IMUGPT 2.0: Language-Based Cross Modality Transfer for Sensor-Based Human Activity Recognition

ZIKANG LENG, College of Computing, Georgia Institute of Technology, USA
AMITRAJIT BHATTACHARJEE, School of Interactive Computing, Georgia Institute of Technology, USA
HRUDHAI RAJASEKHAR, School of Interactive Computing, Georgia Institute of Technology, USA
LIZHE ZHANG, College of Engineering, Georgia Institute of Technology, USA
ELIZABETH BRUDA, College of Computing, Georgia Institute of Technology, USA
HYEOKHYEN KWON, Department of Biomedical Informatics, Emory University, USA
THOMAS PLÖTZ, School of Interactive Computing, Georgia Institute of Technology, USA

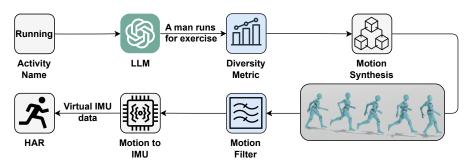


Fig. 1. Overview of the proposed language-based cross modality transfer system for sensor based human activity recognition. An LLM automatically generates textual descriptions of activities, which are then converted into motion sequences by a motion synthesis model. A novel motion filter then screens out incorrect sequences, retaining only relevant motion sequences for virtual IMU data extraction. A new diversity metric measures shifts in the distribution of generated textual descriptions and motion sequences, allowing for the definition of a stopping criterion that controls when data generation should be stopped for most effective and efficient processing and best downstream activity recognition performance.

One of the primary challenges in the field of human activity recognition (HAR) is the lack of large labeled datasets. This hinders the development of robust and generalizable models. Recently, cross modality transfer approaches have been explored that can alleviate the problem of data scarcity. These approaches convert existing datasets from a source modality, such as video, to a target modality (IMU). With the emergence of generative AI models such as large language models (LLMs) and text-driven motion synthesis models, language has become a promising source data modality as well as shown in proof of

Authors' addresses: Zikang Leng, zleng7@gatech.edu, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, USA; Amitrajit Bhattacharjee, amit.bh@gatech.edu, School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia, USA; Hrudhai Rajasekhar, hrajasekhar3@gatech.edu, School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia, USA; Lizhe Zhang, Izhang762@gatech.edu, College of Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA; Elizabeth Bruda, ebruda3@gatech.edu, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, USA; Hyeokhyen Kwon, hyeokhyen.kwon@emory.edu, Department of Biomedical Informatics, Emory University, Atlanta, Georgia, USA; Thomas Plötz, thomas.ploetz@gatech.edu, School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia, USA.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2023 Copyright held by the owner/author(s).

, Vol. 1, No. 1, Article . Publication date: February 2023.

concepts such as IMUGPT. In this work, we conduct a large-scale evaluation of language-based cross modality transfer to determine their effectiveness for HAR. Based on this study, we introduce two new extensions for IMUGPT that enhance its use for practical HAR application scenarios: a motion filter capable of filtering out irrelevant motion sequences to ensure the relevance of the generated virtual IMU data, and a set of metrics that measure the diversity of the generated data facilitating the determination of when to stop generating virtual IMU data for both effective and efficient processing. We demonstrate that our diversity metrics can reduce the effort needed for the generation of virtual IMU data by at least 50%, which open up IMUGPT for practical use cases beyond a mere proof of concept.

 ${\tt CCS\ Concepts: \bullet Human-centered\ computing} \to {\tt Ubiquitous\ and\ mobile\ computing; \bullet Computing\ methodologies} \to {\tt Artificial\ intelligence}.$

Additional Key Words and Phrases: Wearables; Activity recognition; Virtual IMU Data; Motion Synthesis; LLM

ACM Reference Format:

1 INTRODUCTION

The use of wearable sensing devices for Human Activity Recognition (HAR) is one of the central pillars of mobile and ubiquitous computing. HAR finds extensive application in various domains, including fitness tracking [15, 33], health monitoring [7, 46], sign language recognition [49, 54], and in identifying critical events like vehicular accidents and falls [39, 48]. The recognition is typically achieved through the use of supervised learning methods to classify segmented data streams into activities of interest (or the null class). The effectiveness of these methods relies heavily on the availability of accurately annotated datasets. However, one of the biggest challenges that the HAR community faces is the lack of large-scale labeled datasets due to the expensive annotation process, the need for domain experts, and potential issues with invasion of privacy [16, 18, 31, 35, 61].

To address the problem of annotation scarcity, researchers have explored the use of cross modality transfer approaches. These approaches aim to convert data from other modalities, such as video [37], motion capture [82], and optic data [83], into *virtual IMU data*. They leverage existing large-scale annotated datasets in the source data modality, and, in turn, use them to generate large-scale virtual IMU datasets with accompanying annotations. The generated data then serve as (additional) labeled training data for the downstream classifier to boost the model's performance.

With the recent emergence of generative foundation models, natural language has become a new promising candidate data modality for cross modality transfer. Leng et al. [43] piloted the concept of IMUGPT that combines large language models (LLMs), motion synthesis methods, and signal processing techniques to generate virtual IMU data with no manual effort. The key idea is to use LLMs to generate diverse textual descriptions of the different ways that humans can perform certain activities. The generated textual descriptions are then converted to 3D human movement sequences using motion synthesis methods [86]. The resulting sequences of action-specific poses are then converted into virtual IMU training data. In principle, such a system allows for the generation of labeled datasets that are larger and encompass more activities than any existing ones. Such generated datasets would help pave the way for the development of more complex HAR models that are robust, generalizable, and allow for the analysis of more complex human movements and gestures, yet without involving any human participants. While the initial IMUGPT system served as a preliminary proof of concept, it demonstrated promise in small-scale experiments, where the generated virtual IMU data led to significant improvements in performance in the downstream classifier. This paper builds on those initial results by significantly expanding on the proof of concept through a range of technical modifications and additions, that render the approach valuable for practical applications and by thoroughly evaluating it in a large scale experimental evaluation study.

