Additionally, we observe that other data augmentation methods all introduce adversely affect the performance of random forest classifiers. The negative impact likely stems from the failure of these methods to interpolate within a valid range of the statistical metrics, introducing unrealistic values that confuses the downstream classifier.

Our analysis indicates that while the conventional augmentation method generally improves the performance of deep learning-based models, the enhancement is modest. The transformation techniques such as jittering and rescaling, though useful, fail to capture the complex variances inherent in real-world data, resulting in limited effectiveness and potentially introducing anomalous artificial effects. Our method, by relying on the realistic generation of the diffusion model, avoids these pitfalls, preserving the integrity of data semantics and offering more substantial improvements.

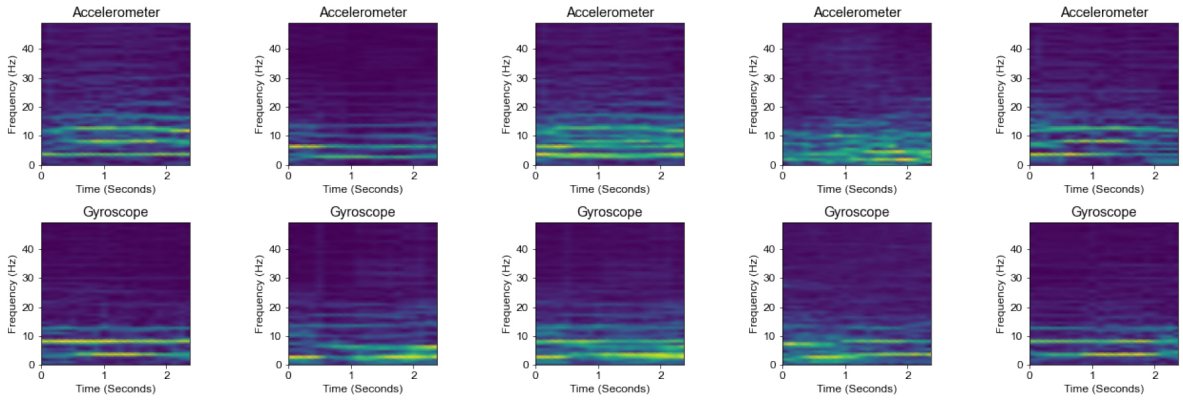
Interpolation in the data space and feature space can distort the semantic content of spectrograms, leading to inconsistent effects on classifier performance. Specifically, the performance of feature space interpolation fluctuates significantly on different dataset-classifier combinations. We speculate that feature space interpolation depends on more deliberate neural network architecture design for both the autoencoder and the downstream classifier by considering their compatibility. In other words, feature space interpolation is not downstream-agnostic, underscoring the extra necessity for fine-tuning the augmentation approach.

To further illustrate the advantages of our approach over other interpolation-based augmentation methods, we visually compare the interpolation in different spaces in Figure 4. Specifically, we present the interpolated short-time Fourier transform spectrograms from two data samples in the RealWorld-HAR dataset [27]. Figure 4a and 4b are the source samples for interpolation, both representing the "jumping" activity. These two real samples all exhibit clear bright bands at certain frequencies, serving as distinctive patterns for the downstream HAR model to identify. Interpolating these samples with equal weight in either data (Figures 4c) or feature space (Figure 4d) results in the blurring of these bright bands. This occurs as the distinct bands from the original spectrograms merge, blurring their boundaries and diminishing their separability. Such an effect demonstrates the potential for data/feature space interpolation to disrupt the semantic content of spectrograms. In contrast, our approach, visualized in Figure 4e, generates results that not only preserve the clarity of these bright bands but also introduce patterns that are distinct from the source samples in a meaningful manner, demonstrating the superiority in maintaining and enhancing the realism of interpolated spectrograms.

