
UniMTS: Unified Pre-training for Motion Time Series

Xiyuan Zhang
UC San Diego
xiyuanzh@ucsd.edu

Diyang Teng
Qualcomm
diyateng@qti.qualcomm.com

Ranak Roy Chowdhury
UC San Diego
rrchowdh@ucsd.edu

Shuheng Li
UC San Diego
shl060@ucsd.edu

Dezhi Hong*
Amazon
hondezhi@amazon.com

Rajesh K. Gupta
UC San Diego
rgupta@ucsd.edu

Jingbo Shang
UC San Diego
jshang@ucsd.edu

Abstract

Motion time series collected from low-power, always-on mobile and wearable devices such as smartphones and smartwatches offer significant insights into human behavioral patterns, with wide applications in healthcare, automation, IoT, and AR/XR. However, given security and privacy concerns, building large-scale motion time series datasets remains difficult, hindering the development of pre-trained models for human activity analysis. Typically, existing models are trained and tested on the same dataset, leading to poor generalizability across variations in device location, device mounting orientation, and human activity type. In this paper, we introduce UniMTS¹, the first unified pre-training procedure for motion time series that generalizes across diverse device latent factors and activities. Specifically, we employ a contrastive learning framework that aligns motion time series with text descriptions enriched by large language models. This helps the model learn the semantics of time series to generalize across activities. Given the absence of large-scale motion time series data, we derive and synthesize time series from existing motion skeleton data with all-joint coverage. We use spatio-temporal graph networks to capture the relationships across joints for generalization across different device locations. We further design rotation-invariant augmentation to make the model agnostic to changes in device mounting orientations. Our model shows exceptional generalizability across 18 motion time series classification benchmark datasets, outperforming the best baselines by **340%** in the zero-shot setting, **16.3%** in the few-shot setting, and **9.2%** in the full-shot setting.

1 Introduction

Recognition of human motion using time series from mobile and wearable devices, such as accelerations and angular velocities, is widely adopted as key context information for various applications from health condition monitoring [4], sports activity analysis [1] to user habit studies [50]. Compared with vision-based approaches, methods based on motion sensor time series offer more energy-efficient and cost-effective solutions with enhanced privacy protection [54], making them preferable.

*Work unrelated to Amazon.

¹Code is available on Github: <https://github.com/xiyuanzh/UniMTS>. Model is available on Hugging Face: <https://huggingface.co/xiyuanzh/UniMTS>.

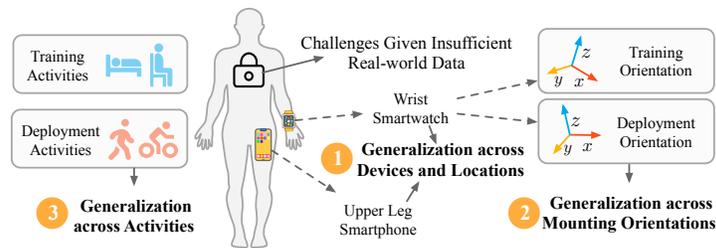


Figure 1: Our framework addresses three key generalization challenges (variation in device location, orientation, and activity) where existing methods fall short.

While valuable, collecting motion time series data at large scale remains challenging due to security or privacy concerns. Labeling motion time series proves even more difficult as such data cannot be easily interpreted by humans for post annotation. This results in data insufficiency that impedes development of supervised learning methods. In other fields such as natural language processing [39, 56] and computer vision [44, 30], pre-trained foundation models have shown remarkable performance in such settings with insufficient data. However, in the motion time series domain, lack of comprehensive datasets and an effective pre-training task makes it difficult to similarly develop pre-trained models that can operate with limited data. Typically, existing models perform training and testing on the same dataset, and struggle to generalize across different datasets given the following three unique challenges within the motion time series problem domain.

We summarize these three unique generalization challenges in Figure 1. First of all, variation in device placement during deployment poses a significant issue; for instance, data from a smartwatch on the wrist vary considerably from data gathered from a smartphone near the upper leg. Therefore, models trained on data from one body location can barely generalize to others during the testing phase. Secondly, devices can experience arbitrary orientations during data collection, making it difficult for models trained on specific device orientations to adapt to new ones during deployment. Thirdly, different motion time series datasets can be focused on different types of human activities. For example, some datasets aim to identify stationary activities such as lying or sitting, while others concentrate on dynamic movements such as walking or cycling. Models trained on specific types of activities typically struggle to generalize to new activities introduced by other datasets.

We introduce UniMTS, the first Unified pre-trained model for Motion Time Series to address all the above three generalization issues, achieving state-of-the-art zero-shot and fine-tuning performance. UniMTS follows a contrastive learning framework that aligns motion time series with LLM-enriched textual descriptions to learn the time series semantics for *activity* generalization. To prepare large-scale motion time series for pre-training, we synthesize these time series based on existing extensive motion skeleton data [19] with comprehensive coverage of different body locations. We model these synthesized time series using graph networks to capture the spatio-temporal relationships across devices for *location* generalization. We further implement rotation-invariant augmentation to ensure the model’s robustness to any device *orientation* during testing.

We summarize our primary contributions as follows:

- We introduce the first unified pre-training procedure for motion time series, UniMTS, which successfully generalizes to various device locations, device orientations and activities.
- We design a contrastive learning framework to align motion time series with corresponding semantic meanings for activity generalization. For device location generalization, we propose to synthesize motion time series covering various body locations and model their spatio-temporal correlations using graph convolutional neural networks. We also design rotation-invariant augmentation to make the model agnostic to different device orientations.
- Our pre-trained model demonstrates state-of-the-art performance across 18 real-world motion time series benchmark datasets, notably with performance improvement of **340%** in the zero-shot setting, **16.3%** in the few-shot setting, and **9.2%** in the full-shot setting, compared with the respective best-performing baselines.

2 Related Work

Conventional motion time series classification approaches train a dedicated classifier for each dataset, and can be categorized into statistical feature extraction methods [15] and deep learning

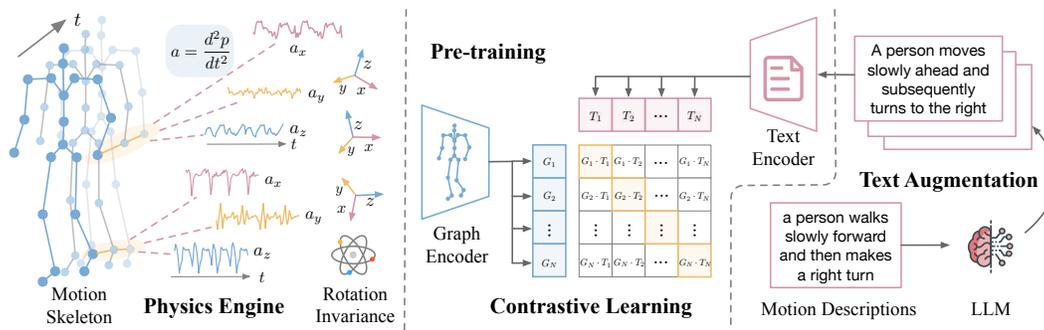


Figure 2: UniMTS pre-training framework: The physics engine computes motion time series for each joint based on motion skeleton data and enhances time series through rotation-invariant augmentation. During pre-training, we adopt contrastive learning to align motion time series encoded by graph convolutional neural networks with corresponding text descriptions augmented by an LLM.

methods, including convolutional neural networks (MA-CNN [45], SenseHAR [23], Rocket [12]), recurrent neural network (DeepConvLSTM [40]), and the attention mechanism based models (AttnSense [36], THAT [29]). Recently, IMUGPT [28, 27] generates motion sequences given activity textual descriptions and trains conventional classification models such as DeepConvLSTM [40], TimesNet [60], GPT4TS [72] and TEST [52] propose task-general time-series models for multiple tasks including classification. SHARE [71] presents a sequence-to-sequence framework that leverages shared structures of label names. However, these models perform training and testing on the same dataset, and cannot generalize across datasets.

Self-supervised motion time series representation learning methods first learn time series representations based on mask reconstruction (TST [68], TARNet [11], LIMU-BERT [62]), contrastive learning (TNC [55], TS-TCC [14], TS2Vec [67], TF-C [70], FOCAL [32], CL-HAR [42], DDLearn [43]) or other self-supervised learning objectives (BioBankSSL [66, 13, 10], Step2Heart [51]). Subsequently, they fine-tune classifier heads for specific downstream tasks. However, the representation learning and fine-tuning phases of these methods generally occur on the same or highly similar datasets, which continues to face challenges in generalization across diverse datasets.