Our goal is to determine not only how but also how much and what kind of virtual IMU data shall be derived from the language-based input, for effective and efficient cross modality transfer. We hypothesize that more diverse and relevant virtual IMU data leads to greater performance improvements for downstream applications. To ensure the relevance of the generated virtual IMU data, we introduce a motion filtering module capable of filtering out generated motion sequences that are irrelevant (due to limitations in motion synthesis models), preventing such virtual IMU data from being used for model training. Additionally, we also introduce two metrics to measure the diversity of virtual IMU data, through diversity of both textual descriptions and motion sequences. Subsequently, we correlate these metrics to the downstream classification performance, and also use them to determine the optimal stopping points for generating virtual IMU data. Such stopping points indicate that the generated data have reached a saturation point in terms of diversity, beyond which, generating additional data does not enhance the diversity of the dataset nor benefit the performance of the downstream classifier. Through these extensions, IMUGPT can automatically determine the optimal quantity of virtual IMU data to generate. This provides researchers with efficient means to produce and utilize virtual IMU data, leading to more robust and accurate HAR applications while saving time and computing resources.

We further conduct a comprehensive evaluation of the IMUGPT system—in its extended form as presented in this paper—that explores the effectiveness of such a language-based cross modality transfer approach across a broader spectrum of activities over a range of practically relevant HAR application scenarios. This evaluation involves comparing different LLMs and motion synthesis models, with specific focus on their impact on the performance of downstream activity recognition systems. Our results show that using GPT-3.5 [2] and T2M-GPT [86] for virtual IMU data generation lead to the best downstream performance. Additionally, we demonstrate that our proposed diversity metrics can reduce the time and compute resources needed for the data generation process by over 50%, making IMUGPT more practically usable.

Through our methodological extensions and the extensive experimental evaluation of IMUGPT, we demonstrate the effectiveness of language-based cross modality transfer methods for deriving robust and effective HAR systems in practical application scenarios, thereby alleviating, if not circumventing, the most notorious challenge for contemporary HAR systems: the lack of large-scale, labeled training data.

2 RELATED WORK

Sensor-based human activity recognition using wearable devices (HAR), i.e., the automated assessment of what a person is doing (and when), has come a long way and it is considered one of the central pillars of mobile, wearable, and ubiquitous computing [36]. Advances in the miniaturization of sensing hardware have enabled the widespread adoption and integration of inertial measurement units (IMUs) to capture movement information of a user as one of the most central contextual factors, that if analyzed properly, allows for situated and often personalized adaptation of computing services [82]. As such, a multitude of applications have been realized, which are based on the core activity recognition backend, including health monitoring [7], exercise and sports tracking [33], and gesture recognition for intuitive user interfaces [49], to name but a few.

Conventional HAR applications are typically based on (some variant of) the "Activity Recognition Chain (ARC)" (e.g., [14]), which divides the analysis problem into five general processing blocks: recording, preprocessing, (sliding-window) segmentation, feature extraction, and classification. While the first two components are typically addressed through classical signal processing approaches, the remainder of the processing pipeline typically involves some form of machine learning methods, often following the supervised learning paradigm for which substantial amounts of *labeled* training data are required.

More contemporary HAR workflows are based on end-to-end deep learning methods that promise more effective, generalizable models, for example, by overcoming the manual design of feature representations as it was

required in the conventional ARC [59, 60]. Yet, even recent state-of-the-art end-to-end learning methods require substantial amounts of labeled sample data for model training – an obstacle that cannot easily be overcome.

In what follows, we first give an overview of the most pressing challenges to sensor- and machine learning based human activity recognition with wearables. We then focus on related work in the field of general cross modality transfer as it is used to overcome shortages of labeled training data, which is the scope of the work presented in this paper. We then give an overview specifically on language-based cross modality transfer, namely the original IMUGPT approach [43], which serves as the basis for this paper.

2.1 Challenges with HAR using Wearables

Despite considerable advancements in HAR and its numerous practical applications, there exist significant challenges that are inherent and specific to HAR using wearable sensing platforms [16, 20, 61, 63]. The most notable ones are: (i) paucity of labeled data – the time-consuming and expensive nature of wearable data collection along with its inherent privacy concerns have led to datasets being relatively smaller in size; (ii) difficulty in data annotation – ambiguity in the target activities and its context results in incorrectly labeled data; (iii) conflicting variance in data – induced by similar activities being performed differently and different activities resulting in similar sensor readings; and (iv) sensor noise – auto-calibration of sensors that depend on temperature and gravity corrections [78], and the underlying architecture of MEMS sensors [56] inducing noise in the data.

Arguably, the most pressing challenge is the lack of labeled training data. While recording data on wearable devices is uncomplicated due to constant activity tracking by sensors, the impracticality of either continuously recording a user's activities or asking users to repeatedly provide ground truth labels renders annotating the sensor data challenging, to say the least. The research community has developed a range of approaches to specifically tackle the sparse data problem, including self-supervised learning [26–28, 67, 73], few-shot learning [21], semi-supervised learning [10], prototypical learning [9, 17], adversarial learning [8, 41], and transfer learning [69]. Recently, the idea of cross modality transfer has gained traction in the community, in which, knowledge is transferred from one data modality to another, e.g., from image to text.

2.2 Cross Modality Transfer

cross modality transfer methods have recently been introduced in a number of application domains with the goal of opportunistically utilizing existing datasets from source modalities other than those targeted by a specific application. The key motivation here is to combat the lack of domain and modality specific labeled data ("small labeled dataset problem") by automatically converting sensor readings from one modality to another, resulting in "virtual sensor data" in the target modality, thereby transferring knowledge in form of the underlying fundamentals of the human movements from one modality to another.

Approaches for generating virtual data through cross modality transfer from different data sources have been explored across various domains (Table 1). cross modality transfer has been actively investigated for the generation of virtual IMU data for HAR from various sources. Xiao et al. [82] use a Convolutional Neural Network (CNN) fine-tuned with real IMU data to generate "skinned multi-person linear" (SMPL) model [52] parameters from motion capture data, which is further used to generate virtual acceleration and orientation data. An extension to this was presented by Uhlenberg et al. [76] who created 3D human surface models and skeletal models from motion capture data, which were then used to simulate daily activities and synthesize inertial data.

Xia et al. [81] introduced a virtual spring-joint based sensor module to augment simulated virtual acceleration data extracted from 2D exercise videos. Recently, IMUTube [36–38] was introduced to convert existing large-scale videos into virtual IMU data through a computer vision pipeline involving 2D pose extraction and conversion to 3D poses on which individual joints are tracked in order to generate tri-axial acceleration and gyroscope data.