Lastly, we found that the conditional diffusion model solely conditioned by activity types behaves inconsistently under different scenarios. This demonstrates that blindly relying on the variations created by the generative model lacks robustness. This finding also highlights the merit in our design of using interpolated statistical metrics as conditions to direct the generation of samples with unseen while realistic variance.

E. Influence of Augmentation Ratio

As the data variability that can be offered by our generative model is not infinite, it remains unclear how many generated samples optimally benefit a downstream model. This address this question, we adjust the augmentation ratio, defined as the

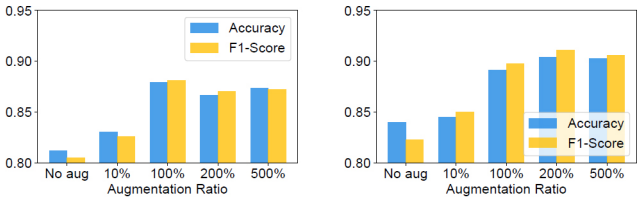


(a) Source sample 1 (b) Source sample 2 (c) Data space interpolation (d) Feature space interpolation (e) Condition space interpolation (Ours)

Fig. 4: Spectrograms visualization from different interpolation methods. Interpolating input samples in either data space (4c) or feature space (4d) results in blurring of the bright spectral bands. Condition space interpolation (4e) preserves clear yet distinct spectral band patterns.

ratio of generated samples to real samples, to investigate how varying the number of samples generated via condition space interpolation affects the downstream classifier's performance.

In our analysis, we utilize Transformer as the downstream classifier. As shown in Figure 5, the optimal augmentation ratio differs across datasets. For RealWorld-HAR dataset, the augmentation ratio peaks at 100%, whereas for the PAMAP2 dataset, a 200% ratio is optimal. Generally, when the augmentation ratio falls below the optimal point, the benefit of the data augmentation is constrained, indicating that the generative model could still producing samples with meaningful unseen variance. After the augmentation ratio reaches the optimal point, additional generated samples cease to enhance the downstream classifier. This plateau indicates that the generative model's capacity for meaningful variance has been fully utilized. Exceeding this threshold leads to redundant variations, which can disrupt the original balance of the data distribution and slightly impair the downstream classification.



(a) RealWorld-HAR (b) PAMAP2

Fig. 5: The influence of augmentation ratio on the performance.

F. Condition Influence on Generated Samples

We speculate that the downstream classifiers will benefit from the dataset augmented by interpolation in the condition space because of the tractable high-level feature augmentation and the realism of the synthetic samples. This is based on the presumption that the conditions have strong control over the generation process of the diffusion model, so that it produces synthetic samples with statistical metrics aligned with the input conditions. In this section, we empirically evaluate the extent to which the conditions influence the statistical metrics of the generated samples.

The conditional diffusion model was trained with a batch size of 100 and an initial learning rate of 10^{-4}, employing the Adam optimizer and a cosine annealing scheduler for dynamic learning rate adjustment throughout the training period. In this analysis, we profile the normalized statistical metric error on the generated samples across various training epochs. At each profiled epoch, we generate the number of interpolated samples equivalent to the size of the training set. Then we calculate the Mean Squared Error (MSE) on the normalized values of the statistical metrics of generated samples and the metric values in their input condition. As shown in Table II, the error decreases and stabilizes as training progresses, underscoring that the generated samples increasingly align with the input conditions over time. The results demonstrate that the statistical metrics as conditions successfully gain stronger control over the generated sample as the training proceeds

TABLE II: Normalized Statistical Metric Error on Testing Set at Different Epochs (Values of errors are scaled by 10^{-3})

	Epoch					
	100	200	400	600	800	1000
RealWorld-HAR	6.348	5.264	4.172	3.787	3.857	3.984
PAMAP2	7.421	5.101	4.219	3.650	3.120	3.297

V. Conclusion

In this paper, we introduced a novel data augmentation method to enhancing HAR through condition space interpola-

tion in a conditional diffusion model. One key innovation lies in utilizing statistical metrics as conditions for controlling the generation process within the diffusion model. To augment the original dataset, we interpolate the metric values of samples within the same class, and use the interpolated metrics to steer the diffusion model to produce samples that exhibit both realism and varied variance. Our experiments highlight the effectiveness of our method over traditional time-series augmentations, other interpolation-based strategies, and the conventional conditional generative model-based augmentation. This research offers a new data augmentation strategy that can advance the robustness of HAR in practical application scenarios.