Pre-trained models for motion time series are inspired by the recent success of large language or multimodal models. ImageBind [16] and IMU2CLIP [38] leverage recent large vision-language models [44] to learn a joint embedding across multiple modalities including motion time series and text. However, both ImageBind and IMU2CLIP are trained on motion time series collected from head-mounted devices [17], limiting their generalizability across different device locations and orientations. Furthermore, several studies have explored directly applying LLMs for motion time series classification. For example, HARGPT [24] processes raw motion time series through LLMs and incorporates role-play and chain-of-thought strategies for prompting. ContextGPT [3] designs prompt engineering approaches leveraging context information. However, since LLMs are not directly trained on raw motion time series, such methods require extensive context information that is not usually available, and struggle with accurately recognizing complex activities.

Other related works on motion classification include multimodal action recognition and domain adaptation methods. Multimodal action recognition such as the Ego4D [17] and Ego-Exo4D [18] benchmarks incorporates video and audio modalities, whereas we focus on a more energy-efficient and challenging scenario of action recognition based purely on motion time series. Domain adaptation methods mostly assume that source and target datasets share the same label names and have the same number of classes, such as cross-user domain adaptation and cross-dataset domain adaptation only for those common classes [22, 35, 21]. We aim for a more generic yet challenging generalization scenario where pre-training and downstream datasets share different label names.

3 Method

UniMTS takes a contrastive learning-based approach that aligns paired motion time series with text descriptions to enable activity generalization, as shown in Figure 2. We simulate motion time series from motion skeleton data (Section 3.1) and augment them for orientation generalization (Section 3.2).

We use graph encoder to model the simulated motion time series, capturing correlations among joints to generalize across different device locations (Section 3.3.1). To enhance semantics learning, we use large language models to augment text descriptions (Section 3.3.2).

3.1 Physics Engine for Motion Time Series Simulation

Motion skeleton data [19] describe the movements of human skeleton joints over time, containing positions and orientations for each joint. On the other hand, motion time series captured by physical sensors typically measure higher-order data such as accelerations and angular velocities. Consequently, we apply motion equations [65] to synthesize these time series of accelerations and angular velocities from motion skeleton data. More specifically, for each skeleton joint J_i , we input both positions $\mathbf{p}_{J_i, \mathcal{G}}$ (mapped from time domain \mathcal{T} to \mathbb{R}^3 , defined in global frame \mathcal{G}), and orientation quaternions $\mathbf{q}_{J_i, \mathcal{GL}}$ (mapped from time domain \mathcal{T} to the Special Orthogonal Group $\text{SO}(3)$, defined in Hamilton convention with subscript \mathcal{GL} representing a frame rotation from local frame \mathcal{L} to global frame \mathcal{G}). We drop the subscript \mathcal{G} and \mathcal{GL} from here on for simplicity of notation. Based on motion equations [65], we calculate velocities \mathbf{v}_{J_i} and accelerations \mathbf{a}_{J_i} by taking the first and second order derivatives of positions \mathbf{p}_{J_i} . These derivatives are then transformed from global frames to local frames using the corresponding orientation sequences \mathbf{q}_{J_i} . Similarly, angular velocities $\boldsymbol{\omega}_{J_i}$ are computed by taking the first order derivatives of orientation quaternions \mathbf{q}_{J_i} . Mathematically,

$$\mathbf{v}_{J_i}(t) = \mathbf{q}_{J_i}^*(t) \otimes \mathbf{p}'_{J_i}(t) \otimes \mathbf{q}_{J_i}(t), \quad (1)$$

$$\mathbf{a}_{J_i}(t) = \mathbf{q}_{J_i}^*(t) \otimes \mathbf{p}''_{J_i}(t) \otimes \mathbf{q}_{J_i}(t), \quad (2)$$

$$\boldsymbol{\omega}_{J_i}(t) = 2\mathbf{q}_{J_i}^*(t) \otimes \mathbf{q}'_{J_i}(t), \quad (3)$$

where \otimes and $*$ represent the quaternion multiplication operator and the quaternion conjugate.

Recognizing the inherent presence of noise carried by sensors in practice, the physics engine incorporates Gaussian noise with a zero mean into the simulated data. Representing the above motion time series as $\mathbf{x}_{J_i}(t)$, which can denote either $\mathbf{a}_{J_i}(t)$ (accelerations) or $\boldsymbol{\omega}_{J_i}(t)$ (angular velocities), the noisy time series $\tilde{\mathbf{x}}_{J_i}(t)$ are formulated as

$$\tilde{\mathbf{x}}_{J_i}(t) = \mathbf{x}_{J_i}(t) + \mathbf{n}_{J_i}(t), \mathbf{n}_{J_i}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}). \quad (4)$$

3.2 Rotation-Invariant Augmentation

A common limitation we have identified from prior studies that leads to their poor generalization is that they fail to consider the impact of latent device orientation factors on the motion time series. For example, end users can potentially wear devices in various orientations, such as with a phone facing towards or against the body in a pocket. Additionally, the software driver API for axis definition can be arbitrarily configured by the developers. For example, the iOS system defines acceleration in an opposite direction compared to the Android system². With the listed risk factors considered, we apply a data augmentation technique to simulate random orientations during pre-training, so that our learned model achieves rotation-invariance during deployment [7, 57, 61]. Specifically, during pre-training, for each iteration we sample a random rotation matrix for each joint J_i ,

$$\mathbf{R}_{J_i}^\delta \sim \text{Uniform}(\text{SO}(3)), \quad (5)$$

and compute the augmented time series $\hat{\mathbf{x}}_{J_i}^t$ at timesteps $t = 1, 2, \dots, T$ as

$$\hat{\mathbf{x}}_{J_i}^t = \mathbf{R}_{J_i}^\delta \tilde{\mathbf{x}}_{J_i}^t. \quad (6)$$

During one iteration, the same $\mathbf{R}_{J_i}^\delta$ is consistently applied to J_i for every time series and every timestep $t = 1, 2, \dots, T$. The rotation-invariant augmentation ensures that the simulated time series are adaptable to any downstream orientation, thereby enhancing the generalization capabilities.

3.3 Contrastive Learning

The physics engine generates sufficient motion time series data, which are subsequently encoded by graph networks and aligned with their corresponding text embeddings through contrastive learning.

²<https://github.com/tszheichoi/awesome-sensor-logger/blob/main/CROSSPLATFORM.md>

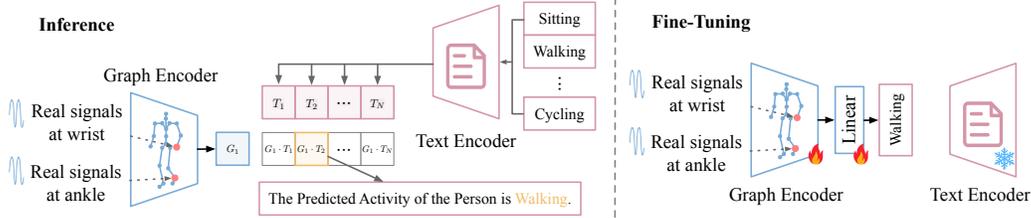


Figure 3: Inference (left) and fine-tuning (right) phases of UniMTS. We assign real signals to the nearest location in the skeleton graph. During inference, we compute the similarity score between the graph embedding and each label candidate, and predict the one with the highest score. During fine-tuning, we freeze the text encoder and update weights of the graph encoder and linear layer.

3.3.1 Graph Encoder

To capture the spatio-temporal correlations among different joints over time, we adopt spatio-temporal graph convolutional network [63] as our motion time series encoder. We denote the initial input graph representation as follows,

$$\mathcal{G} = (\mathcal{V} = \{\hat{\mathbf{x}}_{J_i}\}_{i=1}^V, \mathcal{E}_s = \{(\hat{\mathbf{x}}_{J_i}, \hat{\mathbf{x}}_{J_l}) | (J_i, J_l) \in \mathcal{H}\}, \mathcal{E}_t = \{(\hat{\mathbf{x}}_{J_i}^{t-1}, \hat{\mathbf{x}}_{J_i}^t)\}_{i=1, t=2}^{V, T}). \quad (7)$$

Nodes \mathcal{V} contain skeleton joints with features $\mathbf{X} \in \mathbb{R}^{C \times T \times V}$, where C, T, V represent the number of signal channels, temporal steps and joint nodes. Spatial edges \mathcal{E}_s connect adjacent nodes defined by the skeleton structure \mathcal{H} and temporal edges \mathcal{E}_t connect temporally adjacent frames.