Table 1. Overview of prominent cross modality transfer approaches for virtual data generation.

System	Source Data	Target Data	Task	Method	
Liu et. al. [50]	MRI	CT	Medical Diagnosis	Neural Network	
Xia et. al. [80]	Distance	Image	Hand Gesture Recognition	Neural Network	
Uhlenberg et. al. [76]	Motion Capture	Gyro & Acc		Surface Modeling	
Xiao et. al. [82]	Monon Capture	Gylo & Acc		Neural Network	
Zhang et. al. [90]		Acc	Human Activity Recognition	INCUIAI INCUMOIR	
Xia et. al. [81]		7 ICC	Truman Activity Recognition	Pose Estimation & Biomechanics	
IMUTube [36-38]		Gyro & Acc		1 osc Estimation & Dionicenanies	
Rey et. al. [66]		Gy10 tt 71cc			
ZeroNet [51]	Video	Gyro & Acc	Gyro & Acc		Pose Estimation & Neural Network
Vi2IMU [68]	Video	Gy10 tt 71cc	Sign Language Recognition		
SignRing [44]		Acc		Pose Estimation & Biomechanics	
Lu et. al. [53]		Gyro	Head Motion Recognition	Pose Estimation & Neural Network	
Vid2Doppler [4]		Doppler			
IMU2Doppler [12]	Gyro & Acc	Боррісі		Neural Network	
IMG2IMU [84]	Image	Gyro & Acc	Human Activity Recognition	rectrai rectwork	
AudioIMU [45]	Audio	Gy10 & Acc	Truman Activity Recognition		
Visual Accelerometer [83]	Optic	Acc		Pose Estimation & Biomechanics	
Leng et. al. [43]	Text	Gyro & Acc		Motion Synthesis & Biomechanics	

Similarly, Rey et al. [66] use a combination of real sensor data and 2D poses extracted from laboratory videos to train a regression model. The model takes arbitrary videos as inputs and generates simulated sensor data.

In the context of Sign Language Recognition, ZeroNet [51] uses 3D finger pose extraction on publicly available sign language videos and compares it to IMU data retrieved from a finger ring from an unknown user gesture using a combination of Dynamic Time Warping and Convolutional Neural Networks. SignRing [44] approaches perform Sign Language Recognition using a triangulation-based algorithm to convert 2D videos of signers signing to 3D hand pose and then compute the sensor's 3D acceleration by tracking the movements of the index fingers. Vi2IMU [68] uses a combination of wrist and 3D displacement estimation with LSTM-based architectures based on 2D wrist, arm, and hand joint positions extracted from publicly available videos to generate virtual IMU data.

Lu et al. [53] use Face Tracking [11] and 6DRepNet [29] to generate virtual IMU data for head motions from videos. VisualAcc [83] uses photometric effect-based interrogation and an optic-to-inertia transformer that senses human motion passively and reconstructs inertial data using the Optical Motion Field (OMF).

2.3 Text-Driven Motion Generation

A common challenge in the aforementioned cross modality transfer approaches is the gathering of relevant videos and motion capture datasets. Considering more niche applications like Sign Language Recognition, finding the right video dataset can be a time-consuming process. Furthermore, the quality of videos can be an issue in pose extraction and estimation. In order to address this, Leng et al. [43] proposed IMUPGT, a language-based cross-modal transfer approach that generates diverse virtual IMU data from virtual textual descriptions of activities using a combination of LLMs, motion synthesis models, and signal processing techniques, eliminating the need to search for videos. This system serves as the foundation for the work presented in this paper and more details about it are given in section 2.4.

As a basis for approaches like IMUGPT, textual descriptions need to be converted into human motions that are performing underlying activities of interest. Much research has gone into automatically generating 3D human motion using text descriptions of activities, which serves as a basis for the use of language as the source modality in cross modality transfer to generate virtual IMU data. With the recent introduction of the HumanML3D dataset [23] – a large 3D human motion dataset with text descriptions – motion synthesis and diffusion-based models

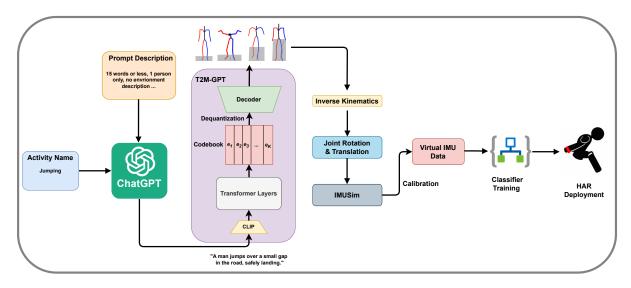


Fig. 2. Overview of Leng et al.'s IMUGPT [43]. ChatGPT is used to generate diverse textual descriptions of the specified activities. Subsequently, a motion synthesis model, T2M-GPT [86], generates human motion sequences using the textual descriptions. Virtual IMU data can then be extracted from the generated motion sequences and used for training HAR models. (Figure adopted from Leng et al. [43] and used with permission)

like MotionGPT [30], T2M-GPT [86], MotionDiffuse [87], and ReMoDiffuse [88] have been introduced that are capable of producing more realistic human motion sequences using textual descriptions as the input.

T2M-GPT and MotionGPT use a motion tokenizer based on a Vector Quantized Variational Autoencoder (VQ-VAE) architecture, and a motion-aware language model based on a Generative Pre-trained Transformer (GPT) architecture. MotionDiffuse adopts a denoising diffusion probabilistic model (DDPM) using a cross modality linear transformer to convert a given text into a motion sequence. On the other hand, ReMoDiffuse enhances a denoising diffusion model using hybrid retrieval and uses a Semantic-Modulated Transformer for sequence generation. In our work, we explore and evaluate these motion synthesis models in the IMUGPT setup.