ACKNOWLEDGMENT

Research reported in this paper was sponsored in part by DEVCOM ARL under Cooperative Agreement W911NF- 17-2-0196, NSF CNS 20-38817, and the Boeing Company. It was also supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[2] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," *arXiv preprint arXiv:1702.05538*, 2017.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[4] X. Liu, Y. Zou, L. Kong, Z. Diao, J. Yan, J. Wang, S. Li, P. Jia, and J. You, "Data augmentation via latent space interpolation for image classification," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 728–733.

[5] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," in *International conference on machine learning*. PMLR, 2013, pp. 552–560.

[6] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[7] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in human behavior*, vol. 15, no. 5, pp. 571–583, 1999.

[8] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[9] 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.

[10] F. Gu, M.-H. Chung, M. Chignell, S. Valaee, B. Zhou, and X. Liu, "A survey on deep learning for human activity recognition," *ACM Comput. Surv.*, vol. 54, no. 8, oct 2021. [Online]. Available: https://doi.org/10.1145/3472290

[11] T. Wang, J. Li, R. Wang, D. Kara, S. Liu, D. Wertheimer, A. Virosi Martin, R. Ganti, M. Srivatsa, and T. Abdelzaher, "Sudokusens: Enhancing deep learning robustness for iot sensing applications using a generative approach," *arXiv preprint arXiv:2402.02275*, 2024.

[12] R. Presotto *et al.*, "Collaborative approaches for sensor-based human activity recognition in data scarcity scenarios," 2023.

[13] T. Wang, D. Kara, J. Li, S. Liu, T. Abdelzaher, and B. Jalaian, "The methodological pitfall of dataset-driven research on deep learning: An iot example," in *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*. IEEE, 2022, pp. 1082–1087.

[14] X. Li, J. Luo, and R. Younes, "Activitygan: Generative adversarial networks for data augmentation in sensor-based human activity recognition," in *Adjunct proceedings of the 2020 ACM international joint conference on pervasive and ubiquitous computing and proceedings of the 2020 ACM international symposium on wearable computers*, 2020, pp. 249–254.

[15] S. Huang, P.-Y. Chen, and J. McCann, "Diffar: adaptive conditional diffusion model for temporal-augmented human activity recognition," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023, pp. 3812–3820.

[16] 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 *Proceedings of the 19th ACM international conference on multimodal interaction*, 2017, pp. 216–220.

[17] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[18] L. Alawneh, T. Alsarhan, M. Al-Zinati, M. Al-Ayyoub, Y. Jararweh, and H. Lu, "Enhancing human activity recognition using deep learning and time series augmented data," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–16, 2021.

[19] S. Shao and V. Sanchez, "A study on diffusion modelling for sensor-based human activity recognition," in *2023 11th International Workshop on Biometrics and Forensics (IWBF)*, 2023, pp. 1–7.

[20] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[22] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 500–22 510.

[23] M. Jeong, H. Kim, S. J. Cheon, B. J. Choi, and N. S. Kim, "Diff-tts: A denoising diffusion model for text-to-speech," *arXiv preprint arXiv:2104.01409*, 2021.

[24] M. W. Lam, J. Wang, D. Su, and D. Yu, "Bddm: Bilateral denoising diffusion models for fast and high-quality speech synthesis," *arXiv preprint arXiv:2203.13508*, 2022.

[25] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 10, 2022, pp. 11 020–11 028.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[27] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9.

[28] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*, 2012, pp. 108–109.

[29] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 351–360.