In practice, devices may not cover the complete joints but are rather positioned at arbitrary subsets of the complete joints. To simulate this, during each pre-training iteration, we randomly select a subset of joints and mask data from the remaining joints with zeros. We denote the mask at one iteration as $\mathbf{M} \in \mathbb{R}^{C \times T \times V}$, where $\mathbf{M}_i \in \mathbb{R}^{C \times T}$ is $\mathbf{1}$ if joint J_i is selected, and $\mathbf{M}_i = \mathbf{0}$ if joint J_i is masked:

$$\tilde{\mathbf{X}} = \mathbf{X} \odot \mathbf{M}, \quad (8)$$

The graph convolution network g_ϕ first computes the spatial output features as

$$\mathbf{X}_{\text{out}} = \sum_k^{K_s} \Phi_k(\tilde{\mathbf{X}}(\Lambda_k^{-\frac{1}{2}} \mathbf{A}_k \Lambda_k^{-\frac{1}{2}})), \quad (9)$$

where K_s denotes the spatial kernel size, \mathbf{A}_k^{il} represents whether node \mathbf{x}_{J_l} belongs to the spatial convolution sampling subset $\mathcal{S}_{J_i}^k$ of node \mathbf{x}_{J_i} , and $\Lambda_k^{ii} = \sum_l (\mathbf{A}_k^{il}) + \alpha$ represents the normalized diagonal matrix, with α set to 0.001 to prevent empty rows [63, 48]. $\Phi_k \in \mathbb{R}^{C' \times C \times 1 \times 1}$ represents weights of the 1×1 convolution operation with C' denoting output channel dimension. Following spatial convolution, we further perform $K_t \times 1$ temporal convolution on the spatial output features \mathbf{X}_{out} , similar to classical convolution operations, where K_t represents the temporal kernel size. The final graph representation $g_\phi(\mathbf{X})$ is derived by averaging features across both spatial and temporal dimensions with a graph average pooling layer at the end.

3.3.2 Text Encoder

To increase the diversity of paired text descriptions in the pre-training motion corpus [19], we apply large language models (GPT-3.5) to augment original motion text descriptions with the following prompt template: *The following one or multiple descriptions are describing the same human activities: <motion descriptions>. Generate k paraphrases to describe the same activities.*

We denote the original text descriptions combined with the LLM-augmented ones as \mathbf{Y} . We encode them using the same text encoder f_θ as CLIP [44], utilizing its pre-trained weights for initialization.

3.3.3 Training and Inference

During pre-training, we maximize the similarities of paired simulated motion time series and text descriptions through contrastive learning:

$$\mathcal{L}_{ctr} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\text{sim}(g_\phi(\mathbf{X}_i), f_\theta(\mathbf{Y}_i)))^{\frac{1}{\gamma}}}{\sum_{k=1}^B \exp(\text{sim}(g_\phi(\mathbf{X}_i), f_\theta(\mathbf{Y}_k)))^{\frac{1}{\gamma}}}, \quad (10)$$

where B, γ represent batch size and temperature parameter that controls distribution concentrations, and sim represents similarity score computed as inner product:

$$\text{sim}(g_\phi(\mathbf{X}_i), f_\theta(\mathbf{Y}_i)) = \langle g_\phi(\mathbf{X}_i), f_\theta(\mathbf{Y}_i) \rangle. \quad (11)$$

We pre-train the graph and text encoders using simulated motion time series and augmented text descriptions. During inference, we evaluate the model on real-world motion time series, as illustrated in the left part of Figure 3. For the text encoder, we input all label candidates. For the graph encoder, we assign real motion time series to the nearest joint in the skeleton graph and assign zeros to the remaining joints. The random mask \mathbf{M} during pre-training emulates the zero-masking process. We compute the similarity score between the graph embedding with text embedding from each label candidate, and choose the label with the highest similarity score as the predicted activity.

We can further fine-tune the pre-trained model on downstream real-world data, as depicted in the right part of Figure 3. Specifically, we freeze the text encoder f_θ and update weights of the graph encoder g_ϕ followed by a linear classifier h_ψ . Following the same process as inference, we assign the real motion time series to the nearest joint in the skeleton graph and assign zeros to the remaining joints to construct the graph input representation \mathbf{X} . We fine-tune the model using \mathbf{X} and one-hot encoded labels \mathbf{z} with D classes based on cross-entropy loss, where $\sigma(\cdot)$ represents the softmax operation:

$$\mathcal{L}_{ce} = -\frac{1}{B} \sum_{i=1}^B \sum_{j=1}^D \mathbf{z}_{ij} \log(\sigma(h_\psi(g_\phi(\mathbf{X}_i)))_j). \quad (12)$$

We report both zero-shot and fine-tuning performance in the subsequent experiment section.

4 Experiments

4.1 Datasets and Experimental Setting

We simulate motion time series from existing motion skeleton dataset HumanML3D [19], which contain both motion skeleton data and corresponding text descriptions as detailed in Section A.1 in Appendix. We further augment the text descriptions as described in Section 3.3.2.

We evaluate on the most extensive motion time series classification benchmark to date, comprising 18 real-world datasets that cover diverse activities. These datasets are collected from various body locations such as head, chest, back, arm, wrist, waist, hip, leg, knee and ankle. We categorize these datasets into three difficulty levels: (1) easy level (with fewer than 10 activities): Opportunity [47], UCI-HAR [2], MotionSense [37], w-HAR [5], Shoab [49], HAR70+ [58], RealWorld [53], TNDA-HAR [64]; (2) medium level (with 10 to 20 activities): PAMAP2 [46], USC-HAD [69], Mhealth [4], Harth [33], UT-Complex [50], Wharf [6], WISDM [59], DSADS [1]; (3) hard level (with more than 20 activities): UTD-MHAD [8], MMAct [26]. We provide the specific number of activities for each dataset in Table 1 and Table 2, and detail their collection settings in Section A.2 in Appendix.

We re-sample the real-world test data to the same sampling frequency as the simulation data (20 Hz), and apply normalization to ensure consistency in unit measurements, e.g., standardizing accelerations to m/s^2 . We pre-train UniMTS using Adam optimizer [25] with a learning rate of 0.0001 on a single NVIDIA A100 GPU. The pre-training process consumes approximately 13 GB of memory given a batch size of 64. For text augmentation, we prompt GPT-3.5 (“gpt-3.5-turbo”) to generate $k = 3$ paraphrases. During each iteration, we randomly generate the mask \mathbf{M} by selecting 1 to 5 joints and mask the remaining joints as zeros. We adopt learnable temperature parameter γ initialized from CLIP. We evaluate the models using accuracy, macro-F1 and the top-2 retrieval performance R@2.

4.2 Zero-Shot Results

We pre-train UniMTS exclusively on simulated data and evaluate on 18 real-world motion time series classification benchmark datasets. We compare UniMTS against classification models with zero-shot capabilities: ImageBind [16], IMU2CLIP [38], IMUGPT [28] and HARGPT [24]. We also input the 2D visualizations of motion time series to pre-trained vision-language model LLaVA [30] for comparison. We detail the configurations of baselines in Section A.3 in Appendix. As shown in Table 1, UniMTS significantly outperforms all baselines in the zero-shot setting. We also apply the Wilcoxon-signed rank test with Holm’s α (5%) following previous works [20, 71]. The Wilcoxon-signed rank test indicates that the improvement of UniMTS compared with all the baselines is

Table 1: Zero-Shot performance. We bold the **best** and underline the second best. UniMTS performs the best compared with both baselines and our model ablations. The last column shows the average performance across 18 datasets with standard deviation.