The emergence of *Large Language Models* (LLMs) such as GPT-3 [13], GPT-4 [57], PaLM 2 [5], and LLama 2 [75] has revolutionized the field of natural language processing (NLP) due to their impressive capabilities in various NLP tasks, made possible by their massive training corpora. The success of LLMs has also attracted attention in other fields, where they are integrated with an array of AI models, with the LLMs facilitating the interaction between the AI models and the end-user [70, 79]. Additionally, in a recent study, Athanasiou et al. [6] discovered that motion and activity information is encoded within LLMs, demonstrating that GPT-3 could be used to identify the body parts involved in different activities. Inspired by this result, we apply LLMs to the problem of activity filtering in this work to filter out irrelevant activity descriptions.

2.4 IMUGPT

Pioneered by Leng et al. [43], language-based cross modality transfer for sensor-based human activity recognition—IMUGPT (Figure 2)—takes textual descriptions of relevant activities as input, which are then converted to sequences of 3D human motions, e.g., through the T2M-GPT model [86]. The IMUTube backend [37] is then applied to the resulting motion sequences and virtual IMU data are generated automatically that can then be used for training HAR models. IMUGPT consists of three major components as follows:

- (1) LLM. What: An LLM is used to generate diverse textual descriptions of a person performing a specified activity. Why: Human activities are diverse in the real world, as the same activity can be performed in different ways by different people (e.g., a skinny teenager vs. a muscular athlete) and under different contexts (e.g., happily vs. sadly). To develop robust and generalizable HAR models capable of recognizing all these variations, it is essential to reflect this diversity in the training data. Generating diverse textual descriptions for activity performance is the first step toward ensuring that the generated virtual IMU data accurately represents real-world diversity. Additionally, using an LLM for text generation automates and simplifies the process, eliminating the need to search for video data as required in video-based cross modality transfer approaches.
- (2) **Motion Synthesis**. What: A motion synthesis model that takes in textual descriptions of the activity and converts them into three-dimensional human motion sequences. Why: The motion synthesis model bridges the gap between text and human motion. To extract virtual IMU data, understanding how a person moves in three-dimensional space is essential. This is similar to how three-dimensional poses are estimated in video-based methods.
- (3) **Motion to IMU**. What: A method to convert generated motion sequences into virtual IMU data, either biomechanically [85] or through neural networks [44, 68]. Why: The virtual IMU data extracted from the motion sequences can be used to train a deployable HAR model, either alone or in conjunction with real IMU data.

IMUGPT was evaluated on locomotion activities [65, 72, 89]. Preliminary experiments demonstrated the promising potential of the approach, showing that the extracted virtual IMU data can lead to significant improvements in the downstream HAR performance.

3 IMUGPT 2.0: TOWARDS PRACTICAL APPLICATIONS OF LANGUAGE-BASED CROSS MODALITY TRANSFER FOR HAR

While the initial prototype of IMUGPT [43], as summarized above, demonstrated the general effectiveness of the idea of language-based cross modality transfer in sensor-based human activity recognition, that proof of concept fell short on two major aspects: *i*) It is unclear when data generation should be stopped, which has implications on the computational efficiency (and costs) of the virtual IMU data generation process; and *ii*) The relevance of the generated virtual IMU data remains uncertain until a downstream HAR system has been trained and evaluated. It may very well be the case that some generated IMU data are not useful or are possibly even detrimental to the targeted HAR application scenario.

In this work, we present extensions to the initial IMUGPT prototype through which we add two additional modules thereby explicitly aiming to address aforementioned limitations. Figure 1 illustrates these extensions in blue. These new modules facilitate language-based cross modality transfer, making it more practical as follows:

- (1) **Diversity Metrics.** What: A method to measure the diversity of textual descriptions and motion sequences generated by LLMs and motion synthesis models, respectively. See section 3.1 for details. Why: We hypothesize that the downstream model performance depends on the diversity of the virtual IMU data, as more diverse data can enhance model performance. With the proposed diversity metrics, we introduce a saturation point identification algorithm (algorithm 1) that identifies the point at which generating additional textual descriptions no longer adds meaningful information to the existing pool of generated texts. This indicates a stopping point for text generation. Automatically identifying when to stop data generation saves time and computing resources, which are not unlimited.
- (2) **Motion Filter**. What: A pipeline that identifies and filters out motion sequences that do not accurately portray the specified activity. See section 3.2 for details. Why: The motion synthesis model does not

always generate motion sequences that accurately portray a person performing the specified activity, as it can confuse closely related activities (e.g., climbing up stairs vs. climbing downstairs), resulting in irrelevant motion sequences. We hypothesize that irrelevant motion sequences will negatively impact the downstream classifier's performance by introducing noise.

3.1 Diversity Metrics

A key limitation of IMUGPT [43] is that there is no clear indication as to when the generation of data should be stopped. While this was acceptable for a proof of concept, in order to make the system viable and practical, it is important to have an explicit stopping point for data generation. This would ensure control over the time and computation costs associated with the generation process, as well as making sure that the generated virtual data actually are of benefit for the HAR modeling task. For this reason, we propose utilizing the diversity of the generated data to determine the optimal stopping point. Note that the diversity of the virtual IMU data can be measured either through the textual descriptions or the motion sequences it is generated from. Since the motion sequences are obtained from the textual descriptions themselves, we hypothesize that the diversity of both should be correlated. We validate this in section 4.4. Intuitively, calculating the diversity of the textual descriptions is more practical as it precedes the motion sequence generation step, saving computational costs.

Diversity enables us to quantify the amount of useful information present in the data, and then utilize that to make a judgment about downstream performance without running the entire pipeline. If the diversity is high, then downstream models would be exposed to a broader range of data points during training. This results in the model learning to make predictions on a wider variety of cases, thereby improving the performance. Diverse training data also helps prevent the models from overfitting, since a limited dataset would lead to overexposure to a small section of the feature space. It is helpful to have some practical measures that facilitate the estimation of downstream performance at the time of generation itself, assuming that diversity is computationally inexpensive to calculate.

In order to calculate the diversity of data (whether textual descriptions or motion sequences), we generate embeddings for the data. Embeddings are vectorial representations of data that serve to capture the semantic and syntactic information present in the data. Each data point is mapped to a vector in an 'embedding space'. Since vector operations can be utilized to operate on the data in an embedding format, they are a useful representation for calculating diversity metrics. We generate embeddings as follows:

- Textual descriptions: The text prompts are passed through SentenceTransformers's 'all-mpnet-base-v2 model' [1, 64] to generate embeddings for each prompt. This model was trained on one billion sentence pairs to capture the semantic information of its input text, and thus, the generated embeddings serve as a suitable representation of the sentence.
- Motion Sequences: Each motion sequence is passed through a model trained on the HumanML3D dataset [23] to generate the embeddings for the sequence. This model is drawn from the motion feature extractor trained in Guo et al. [23], and is commonly used in the community [86].