Dataset	Metrics	Opportunity	UCI-HAR	MotionSense	w-HAR	Shoahib	HAR70+	RealWorld	TNDA-HAR	PAMAP2	USC-HAD	Mhealth	Hearth	UT-Complex	Wharf	WISDM	DSADS	UTD-MHAD	MMAct	Average
Number of Classes		4	6	6	7	7	7	8	8	12	12	12	12	13	14	18	19	27	35	
Level		Easy								Medium								Hard	Avg	
ImageBind [16]	Acc	<u>50.2</u>	14.1	18.1	13.1	19.4	0.0	18.9	19.3	14.6	11.8	7.9	9.8	9.7	3.2	7.1	2.1	3.3	2.9	12.5 _(11.0)
	F1	<u>30.0</u>	5.0	16.4	8.6	15.5	0.0	10.1	13.7	8.1	7.3	1.7	6.1	6.3	1.9	4.5	1.2	1.6	1.5	7.8 _(7.2)
	R@2	83.6	21.4	42.8	54.1	35.5	0.2	23.9	32.8	15.6	25.0	21.3	22.5	17.8	6.8	13.7	6.9	5.1	3.7	24.0 _(20.0)
IMU2CLIP [38]	Acc	25.9	17.8	15.5	6.6	16.7	5.6	6.1	8.5	1.9	12.8	7.9	2.5	7.3	1.4	4.3	4.6	3.7	6.4	8.6 _(6.4)
	F1	10.3	11.4	8.6	3.3	9.2	1.8	4.5	4.2	1.1	9.3	1.3	1.1	3.4	0.7	2.7	1.1	2.1	1.6	4.3 _(3.6)
	R@2	65.5	35.1	39.1	19.7	34.6	22.2	19.7	22.4	9.8	19.8	13.4	4.8	16.3	5.5	9.6	8.3	6.5	10.1	20.1 _(14.9)
IMUGPT [28]	Acc	10.1	1.1	11.8	67.2	12.4	0.0	16.9	14.3	8.9	6.0	9.8	4.8	11.6	2.7	8.3	7.5	3.7	2.0	11.1 _(14.4)
	F1	10.4	0.3	3.5	38.8	6.2	0.0	4.0	6.1	1.5	6.9	2.5	1.9	8.3	1.8	6.6	2.0	0.3	0.7	5.7 _(8.6)
	R@2	33.7	18.2	40.4	67.2	32.1	0.0	33.7	28.5	19.3	31.8	19.5	27.8	17.4	17.7	13.9	14.6	8.8	5.1	23.9 _(14.9)
HARGPT [24]	Acc	28.8	15.0	11.6	4.9	21.0	34.3	12.7	13.7	11.1	9.5	10.4	28.8	7.5	5.9	5.5	5.8	3.3	2.3	12.9 _(9.2)
	F1	17.3	12.7	5.6	3.1	12.4	10.6	5.3	5.4	2.1	3.6	7.4	7.5	4.4	1.4	3.5	3.4	1.5	1.1	6.0 _(4.4)
	R@2	47.0	31.4	35.9	11.5	38.6	51.9	31.7	25.2	23.0	17.6	27.4	48.6	16.5	11.4	11.8	12.1	9.3	5.0	25.3 _(14.2)
LLaVA [30]	Acc	40.1	16.3	22.8	0.0	16.7	10.3	16.8	12.9	10.3	11.1	18.9	16.3	2.1	3.6	5.6	5.3	3.7	4.0	12.0 _(9.4)
	F1	14.3	6.5	6.2	0.0	4.8	3.7	3.7	2.9	1.6	2.6	7.4	5.0	0.6	0.7	0.6	0.5	0.3	0.2	3.4 _(3.5)
	R@2	67.6	34.6	34.4	0.0	33.3	43.4	28.4	25.2	18.7	19.6	33.5	16.4	9.0	3.6	10.6	10.5	7.4	7.3	22.4 _(16.5)
UniMTS	Acc	45.9	35.2	45.2	<u>59.0</u>	63.6	68.2	<u>43.6</u>	59.1	47.2	30.5	70.7	68.9	34.8	<u>18.2</u>	<u>27.8</u>	31.5	22.8	10.2	43.5 _(17.9)
	F1	42.2	<u>22.0</u>	<u>33.7</u>	42.9	57.2	34.8	<u>36.7</u>	53.7	43.6	27.8	61.8	41.1	29.2	<u>13.7</u>	<u>25.5</u>	23.7	18.5	10.0	34.3 _(14.2)
	R@2	<u>80.0</u>	<u>53.1</u>	<u>57.2</u>	<u>60.7</u>	82.1	86.6	<u>64.0</u>	<u>77.5</u>	63.2	<u>45.4</u>	78.7	85.0	44.2	<u>38.6</u>	47.1	46.0	32.6	<u>18.7</u>	58.9 _(19.3)
w/o rot aug	Acc	37.7	18.6	25.1	36.1	19.4	53.4	55.5	35.6	32.5	20.7	27.4	59.4	9.9	3.6	13.5	21.5	<u>13.5</u>	4.3	27.1 _(16.3)
	F1	30.4	8.2	11.2	26.4	13.5	27.7	41.1	29.6	28.3	10.5	24.0	20.5	4.9	4.8	10.6	13.4	<u>7.2</u>	4.7	17.6 _(10.7)
	R@2	74.3	40.1	64.3	52.5	40.1	76.5	76.3	54.4	<u>48.8</u>	29.7	49.4	61.2	28.5	6.8	23.3	34.0	<u>32.1</u>	7.6	44.4 _(20.8)
w/o text aug	Acc	52.7	<u>36.3</u>	<u>43.4</u>	57.4	<u>55.6</u>	<u>61.0</u>	40.7	<u>40.9</u>	<u>38.4</u>	<u>29.6</u>	<u>59.2</u>	<u>62.8</u>	<u>30.3</u>	28.2	31.3	<u>29.3</u>	6.5	12.6	<u>39.8</u> _(15.8)
	F1	<u>39.4</u>	21.3	34.7	<u>41.4</u>	<u>49.5</u>	<u>32.8</u>	29.7	<u>32.7</u>	<u>33.9</u>	21.6	<u>49.0</u>	<u>26.7</u>	<u>21.2</u>	19.6	27.2	<u>22.2</u>	4.7	10.3	<u>28.8</u> _(11.6)
	R@2	70.9	39.6	<u>63.7</u>	57.4	<u>78.7</u>	<u>77.4</u>	60.4	<u>63.5</u>	<u>48.8</u>	49.1	<u>63.4</u>	<u>77.9</u>	<u>41.4</u>	43.2	<u>45.1</u>	<u>41.7</u>	10.7	23.2	<u>53.1</u> _(18.1)
w/o graph	Acc	41.4	37.6	18.9	23.0	50.9	18.8	42.7	33.8	30.0	22.4	28.7	34.8	23.4	0.5	12.4	19.8	1.9	<u>11.2</u>	25.1 _(13.4)
	F1	20.5	28.5	21.6	17.9	38.6	9.1	23.7	28.9	23.6	<u>23.0</u>	19.3	15.3	11.5	1.1	8.5	16.0	2.7	7.7	17.6 _(9.5)
	R@2	63.7	66.7	54.2	34.4	66.1	35.3	51.1	62.1	41.8	41.7	43.9	51.6	35.6	8.2	23.7	29.1	4.7	18.0	40.7 _(18.4)

statistically significant, with p-values significantly lower than 0.05 (e.g., p-value = 8×10^{-6} for ImageBind, which has the highest F1 score among the baselines).

Compared with UniMTS, ImageBind and IMU2CLIP are trained on data from single location (head-mounted devices), limiting their generalization to data collected from other locations. IMUGPT struggles to generalize across datasets featuring different activities and requires individual training for each downstream dataset. Both HARGPT and LLaVA focus on simple and easily distinguishable activities as these language or vision models are not originally trained on motion time series, and they also require careful prompt designs. Another limitation for all the above models is that they do not generalize across device orientations. In contrast, UniMTS shows remarkable generalizability to various downstream device locations, orientations and activities, achieving state-of-the-art performance. We also compare with a few ablations of UniMTS as illustrated in the Ablation Study section.

4.3 Few-Shot Fine-tuning Results

Apart from the zero-shot setting, we provide a few real samples for each activity and fine-tune UniMTS and the baselines. More specifically, we provide 1, 2, 3, 5, 10 samples for each activity and compare UniMTS against ImageBind [16], IMU2CLIP [38], IMUGPT [28], GPT4TS [72], BioBankSSL [66] and a randomly initialized model with the same model architecture as UniMTS (referred to as Random). We report both the mean and the standard deviation in Figure 4. UniMTS also demonstrates state-of-the-art performance in the few-shot fine-tuning setting, showing the effectiveness of pre-training. Following the same Wilcoxon-signed rank test as in the zero-shot setting, we observe p-values far below 0.05 (e.g., p-value = 2×10^{-25} for the best-performing baseline ImageBind), indicating the statistical significance of our improvement.

4.4 Full-Shot Results

We also compare the full-shot performance where UniMTS and the baselines are fine-tuned or trained using all the available training samples of the downstream datasets. We compare UniMTS with pre-trained models (ImageBind [16], IMU2CLIP [38]), self-supervised models (TST [68], TARNet [11],

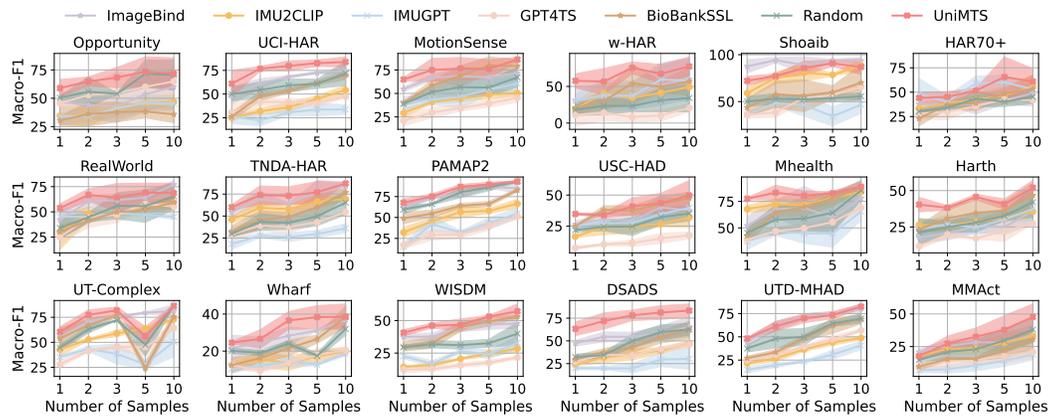


Figure 4: Few-shot fine-tuning results. UniMTS consistently outperforms both baselines and our model ablation. We repeat 3 runs and report both mean and standard deviation.

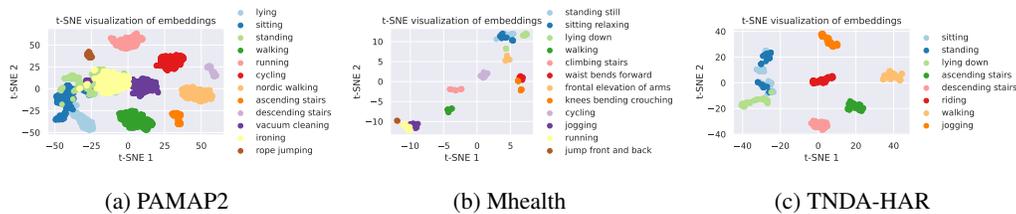


Figure 5: T-SNE visualizations show that signal clusters align with their semantic meanings.

TS2Vec [67], BioBankSSL [66]), and conventional models (DeepConvLSTM [40], MA-CNN [45], XGBoost [9], THAT [29], IMUGPT [28], TimesNet [60], GPT4TS [72], SHARE [71]). Baselines are detailed in Section A.3 in Appendix. We also compare pre-trained UniMTS with a randomly initialized UniMTS (referred to as Random). As shown in Table 2, UniMTS also demonstrates state-of-the-art performance in the full-shot setting, outperforming pre-trained, self-supervised and conventional models. Due to space limit, we report baselines before 2021 in Table 3 in Appendix. Following the same Wilcoxon-signed rank test, we observe p-values far below 0.05 (e.g., p-value = 0.018 for the best-performing baseline), indicating the statistical significance of our improvement. UniMTS also demonstrates space and time efficiency, as detailed in Section A.5 in Appendix.

4.5 Ablation Study

In the zero-shot setting, we compare UniMTS with a few ablations by removing rotation-invariant augmentation (w/o rot aug), removing text augmentation (w/o text aug) and by replacing the graph encoder with a CNN-based encoder that directly concatenates joints without modeling their spatial relationships (w/o graph). We can observe in Table 1 that the performance declines after removing each of the above components, verifying their respective importance in improving generalization across locations (graph encoder), orientations (rotation-invariant augmentation) and activities (text augmentation). We also compare the pre-trained UniMTS with randomly initialized UniMTS in both few-shot and full-shot settings. As shown in Figure 4 and Table 2, pre-trained UniMTS consistently outperforms randomly initialized UniMTS, highlighting the benefits of pre-training.

4.6 Case Study

UniMTS’s time series embeddings align with corresponding semantic meanings. As shown in Figure 5, the t-SNE visualizations of UniMTS’s time series embeddings form distinguishable clusters that align with their semantic meanings. Notably, UniMTS is only pre-trained on the simulated data but its embeddings for real-world data closely align with the semantic space, which again demonstrates our model’s zero-shot generalization due to contrastive learning. For example, in Figure 5a, stationary activities such as lying and sitting group together; light-movement activities such as standing, ironing, and vacuum cleaning are close to each other; while high-intensity activities such as running and cycling cluster closer in the embedding space.

Table 2: Full-Shot performance. We bold the **best** and underline the second best. UniMTS performs the best compared with both pre-trained, self-supervised and conventional models. The last column shows the average performance across 18 datasets with standard deviation.

Dataset	Metrics	Opportunity	UCI-HAR	MotionSense	w-HAR	Shogib	HAR70+	RealWorld	TNDA-HAR	PAMAP2	USC-HAD	Mhealth	Harth	UT-Complex	Wharf	WISDM	DSADS	UTD-MHAD	MMAct	Average
Number of Classes		4	6	6	7	7	7	8	8	12	12	12	12	13	14	18	19	27	35	
Level		Easy								Medium								Hard		Avg
ImageBind [16]	Acc	82.8	90.9	94.0	95.1	98.5	91.8	<u>83.8</u>	95.0	94.3	52.1	82.9	93.0	97.0	71.4	77.2	90.0	74.0	65.8	85.0 _(12.2)
	F1	84.6	90.7	93.1	61.9	98.5	67.7	86.1	94.9	94.6	52.2	83.2	62.7	96.9	43.3	76.8	89.3	71.6	58.3	78.1 _(16.5)
IMU2CLIP [38]	Acc	71.9	87.1	79.3	88.5	98.2	87.8	79.9	97.1	90.7	55.7	94.5	93.9	90.6	52.3	62.7	88.7	62.8	66.9	80.5 _(14.3)
	F1	70.0	87.0	77.2	75.1	98.1	64.4	80.5	97.1	91.2	52.0	95.0	65.7	90.2	35.6	62.8	88.2	59.6	60.3	75.0 _(17.0)
TST [68]	Acc	91.1	91.6	91.4	75.4	73.5	95.8	68.9	94.1	88.9	<u>66.2</u>	86.0	<u>98.8</u>	92.9	59.1	70.2	80.0	61.4	54.5	79.9 _(13.6)
	F1	91.1	91.4	89.6	79.3	72.3	62.9	67.2	94.1	89.8	60.9	83.8	<u>72.7</u>	92.8	34.5	69.1	76.1	59.6	48.4	74.2 _(16.3)
TARNet [11]	Acc	82.6	91.1	72.0	78.7	67.9	97.2	60.9	92.5	92.4	57.6	82.9	94.7	94.4	63.2	58.5	30.7	78.1	56.7	75.1 _(17.5)
	F1	83.4	91.2	70.0	41.0	66.1	76.9	54.4	92.4	90.6	48.7	77.5	47.8	94.4	38.1	58.5	23.0	75.5	55.9	65.9 _(20.5)
TS2Vec [67]	Acc	80.7	92.1	94.2	100.0	87.4	98.4	74.6	95.9	91.8	56.1	93.3	98.2	98.5	73.2	69.5	84.3	80.0	53.4	84.5 _(13.9)
	F1	83.3	92.1	93.9	100.0	87.2	88.1	68.8	95.9	92.9	53.6	93.8	58.3	98.5	47.9	68.8	82.3	76.7	53.4	79.8 _(16.7)
BioBankSSL [66]	Acc	85.9	92.7	99.4	93.4	<u>99.1</u>	93.2	75.5	89.4	87.3	65.8	<u>95.1</u>	98.0	96.4	<u>81.4</u>	83.3	65.8	<u>80.5</u>	62.5	85.8 _(11.5)
	F1	86.1	92.9	99.2	81.3	<u>99.1</u>	85.9	68.0	89.6	89.3	72.6	94.0	68.1	96.4	62.7	82.9	83.1	<u>78.4</u>	58.4	82.7 _(12.1)
THAT [29]	Acc	83.1	86.8	87.9	77.1	92.3	95.8	67.7	97.7	<u>96.5</u>	53.9	89.0	97.6	86.9	60.0	60.3	82.1	71.2	55.0	80.1 _(14.7)
	F1	84.5	86.7	86.5	65.0	92.2	73.9	62.9	97.7	96.7	55.0	89.9	71.9	87.0	29.5	60.9	79.0	70.0	52.3	74.5 _(17.5)
IMUGPT [28]	Acc	84.8	87.0	86.2	85.3	61.7	<u>98.3</u>	62.2	87.7	85.6	41.1	71.3	98.3	86.5	54.6	67.8	71.9	57.7	51.9	74.4 _(16.3)
	F1	84.9	87.0	85.2	48.8	61.8	78.0	56.0	87.5	83.8	42.4	63.6	65.2	85.8	24.6	67.4	71.0	53.2	49.0	66.4 _(17.7)
TimesNet [60]	Acc	80.0	88.5	90.5	88.5	82.4	97.5	65.4	93.2	88.0	58.3	81.7	97.7	92.7	41.4	67.4	77.3	62.3	50.9	78.0 _(16.2)
	F1	82.4	88.6	88.3	79.5	82.6	77.2	62.2	93.1	88.0	48.7	79.1	61.4	92.8	30.2	67.1	78.5	61.0	45.7	72.6 _(17.3)
GPT4TS [72]	Acc	83.6	88.4	92.3	70.5	73.5	97.8	66.5	95.7	92.6	61.7	87.8	97.2	95.3	51.8	63.8	79.9	66.1	52.6	78.7 _(15.2)
	F1	86.1	88.5	92.1	54.8	74.1	76.6	56.6	95.7	93.4	58.9	85.2	58.2	95.2	34.4	64.5	78.0	63.7	48.8	72.5 _(17.7)
SHARE [71]	Acc	<u>89.0</u>	<u>92.2</u>	<u>99.6</u>	96.7	77.5	97.7	66.0	95.9	94.1	72.9	97.0	99.1	<u>98.7</u>	63.2	77.0	<u>92.0</u>	73.0	62.1	85.8 _(13.2)
	F1	<u>90.2</u>	92.1	<u>99.6</u>	85.5	78.0	77.5	57.1	95.8	94.8	<u>66.7</u>	<u>97.1</u>	<u>74.0</u>	<u>98.8</u>	40.6	76.5	<u>91.9</u>	69.3	56.5	80.1 _(16.5)
UniMTS	Acc	88.2	92.1	99.8	98.4	99.7	96.9	84.0	99.6	98.0	58.3	97.0	98.0	99.1	82.7	<u>81.3</u>	94.3	85.6	77.1	90.6 _(10.6)
	F1	89.1	<u>92.2</u>	99.8	<u>98.8</u>	99.7	<u>87.9</u>	<u>81.2</u>	99.6	98.1	63.1	97.3	86.7	99.2	<u>51.7</u>	<u>80.9</u>	94.2	83.2	71.7	87.5 _(13.4)
Random	Acc	87.7	89.3	95.5	95.1	66.1	98.1	74.2	98.7	96.2	48.5	82.9	98.6	98.5	76.8	79.4	90.3	74.4	<u>73.9</u>	84.7 _(13.5)
	F1	88.9	89.4	95.5	60.1	65.0	85.9	74.0	<u>98.7</u>	<u>96.8</u>	51.9	78.2	69.1	98.5	50.8	79.3	90.1	69.7	<u>71.6</u>	78.5 _(15.1)