Overall, we compute two types of diversity: *i)* absolute diversity, and *ii)* comparative diversity. Note that these metrics are applicable to any sets of embeddings, irrespective of whether they are generated from textual descriptions or motion sequences. The diversity calculation methods themselves are universally applicable to any collection of embeddings. Absolute diversity provides a quantitative measure for the amount of diverse information contained in a set, while comparative diversity showcases how much two sets differ in terms of their diversity. These measures are detailed further in the subsequent sections.

3.1.1 Absolute Diversity Metrics. We use two methods to compute the absolute diversity of a set of embeddings – the standard deviation method [40] and the centroid method [19]. For both metrics, a higher value corresponds to a more diverse set, and vice versa.

Standard Deviation Metric. The standard deviation method is based on the diversity metric introduced in Lai et al. [40]. Interpreting embeddings as vectors in a high-dimensional embedding space, the goal is to characterize the dispersion (the "spread") of a cluster of such vectors. If the cluster is assumed to be distributed as a multi-variate Gaussian, each isocontour will be shaped as an axis-aligned ellipsoid. The radii of the ellipsoid along each axis can be computed by calculating the standard deviation of the vectors in the cluster along each of the axes. Thus, computing the geometric mean of the radii will capture the generalized radius of the cluster, providing a metric for the diversity. Assuming the set of embeddings S consists of n embedding vectors, each of dimension k, the set can be formalized as $S = \{x_i\}_{i=1}^n \in \mathbb{R}^k$. Thus, the standard deviation along an axis $j \in [1, k]$ is computed as follows:

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^n \left(x_i^j - \mu^j\right)^2}{n}}, \text{ wherein } \mu^j = \frac{\sum_{i=1}^n x_i^j}{n}$$
 (1)

The final diversity score M_{std} is then calculated as:

$$M_{std} = \left(\prod_{j=1}^{k} \sigma_j\right)^{\frac{1}{k}} \tag{2}$$

Centroid Metric [19]. Assume the set of embeddings $S = \{x_i\}_{i=1}^n \subset \mathbb{R}^k$, consisting of n embedding vectors, each of dimension k. First, the centroid vector x_{cent} of all embedding vectors is calculated:

$$x_{cent} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{3}$$

The diversity score M_{cent} is computed as the mean of the sum-squared distances of all embedding vectors from the centroid vector:

$$M_{cent} = \frac{\sum_{i=1}^{n} (d(x_i, x_{cent}))^2}{n}, \text{ wherein } d \text{ is the Euclidean distance } d(x_i, x_{cent}) = \sqrt{\sum_{i=1}^{k} \left(x_i^j - x_{cent}^j\right)^2}$$
(4)

The intuition for this method stems from the fact that the higher the diversity for a set of embeddings, the farther apart their vectors will be spread in the embedding space. As a result, the points will tend to be at a farther distance from the overall mean than for a less diverse set, and the centroid metric quantifies this aspect. Since our goal is to quantify the dispersion of all the vectors in the embedding space, we only utilize a single centroid for the entire set of embeddings.

3.1.2 Comparative Diversity. The textual description generation process starts by generating an initial set of descriptions. Then, newly generated batches of textual descriptions are appended sequentially to the pre-existing set of descriptions. If adding a new batch of textual descriptions improves the diversity of the pre-existing set, then continuing to further generate newer batches is well-motivated as it would contribute to the diversity of the virtual IMU data, leading to better downstream performance. On the contrary, if diversity does not improve, newer batches should not be generated. The goal of comparative diversity is to quantify this change in diversity that occurs upon adding a new batch to the pre-existing set.

The comparative diversity between two sets of embeddings is computed using the Maximum Mean Discrepancy (MMD) test [22]. MMD is a kernel-based statistical test which indicates whether two given sets of samples are

Algorithm 1: Saturation Point Identification

```
Function to calculate MMD: MMD calculator
                                                                                           // n: size of the set
Pre-existing set of text embeddings: S \ emb[n]
perc \leftarrow 0.05
                                                     // Percentage of text prompts generated at each step
early\_stop \leftarrow 5
                                                        // Hyperparameter to determine the stopping point
stop\_count = 0
range\_min, range\_max \leftarrow -1, -1
while stop condition \leq early stop do
    B[perc * n] \leftarrow Generate new batch of prompts
    B_{emb}[perc * n] \leftarrow Compute embeddings for B
    score, standard\_dev \leftarrow MMD\_calculator(S\_emb, S\_emb + B\_emb)
    if score > range_min and score < range_max then</pre>
        stop\_condition + = 1
        range\_min = min(score - standard\_dev, range\_min)
        range max = max(score + standard dev, range max)
    else
        range\_min = score - standard\_dev
        range\_max = score + standard\_dev
        stop\_condition = 0
    S\_emb \leftarrow S\_emb \cup B\_emb
                                                           // Append current batch to the pre-existing set
end
return S_{-emb}
```

drawn from the same distribution or not. The higher the value, the farther the distributions of the two sets of samples. Given a space \mathbb{R}^d and independent and identically distributed samples $X_i \in \mathbb{R}^d$, $i = 1, ..., N_X$ sampled from $X \sim P_X$ and $Y_i \in \mathbb{R}^d$, $i = 1, ..., N_y$ sampled from $Y \sim P_Y$, the MMD quantifies the difference between P_X and P_Y . It is calculated as follows:

$$MMD = \sum_{i=1}^{N_x} \sum_{j=1}^{N_x} K(X_i, X_j) + \sum_{i=1}^{N_y} \sum_{j=1}^{N_y} K(Y_i, Y_j) - 2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} K(X_i, Y_j).$$
 (5)

Here, *K* is the Gaussian kernel. Thus, we interpret any two sets of embedding as being drawn from two distributions, and calculate the "distance" between these two distributions to serve as the difference in diversity. If the distributions of the two sets of embeddings are similar, they will have a similar diversity, and correspondingly the MMD value for the pair of sets will be lower. Further details about how the comparative diversity is utilized to halt the generation process is highlighted in the subsequent sections.