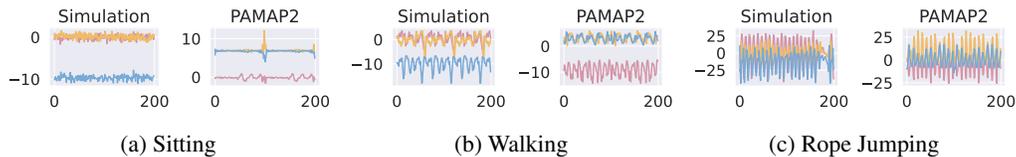


Figure 6: Simulated motion time series closely resemble patterns of the real PAMAP2 time series.

UniMTS generalizes well to new activities unseen in pre-training. To show UniMTS’s capability to generalize to new activities not seen during pre-training, we visualize the *zero-shot* performance for some example new activities in Figure 7. Compared with the best-performing baseline ImageBind, UniMTS shows significant performance improvement on these previously unseen activities, verifying the effectiveness of semantic generalization via contrastive learning.

Simulated data from our physics engine closely resemble real signal patterns. In Figure 6, we compare some example simulated data alongside their real-world counterparts. We show three example activities of different intensity levels (i.e., sitting, walking, rope jumping), where both simulated data and real-world data are near the wrist. We can observe that the patterns in simulated data closely resemble those of real data, in terms of both magnitude and frequency. Although the tri-axial distributions might differ, these variations are mostly due to orientation differences and are effectively managed by our rotation-invariant augmentation. We observe similar patterns between simulated data and real data for other device locations, as shown in Section A.4 in Appendix.

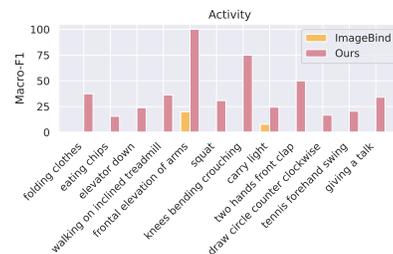


Figure 7: UniMTS shows significant performance improvement compared with the best baseline when evaluated on new activities not seen.

5 Conclusion and Discussion

Conclusion. In this paper, we present the first unified pre-training procedure, UniMTS, for motion time series classification. Our model is pre-trained only on physics-simulated data, and yet demonstrates remarkable generalization across diverse real-world motion time series datasets featuring different device locations, orientations and activities. The simulated data with all-joint coverage are augmented for rotation invariance and modeled by a graph encoder, improving generalization across various device factors. During pre-training, contrastive learning aligns time series with their semantic meanings to improve generalization across activities. Extensive evaluation in zero-shot, few-shot and full-shot settings consistently demonstrates the state-of-the-art performance of UniMTS.

Limitation and Future Work. We acknowledge a few limitations which we leave as future work. (1) Simulated motion time series can only be approximations of real signals, which are usually collected near – rather than directly on – the body joints. For example, sensors on smartwatches collect data near the wrist, not on the wrist joint itself. We plan to incorporate random offset vectors to better simulate real-world signal variations near joints. (2) While our framework effectively addresses the classification task, we intend to extend its applicability to other motion time series tasks such as inertial navigation. (3) Our current pre-training utilizes existing motion datasets, and we plan to enrich our pre-training corpus with additional motion data extracted from large-scale video-based pose estimation. (4) We also plan to integrate our model with efficient inference optimization techniques such as quantization, pruning and distillation for deployment on edge devices.

Broad Impact. UniMTS is the first pre-trained motion time series classification model that generalizes to diverse downstream datasets, irrespective of device locations, orientations and activities. The primary societal concern centers around privacy as motion time series might reveal personal information, so we ensure strict privacy controls at the earliest stages of model development by pre-training exclusively on synthetic data. With UniMTS’s state-of-the-art performance in zero-shot, few-shot and full-shot settings, we believe it would bring broad, positive impact to the community.

6 Acknowledgement

Our work is supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

Our work is also sponsored in part by NSF CAREER Award 2239440, NSF Proto-OKN Award 2333790, as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

References

- [1] Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.
- [2] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, page 3, 2013.
- [3] Luca Arrotta, Claudio Bettini, Gabriele Civitarese, and Michele Fiori. Contextgpt: Infusing llms knowledge into neuro-symbolic activity recognition models. *arXiv preprint arXiv:2403.06586*, 2024.
- [4] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. mhealthdroid: a novel framework for agile development of mobile health applications. In *Ambient Assisted Living and Daily Activities: 6th International Work-Conference, IWAAL 2014, Belfast, UK, December 2-5, 2014. Proceedings 6*, pages 91–98. Springer, 2014.
- [5] Ganapati Bhat, Nicholas Tran, Holly Shill, and Umit Y Ogras. w-har: An activity recognition dataset and framework using low-power wearable devices. *Sensors*, 20(18):5356, 2020.
- [6] Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa, Tullio Vernazza, and Renato Zaccaria. Analysis of human behavior recognition algorithms based on acceleration data. In *2013 IEEE International Conference on Robotics and Automation*, pages 1602–1607. IEEE, 2013.
- [7] Sara Caramaschi, Gabriele B Papini, and Enrico G Caiani. Device orientation independent human activity recognition model for patient monitoring based on triaxial acceleration. *Applied Sciences*, 13(7):4175, 2023.
- [8] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International conference on image processing (ICIP)*, pages 168–172. IEEE, 2015.
- [9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [10] Yuanyuan Chen, Shing Chan, Derrick Bennett, Xiaofang Chen, Xianping Wu, Yalei Ke, Jun Lv, Dianjianyi Sun, Lang Pan, Pei Pei, et al. Device-measured movement behaviours in over 20,000 china kadoorie biobank participants. *International Journal of Behavioral Nutrition and Physical Activity*, 20(1):138, 2023.
- [11] Ranak Roy Chowdhury, Xiyuan Zhang, Jingbo Shang, Rajesh K Gupta, and Dezhi Hong. Tarnet: Task-aware reconstruction for time-series transformer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA*, pages 14–18, 2022.
- [12] Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.
- [13] Aiden Doherty, Dan Jackson, Nils Hammerla, Thomas Plötz, Patrick Olivier, Malcolm H Granat, Tom White, Vincent T Van Hees, Michael I Trenell, Christopher G Owen, et al. Large scale population assessment of physical activity using wrist worn accelerometers: the uk biobank study. *PLoS one*, 12(2):e0169649, 2017.
- [14] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021.

- [15] Davide Figo, Pedro C Diniz, Diogo R Ferreira, and Joao MP Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.
- [16] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.
- [17] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.
- [18] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024.
- [19] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022.
- [20] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.
- [21] Zhiqing Hong, Zelong Li, Shuxin Zhong, Wenjun Lyu, Haotian Wang, Yi Ding, Tian He, and Desheng Zhang. Crosshar: Generalizing cross-dataset human activity recognition via hierarchical self-supervised pretraining. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(2):1–26, 2024.
- [22] Rong Hu, Ling Chen, Shenghuan Miao, and Xing Tang. Swl-adapt: An unsupervised domain adaptation model with sample weight learning for cross-user wearable human activity recognition. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 37, pages 6012–6020, 2023.
- [23] Jeya Vikranth Jeyakumar, Liangzhen Lai, Naveen Suda, and Mani Srivastava. Sensehar: a robust virtual activity sensor for smartphones and wearables. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 2019.
- [24] Sijie Ji, Xinzhe Zheng, and Chenshu Wu. Hargpt: Are llms zero-shot human activity recognizers? *arXiv preprint arXiv:2403.02727*, 2024.
- [25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [26] Quan Kong, Ziming Wu, Ziwei Deng, Martin Klinkigt, Bin Tong, and Tomokazu Murakami. Mmact: A large-scale dataset for cross modal human action understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8658–8667, 2019.
- [27] Zikang Leng, Amitrajit Bhattacharjee, Hrudhai Rajasekhar, Lizhe Zhang, Elizabeth Bruda, Hyeokhyen Kwon, and Thomas Plötz. Imugpt 2.0: Language-based cross modality transfer for sensor-based human activity recognition. *arXiv preprint arXiv:2402.01049*, 2024.
- [28] Zikang Leng, Hyeokhyen Kwon, and Thomas Plötz. Generating virtual on-body accelerometer data from virtual textual descriptions for human activity recognition. In *Proceedings of the 2023 ACM International Symposium on Wearable Computers*, pages 39–43, 2023.
- [29] Bing Li, Wei Cui, Wei Wang, Le Zhang, Zhenghua Chen, and Min Wu. Two-stream convolution augmented transformer for human activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 286–293, 2021.

- [30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.
- [31] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019.
- [32] Shengzhong Liu, Tomoyoshi Kimura, Dongxin Liu, Ruijie Wang, Jinyang Li, Suhas Diggavi, Mani Srivastava, and Tarek Abdelzaher. Focal: Contrastive learning for multimodal time-series sensing signals in factorized orthogonal latent space. *Advances in Neural Information Processing Systems*, 36, 2024.
- [33] Aleksej Logacjov, Kerstin Bach, Atle Kongsvold, Hilde Bremseth Bårdstu, and Paul Jarle Mork. Harth: a human activity recognition dataset for machine learning. *Sensors*, 21(23):7853, 2021.
- [34] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.
- [35] Wang Lu, Jindong Wang, Yiqiang Chen, Sinno Jialin Pan, Chunyu Hu, and Xin Qin. Semantic-discriminative mixup for generalizable sensor-based cross-domain activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–19, 2022.
- [36] Haojie Ma, Wenzhong Li, Xiao Zhang, Songcheng Gao, and Sanglu Lu. Attnsense: Multi-level attention mechanism for multimodal human activity recognition. In *IJCAI*, pages 3109–3115, 2019.
- [37] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In *Proceedings of the international conference on internet of things design and implementation*, pages 49–58, 2019.
- [38] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Alireza Dirafzoon, Aparajita Saraf, Amy Bearman, and Babak Damavandi. Imu2clip: Multimodal contrastive learning for imu motion sensors from egocentric videos and text. *arXiv preprint arXiv:2210.14395*, 2022.
- [39] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.
- [40] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.
- [41] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016.
- [42] Hangwei Qian, Tian Tian, and Chunyan Miao. What makes good contrastive learning on small-scale wearable-based tasks? In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3761–3771, 2022.
- [43] Xin Qin, Jindong Wang, Shuo Ma, Wang Lu, Yongchun Zhu, Xing Xie, and Yiqiang Chen. Generalizable low-resource activity recognition with diverse and discriminative representation learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1943–1953, 2023.
- [44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [45] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D Lane, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. Multimodal deep learning for activity and context recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–27, 2018.

- [46] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pages 108–109. IEEE, 2012.
- [47] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczeck, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*, pages 233–240. IEEE, 2010.
- [48] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019.
- [49] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14(6):10146–10176, 2014.
- [50] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors*, 16(4):426, 2016.
- [51] Dimitris Spathis, Ignacio Perez-Pozuelo, Soren Brage, Nicholas J Wareham, and Cecilia Mascolo. Self-supervised transfer learning of physiological representations from free-living wearable data. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 69–78, 2021.
- [52] Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm’s ability for time series. *arXiv preprint arXiv:2308.08241*, 2023.
- [53] Timo Sztyler and Heiner 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)*, pages 1–9. IEEE, 2016.
- [54] Shuhan Tan, Tushar Nagarajan, and Kristen Grauman. Egodistill: Egocentric head motion distillation for efficient video understanding. *Advances in Neural Information Processing Systems*, 36:33485–33498, 2023.
- [55] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.
- [56] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [57] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana 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*, pages 216–220, 2017.
- [58] Astrid Ustad, Aleksej Logacjov, Stine Øverengen Trollebø, Pernille Thingstad, Beatrix Vereijken, Kerstin Bach, and Nina Skjæret Maroni. Validation of an activity type recognition model classifying daily physical behavior in older adults: the har70+ model. *Sensors*, 23(5):2368, 2023.
- [59] Gary M Weiss. Wisdm smartphone and smartwatch activity and biometrics dataset. *UCI Machine Learning Repository: WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set*, 7:133190–133202, 2019.
- [60] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The eleventh international conference on learning representations*, 2022.
- [61] Huatao Xu, Pengfei Zhou, Rui Tan, and Mo Li. Practically adopting human activity recognition. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

- [62] Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 220–233, 2021.
- [63] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [64] Yan Yan, Dali Chen, Yushi Liu, Jinjin Zhao, Bo Wang, Xuankun Wu, Xiaohao Jiao, Yuqian Chen, Huihui Li, and Xuchao Ren. Tnda-har, 2021.
- [65] Alexander D Young, Martin J Ling, and Damal K Arvind. Imusim: A simulation environment for inertial sensing algorithm design and evaluation. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 199–210. IEEE, 2011.
- [66] Hang Yuan, Shing Chan, Andrew P Creagh, Catherine Tong, Aidan Acquah, David A Clifton, and Aiden Doherty. Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *NPJ digital medicine*, 7(1):91, 2024.
- [67] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987, 2022.
- [68] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124, 2021.
- [69] Mi Zhang and Alexander A Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 1036–1043, 2012.
- [70] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022.
- [71] Xiyuan Zhang, Ranak Roy Chowdhury, Jiayun Zhang, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. Unleashing the power of shared label structures for human activity recognition. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23*, page 3340–3350, 2023.
- [72] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36, 2024.

A Appendix / supplemental material

A.1 Pre-training Datasets

HumanML3D [19] is a large-scale motion skeleton data consisting of 14,616 3D human motion skeletons spanning 28.59 hours. The average motion skeleton sequence length is 7.1 seconds. Paired with each motion skeleton sequence there is an average of 3 textual descriptions, resulting in a total of 44,970 textual descriptions with a vocabulary size of 5,371. The average and median lengths of these descriptions are 12 and 10 words. We further augment the textual descriptions using large language models as described in Section 3.3.2. All motion skeletons follow the skeleton structure of SMPL [34] with 22 joint nodes. Figure 8 provides an example skeleton of “a person waves his hands”.

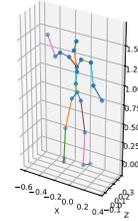


Figure 8: Skeleton of “waving hands”.

We also tried to incorporate additional motion skeleton datasets into pre-training, such as KIT-ML [41] and NTU RGB+D 120 [31]. However, these data are relatively less diverse in terms of both motion skeletons and textual descriptions. We did not observe performance improvement from adding them, and therefore use HumanML3D as our primary pre-training corpus.

A.2 Downstream Evaluation Datasets

We detail the information for each downstream real-world evaluation dataset as follows.

Opportunity [47] contains data collected from back, upper arms and lower arms, and features multiple sets of activities. We aim to predict the modes of locomotion such as standing and walking.

UCI-HAR [2] collects motion data from a smartphone located on the subject’s waist. The subject performs daily activities such as walking upstairs and walking downstairs.

MotionSense [37] collects data from a smartphone in the participant’s front pocket, featuring daily activities such as sitting and jogging.

w-HAR [5] contains motion time series data collected from the ankle. It captures daily physical activities such as jumping and lying down.

Shoaib [49] contains daily activities such as biking. Each participant is equipped with five smartphones on five positions: right jean’s and left jean’s pockets, belt, right upper arm and right wrist.

HAR70+ [58] tracks activities such as shuffling for older adult subjects. The motion time series are collected from the right front thigh and the lower back.

RealWorld [53] records daily activities such as climbing stairs from multiple body positions including chest, forearm, head, shin, thigh, upper arm, and waist.

TNDA-HAR [64] collects static as well as periodic daily activities such as cycling, from devices located at multiple body positions such as wrist, ankle and back.

PAMAP2 [46] monitors physical activities such as ironing, vacuum cleaning and rope jumping using devices located on the wrist, chest and ankle.

USC-HAD [69] records daily activities such as sleeping and taking the elevator with devices attached to the subject’s front right hip.

Mhealth [4] comprises body motion for common activities such as waist bending forward, frontal elevation of arms and knees bending. Devices are placed on the user’s chest, right wrist and left ankle.

Harth [33] records data in a free-living setting with devices located at the right thigh and lower back.

UT-Complex [50] contains different smartphone sensor data such as typing, drinking coffee and giving a talk, with devices positioned at wrist and pocket positions.

Wharf [6] records activities from wrist-worn devices, such as combing hair and getting up bed.

WISDM [59] collects diverse daily activities such as brushing teeth, eating soup, playing balls, and folding clothes, using data from the smartphone in the pocket and smartwatch on hand.