3.1.3 Saturation Point Identification. Given the comparative diversity measure introduced in the previous section, our goal is now to utilize this metric in a manner wherein the generation of textual descriptions can be halted once the generated set of descriptions saturates (i.e. generating further textual descriptions would not improve the diversity of the pre-existing set). In order to determine this halting point, we propose the Saturation Point Identification algorithm (algorithm 1). Given a pre-existing set of textual descriptions S of size n, the algorithm iteratively generates new batches of descriptions and tests if adding the new batch to the pre-existing set alters the comparative diversity significantly. The size of the new batch is a percentage of the size of the pre-existing set, and is a tunable hyperparameter. Setting a higher value for the percentage would imply a coarser steps towards the saturation point.

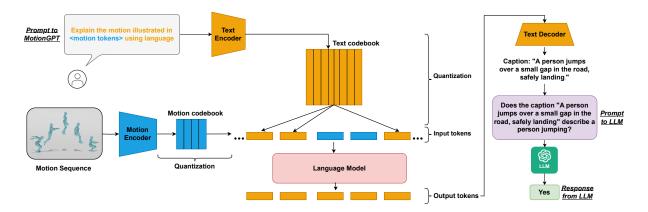


Fig. 3. Overview of the proposed motion filter. Using MotionGPT [30], we obtain motion captions which are textual descriptions of the input motion sequence. To obtain the motion caption, a language model takes in encoded text and motion tokens and then generate output tokens. The output tokens are decoded to recover the motion caption. Then, we pass the motion caption into a LLM to determine if the motion sequence correctly portrays a specified activity.

If adding the new batch alters the diversity substantially, then the diversity hasn't saturated yet, and the iteration continues. On the other hand if the diversity does not change meaningfully, it may be close to reaching saturation, and the algorithm terminates. Actual termination is controlled via a hyperparameter that controls for tolerance for saturation. Note that for calculating the MMD between two sets of embeddings, the sets must have the same size. However, in this case, we calculate the MMD between two sets of differing sizes (in algorithm 1, these are S_{emb} and $S_{emb} \cup B_{emb}$ respectively). In order to address these, we randomly resample the smaller set (since random resampling does not alter the underlying population distribution) until it has the same size as the larger set and then calculate the MMD. Note that this process of resampling is conducted multiple times, resulting in multiple values of MMD, and thus the function returns the mean score along with the standard deviation for MMD calculation ($MMD_{calculator}$ function in algorithm 1). Our algorithm is evaluated in section 4.4.2.

3.2 Filtering out incorrectly generated motion sequences

Another limitation of IMUGPT is that the motion synthesis model may generate motion sequences that do not accurately describe the intended activity, which leads to irrelevant virtual IMU data being extracted, potentially degrading the downstream performance. To address this, we propose a motion filter that can filter out incorrectly generated motion sequences. Figure 3 illustrates the operation of the motion filter. To determine if a given motion sequence accurately portrays the activity of interest, the sequence generated by the motion synthesis model is first processed by a motion captioning model [30]. This model outputs a textual description of the motion sequence. The resulting motion caption is then evaluated by an LLM, which provides a binary 'yes' or 'no' answer indicating whether the caption accurately describes the specified activity. This output from the LLM enables us to filter out any incorrectly generated motion sequences.

3.2.1 Motion Captioning. The motion captioning task, also called the motion-to-text task, refers to generating a text description for a given human motion sequence. TM2T [24] and MotionGPT [30] are two recently introduced models for motion captioning. In this work, we use MotionGPT to caption the generated motion sequences due to its superior performance. MotionGPT is a motion-language model that is trained on large amounts of language data and motion data to handle numerous motion-relevent tasks such as text-driven motion synthesis, motion captioning, motion prediction, and motion in-betweening.

To caption a given input motion sequence of length M frames $m^{1:M} = \{x_i\}_{i=1}^M$, the motion sequence is first encoded into L discrete motion tokens $z^{1:L} = \varepsilon(m^{1:M}) = \{z_i\}_{i=1}^L, L = M/l$, using a motion encoder ε , where l is the temporal downsampling rate on motion length. The motion encoder is part of a motion tokenizer that is based on the Vector Quantized Variational Autoencoders (VQ-VAE) architecture [77]. This allows the motion sequence to be represented as a language. The motion encoder consists of two parts: 1D convolutions and quantization. In the first part, 1D convolution layers are applied to the motion sequence $m^{1:M}$ along the time dimension to obtain the latent vectors $\hat{z}_{1:L}$. Through discrete quantization, the latent vectors are converted into code indices using a learnable codebook $Z = \{z_i\}_{i=1}^K \subset \mathbb{R}^d$ with K latent embedding vectors of dimension d. During quantization, each latent vector is replaced with the nearest vector in Z, which minimizes the Euclidean distance:

$$z_i = \arg\min_{z_k \in Z} \|\hat{z}_i - z_k\|_2 \tag{6}$$

Here, z_i is the quantized latent vector. The code indices corresponding to the quantized latent vectors in the codebook are called motion tokens, which can be interpreted as the vocabulary for human motion.

In addition to taking in a motion sequence as input, MotionGPT also processes a textual prompt describing the specific task to be performed, in this case, motion captioning. Similar to the input motion sequence, the input textual prompt is tokenized using the text encoder of a pre-trained text tokenizer, SentencePiece [34], with a vocabulary of K_t text tokens. The text vocabulary $V_t = \{v_t^i\}_{i=1}^{K_t}$ is combined with the motion vocabulary $V_m = \{v_m^i\}_{i=1}^{K_m}$, which includes motion tokens and additional special tokens (e.g., the tokens indicating the start and end of a motion sequence) to form the unified vocabulary $V = \{V_t, V_m\}$ of a language model. The tokens/words within the new vocabulary can represent text, human motion, or a mixture of the two. This flexibility in representation allows MotionGPT to use both text and motion as input and output to accomplish a range of motion-related tasks with a single model.

The sequence of text and motion tokens $X_s = \{x_s^i\}_{i=1}^N$, $x_s \in V$, are passed to a transformer-based language model as input. The language model generates a sequence of output tokens $X_t = \{x_t^i\}_{i=1}^L$, in an autoregressive manner, and it is trained using the following loss function

$$\mathcal{L}_{LM} = -\sum_{i=0}^{L_t - 1} \log p_{\theta}(x_t^i | x_t^0, \dots, x_t^{i-1}, x_s)$$
 (7)

Lastly, the sequence of output tokens is decoded using a text decoder and a motion decoder to recover the output texts, motion caption in this case.

In our implementation, the codebook size of the motion encoder is set to $Z \in \mathbb{R}^{512x512}$. Additionally, the motion encoder uses a temporal downsampling rate of 4. T5 [62] is chosen as the transformer-based language model with 12 layers in both the encoder and decoder. The model is trained on the HumanML3D dataset [23], a dataset containing large amounts of human motion capture data along with the corresponding textual descriptions, using the AdamW optimizer. We use the pre-trained model that Jiang et al. [30] released.¹

3.2.2 Activity Filtering Using Large Language Models. The problem is framed as a binary classification task where the input is a textual description m, the motion caption of a human motion sequence, along with a specific activity of interest a. The output is a binary label $y \in \{0, 1\}$ that indicates whether the input textual description accurately depicts a person performing the activity in question. Let $\mathcal M$ represent the space of all possible textual motion captions, and let $\mathcal A$ represent the set of all predefined activities of interest. We seek to learn a function f that maps a pair consisting of a motion caption and an activity to a binary label: $f: \mathcal M \times \mathcal A \to \{0, 1\}$. The goal of f is to determine whether a given motion caption $m \in \mathcal M$ correctly describes a person performing a given activity $a \in \mathcal A$. With conventional supervised methods in NLP, learning such a function would require large amounts

¹https://motion-gpt.github.io

of training data containing motion captions for specific activities. However, collecting such a dataset would be extremely time-consuming and practically infeasible.

Hence, we decided to turn to LLMs for this problem as they have shown impressive performances on zero-shot NLP tasks [13, 32]. Through our early explorations, we believe the function f has been implicitly learned by the LLMs during the training process due to the massive training corpus, as we empirically found that the LLMs have a good understanding of the correspondence between the motion descriptions and activities.

To obtain labels from the LLM, we first assign it a task via a system message, which is a type of message used to define the role of the LLM. In this message, the LLM is asked to provide 'yes' or 'no' responses, indicating whether the user-provided motion captions describe the specified activity accurately. Using this system message helps the LLM understand its role and significantly reduces post-processing effort. Without this message, we found that the LLM often produces additional, miscellaneous texts unrelated to the task. After setting up the system message, we provide the specified activity name and a list of motion captions, with 10 captions per prompt, to the LLM. The LLM then outputs 'yes' or 'no' for each caption. A 'yes' response indicates that the caption correctly describes someone performing the specified activity, and 'no' indicates otherwise. We use the LLM's responses to filter out incorrectly generated motion sequences.

This approach is a zero-shot task for the LLM, as we do not provide any example captions and labels for it to learn from. We detail the exact prompts used, along with some example motion captions and the corresponding labels generated by the LLM, in Figure 4 in Appendix A. For the labels shown in Figure 4, we used GPT-4 [57] as our LLM. It's important to note that the LLM is not infallible; for example, it incorrectly labeled the caption 'a person jogs in place then stops' as 'no' for the running activity. We evaluate how well the proposed motion filter can filter out incorrectly generated motion sequences in section 4.5.

4 ON THE EFFECTIVENESS OF LANGUAGE-BASED CROSS MODALITY TRANSFER FOR HAR

As the second major contribution, we conduct an extensive study on the–now extended–language-based cross modality transfer approach: we run a large-scale experimental evaluation of IMUGPT 2.0 that specifically focuses on the practical aspects of this new cross modality transfer paradigm with the goal of assessing its relevance for real-world HAR applications. We run our study in two parts. The first part evaluates the original IMUGPT, i.e., the first language-based cross modality transfer approach, without our newly proposed components. In contrast to the pilot study conducted by Leng et al. [43], this experimental evaluation is on a much larger scale with larger numbers of activities compared to the previous study [43]. We conduct experiments using different LLMs (section 4.2) and motion synthesis models (section 4.3) for generating textual descriptions and motion sequences. Through these experiments, we examine the impact that different generative models may have on downstream HAR performance and, in turn, reveal insights into the suitability of models, which will help practitioners design language-based cross modality transfer systems.

In the second part of our study, we conduct an evaluation using our newly proposed components, i.e., the motion filter and diversity metric, thereby building on the results of part 1 of our study. We start with the proposed diversity metrics and validate our hypothesis that diversity in the generated textual descriptions and motion sequences can serve as a predictor for downstream model performance and evaluate the effectiveness of our proposed saturation point identification algorithm (section 4.4). Following this, we evaluate the new motion filter and determine how effectively it can filter out incorrectly generated motion sequences and its impact on the downstream model performance (section 4.5).

4.1 Human Activity Recognition

4.1.1 Datasets. In line with previous work in the field, we conduct our evaluation on five public HAR datasets: i) RealWorld [72]; ii) PAMAP2 [65]; iii) USC-HAD [89]; iv) HAD-AW [55]; and v) MyoGym [33]. These datasets

contain IMU data recorded from varying on-body locations like head, chest, arm, waist, and leg for daily activities. Each dataset covers different activities and a varying number of subjects as summarized in Table 10 in Appendix B.

4.1.2 Classification Model. Our evaluations are based on three models that are widely used in the HAR research community: i) a Random Forest classifier; ii) DeepConvLSTM [58]; and iii) DeepConvLSTM with self-attention [71]. We use sliding windows two seconds long with a 50% overlap between consecutive frames to split the IMU data. We trained the Random Forest classifier on ECDF features [25] (15 components). We trained DeepConvLSTM and DeepConvLSTM with self-attention on IMU raw data for a maximum of 30 epochs with an Adam optimizer and a ReduceLROnPlateau learning rate scheduler [3]. We used grid search to determine the learning rate and weight decay. The learning rate varied from 10^{-6} to 10^{-2} and the weight decay varied from 10^{-4} to 10^{-3} . For RealWorld, PAMAP2, USC-HAD, and MyoGym datasets, we used leave-one-subject-out cross validation for all 3 models. For the HAD-AW dataset, we used 5-fold stratified cross-validation instead since not all subjects performed the entire set of activities in the released dataset. For each model, we repeat the cross-validation for 3 random seeds and report the average macro F1 score and the standard deviation across the three runs.