Table 3: Full-Shot performance on additional baselines before 2021. We bold the **best** results. The last column shows the average performance across 18 datasets with standard deviation.

Dataset	Metrics	Opportunity	UCL-HAR	MotionSense	w-HAR	Shoab	HAR70+	RealWorld	TNDA-HAR	PAMAP2	USC-HAD	Mhealth	Harth	UT-Complex	Wharf	WISDM	DSADS	UTD-MHAD	MMAct	Average
Number of Classes		4	6	6	7	7	7	8	8	12	12	12	12	13	14	18	19	27	35	
Level		Easy								Medium								Hard	Avg	
DeepCL [40]	Acc	84.4	86.9	86.2	93.4	62.4	97.7	69.9	89.6	82.7	45.2	73.8	95.5	88.0	46.8	69.0	69.6	54.9	50.2	74.8 _(16.7)
	F1	86.4	87.0	84.5	67.1	63.4	92.2	60.3	89.3	77.4	46.4	75.8	56.7	88.2	25.9	69.0	66.1	53.0	46.9	68.6 _(17.8)
MA-CNN [45]	Acc	82.9	89.8	92.0	67.2	93.2	86.1	73.9	96.0	94.7	37.3	78.7	97.9	93.8	20.9	51.5	84.5	48.8	28.9	73.2 _(24.2)
	F1	84.8	89.5	91.7	51.7	93.1	59.8	71.2	96.0	95.2	35.3	70.7	53.2	94.0	17.2	51.8	83.9	48.5	18.4	67.0 _(25.5)
XGBoost [9]	Acc	83.1	90.2	92.9	68.9	94.8	97.7	78.2	93.2	93.9	47.8	79.9	97.2	96.6	52.3	66.6	80.5	61.9	53.0	79.4 _(16.5)
	F1	85.1	90.1	91.6	56.1	94.7	77.5	77.6	93.2	93.9	47.7	74.1	64.4	96.7	30.2	66.0	79.4	60.3	51.4	73.9 _(18.6)
UniMTS	Acc	88.2	92.1	99.8	98.4	99.7	96.9	84.0	99.6	98.0	58.3	97.0	98.0	99.1	82.7	81.3	94.3	85.6	77.1	90.6 _(10.6)
	F1	89.1	92.2	99.8	98.8	99.7	87.9	81.2	99.6	98.1	63.1	97.3	86.7	99.2	51.7	80.9	94.2	83.2	71.7	87.5 _(13.4)

DSADS [1] comprises daily and sports activities such as exercising and rowing. Multiple devices are positioned at the torso, right arm, left arm, right leg, and left leg.

UTD-MHAD [8] contains diverse activities such as swiping arms, hand clapping, throwing, arm crossing, drawing and squatting. The devices are worn on the subject’s right wrist or the right thigh depending on whether the action is mostly an arm or a leg type of action.

MMAct [26] presents a large-scale activity dataset covering a wide range of daily life activities such as carrying, talking on phone and falling. Devices recording motion time series include a smartwatch as well as a smartphone inside the pocket of the subject’s pants.

A.3 Baselines

We detail the baseline settings as follows.

ImageBind [16]: We employ the pre-trained weights from “imagebind_huge” for zero-shot evaluation. During fine-tuning, we add a linear layer to map ImageBind embeddings to the number of activity classes. We fine-tune both ImageBind and the linear layer during fine-tuning, which performs better than simply tuning the linear layer.

IMU2CLIP [38]: The pre-trained weights of IMUCLIP are not released. Therefore, we first follow their pre-training implementation³ to pre-train on Ego4D datasets [17]. During fine-tuning, we add a linear layer after IMU2CLIP embeddings and fine-tune both IMU2CLIP and the linear layer.

IMUGPT [28]: We choose DeepConvLSTM as the backbone model, which shows the best performance as reported in their original paper. We remove the supervised distribution calibration phase, which relies on labeled downstream data and conflicts with the zero-shot setting objectives.

HARGPT [24]: The method directly prompts large language models to classify motion time series. We down-sample motion time series to 10 Hz as used in their paper and follow their prompt template.

LLaVA [30]: We visualize motion time series as 2D plots and use these visualizations as input for the pre-trained model of “llava-v1.5-7b”.

TST [68]: This is a Transformer-based representation learning framework with several downstream tasks including multivariate time-series classification. We follow the framework to first pre-train the Transformer model in an unsupervised fashion and then fine-tune the pre-trained model on the downstream classification task.

TARNet [11]: The model proposes task-aware representation learning that reconstructs important timestamps guided by self-attention score distribution from end-task training. We jointly train the reconstruction task and the classification task.

³<https://github.com/facebookresearch/imu2clip>

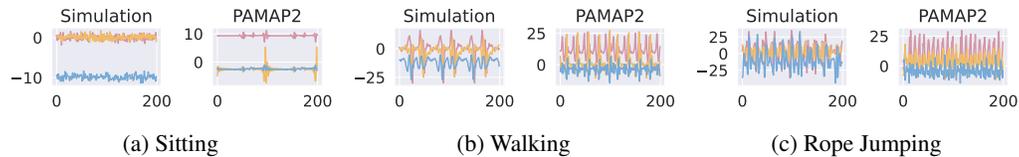


Figure 9: Simulated motion time series show similar patterns as real PAMAP2 time series.

TS2Vec [67]: The method performs contrastive learning to learn contextual representations of time series. We follow their implementation to first apply contrastive learning and then train a linear regression model for each dataset.

BioBankSSL [66]: The paper proposes a pre-trained model for human activity recognition using a large-scale UK Biobank wrist accelerometer dataset with multi-task self-supervised learning.

DeepConvLSTM [40] (shown as “DeepCL” in Table 3 due to space limit): The model applies convolutional layers to automatically learn feature representations and further applies LSTM to capture the temporal dependencies between their activations.

MA-CNN [45]: The model first extracts preliminary features for each motion time series modality through its own dedicated convolutional layers, then the extracted intra-modality features are combined through fully-connected layers for motion time series classification.

XGBoost [9]: This is a scalable end-to-end machine learning system for tree boosting, which has been widely recognized in machine learning and time series analysis.

THAT [29]: The model proposes a two-stream convolution augmented human activity transformer which captures both time-over-channel and channel-over-time features in a two-stream structure.

TimesNet [60]: This is a task-general backbone for time series analysis including classification by modeling the multi-periodicity and extracting temporal variations.

GPT4TS [72]: This is also a task-general framework that includes time series classification. The model is based on a frozen pre-trained language model, so we also adopt it as a few-shot fine-tuning baseline. However, the method is not suitable for zero-shot evaluation, as it requires training a separate classifier head for each downstream dataset and does not generalize across activities.

SHARE [71]: This is a sequence-to-sequence model that contains an encoder to extract motion time series features, as well as a decoder to generate label name sequences to capture label semantics.

A.4 Simulated Data

In addition to the simulated data for wrist as shown in Figure 6, we present more examples for other device locations such as ankle in Figure 9. We observe consistently similar patterns between simulated motion time series and real PAMAP2 data across activities of various intensity levels, ranging from sitting to walking and rope jumping.

A.5 Efficiency Analysis

For *space complexity*, the graph encoder of UniMTS contains only 4.94M parameters, which is significantly smaller compared with the 18.69M used in the IMU encoder of the best existing baseline ImageBind. For *time complexity*, fine-tuning of UniMTS is also efficient. On one example dataset of UCI-HAR, full-shot fine-tuning of UniMTS takes approximately 1.3 minutes to converge while it takes approximately 9.8 minutes for ImageBind to converge. Moreover, we have run a power estimate assuming 0.1Hz cadence (i.e., 10-second window size), and it takes approximately 22.64 mW to run the whole graph model on an eNPU (embedded Neural Processing Unit), which is much smaller than ImageBind IMU encoder’s power consumption of approximately 702 mW. Therefore, UniMTS is efficient for real-world applications and suitable to be deployed on edge devices.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The main claims in the abstract and introduction are that we build the first unified pre-trained model for motion time series that is able to generalize to various device locations, orientations and activities, which are verified by our experimental results.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss four limitations and corresponding future works as detailed in "Limitation and Future Work" of Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper does not involve new theoretical results. We have properly cited the literature for prior results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed experimental settings in Section 4.1, as well as Section A.1, Section A.2 and Section A.3 in Appendix. We also provide the code in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide them as a zip file in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify all the training and test details in Section 4.1, as well as Section A.1, Section A.2 and Section A.3 in Appendix. We also provide the code in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report error bars and apply statistical significance tests for all experiments, including both zero-shot, few-shot and full-shot settings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We specify computer resource details in Section 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics in every aspect.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the broader impacts and limitations in Section 5.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not pose high risk for misusing data or models. We pre-train the model exclusively on synthetic data to ensure strict privacy control at the earliest stages of model development.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite all the code and data used in this paper, and respect their license and terms of use.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We provide the new assets as a zip file in the supplementary material.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.