4.2 Impact of LLM on Textual Description Generation and HAR

We explored various LLMs to explore their capability to generate textual descriptions regarding diverse human activities, and their effect on downstream classifier performance.

4.2.1 Experimental Setting. We generated motion descriptions using five LLMs: i) GPT-3.5 [2]; ii) GPT-4 [57]; iii) Palm 2 (Bard) [5]; iv) Gemini [74]; and v) LLaMa 2 [75]. All LLMs, except LLaMa 2, were available as APIs, while the LLaMa 2 model with 70 billion parameters was deployed and hosted on a server for use as an API. An automated pipeline was developed with parameters for the type of model and the dataset. We briefly experimented with each LLM to determine the optimal prompt for the generation of text descriptions. The activities from all five datasets were provided to the LLMs as a text, along with sample descriptions and the LLMs were asked to describe a person performing the activity.

The generation of 1,000 descriptions for each activity was done in batches of 50, as descriptions for larger batches resulted in errors. The resulting responses were parsed programmatically to remove empty lines, serial numbers, and responses containing auxiliary text like "Here are the descriptions for the activity". The cleaned descriptions were passed into the motion synthesis models to create motion sequences, and the virtual IMU data was extracted, following the IMUGPT pipeline. The sizes of the real IMU dataset and the virtual IMU datsets are shown in Table 11 of Appendix C.

4.2.2 Qualitative Evaluation of Generated Text. Examples of the generated textual descriptions are listed in Table 14 in Appendix D. We also provide all generated textual descriptions in the supplementary material for this paper. The generated textual descriptions were diverse on first look but an analysis revealed interesting results. On programmatic evaluation, GPT-3.5, GPT-4, and LLaMa 2 generated descriptions for the activities with minimal errors. Examples of errors include: i) inability to generate descriptions – the LLM responds with "Sorry, I could not generate descriptions for the activity"; ii) incorrect activities described – the LLM generate text descriptions that did not describe the activity, instead described an emotion associated with the activity. On the contrary, Palm 2 and Gemini had conflicting results. In some instances, the text described inanimate objects (sponge, rocket) and animals (cat, dog) performing the activities. In other cases, the models generated highly repetitive descriptions for a particular activity. The generation of prompts with Palm 2 and Gemini were also tedious with a larger number of failures, irrelevant text, and missing text. This was primarily noticeable in complex activities like workout-related activities present in MyoGym. Overall, GPT-3.5, GPT-4, and LLaMa 2 produced better textual descriptions of activities and expect the generated virtual IMU data to lead to better downstream performance compared to Palm 2 and Gemini.

Table 2. Model performance (Macro F1 score) when trained on both the real IMU data and the virtual IMU data when various LLMs are used to generate the textual descriptions in IMUGPT. "Real Data" denotes the baseline experiments not including any generated, virtual IMU data.

	RealWorld	PAMAP2	USC-HAD	HAD-AW	MyoGym				
Random Forest Classifier									
GPT-3.5	79.70 ± 0.38	69.20 ± 0.29	49.72 ± 0.67	48.98 ± 0.11	47.93 ± 0.58				
GPT-4	79.41 ± 0.29	67.73 ± 0.70	49.07 ± 0.27	48.61 ± 0.20	47.64 ± 0.17				
LLaMa 2	79.02 ± 1.01	67.82 ± 0.30	49.41 ± 0.28	48.40 ± 0.08	47.95 ± 0.25				
Palm 2 (Bard)	78.79 ± 0.65	68.42 ± 0.43	49.06 ± 0.31	49.36 ± 0.05	47.91 ± 0.25				
Gemini	78.93 ± 0.17	67.53 ± 0.10	49.65 ± 0.51	48.70 ± 0.06	47.98 ± 0.22				
Real Data	71.53 ± 1.07	67.08 ± 0.53	47.23 ± 0.08	52.77 ± 0.07	46.61 ± 0.09				
		Deep Co	nvLSTM						
GPT-3.5	82.25 ± 0.32	75.16 ± 0.82	61.44 ± 0.49	51.98 ± 0.28	49.46 ± 0.88				
GPT-4	80.20 ± 0.88	73.59 ± 0.75	61.43 ± 0.59	51.11 ± 0.28 50.94 ± 0.17	46.85 ± 0.28 48.78 ± 0.45				
LLaMa 2	82.33 \pm 0.34 74.12 \pm	74.12 ± 0.88	60.93 ± 0.39						
Palm 2 (Bard)	81.74 ± 0.48	74.66 ± 0.96	61.17 ± 0.44	51.03 ± 0.11	49.00 ± 0.39				
Gemini	80.86 ± 0.67	72.76 ± 0.37	61.02 ± 0.15	51.44 ± 0.49	48.82 ± 0.31				
Real Data	77.79 ± 0.85	69.26 ± 1.07	63.35 ± 0.67	56.16 ± 0.38	50.69 ± 0.61				
	De	ep ConvLSTM v	with self attenti	on					
GPT-3.5	80.47 ± 0.49	73.45 ± 0.85	59.12 ± 0.62	47.67 ± 0.73	47.35 ± 0.41				
GPT-4	80.31 ± 0.36	73.68 ± 1.26	57.97 ± 0.68	47.64 ± 0.50	46.03 ± 0.29				
LLaMa 2	80.36 ± 0.67	73.77 ± 0.85	59.09 ± 0.89	48.75 ± 1.07	47.58 ± 0.47				
Palm 2 (Bard)	80.77 ± 0.72	73.23 ± 0.92	58.66 ± 1.01	47.03 ± 0.90	47.21 ± 0.56				
Gemini	80.82 ± 0.61	72.89 ± 0.34	58.84 ± 0.33	47.99 ± 0.06	47.10 ± 0.28				
Real Data	77.50 ± 0.76	64.36 ± 0.52	61.82 ± 0.82	56.51 ± 0.19	50.63 ± 0.88				

4.2.3 Results. Table 2 shows the downstream performance when different LLMs are used in IMUGPT for virtual IMU data generation. Overall, we find that the downstream performance is best when GPT-3.5 is used to generate the textual descriptions. With the Random Forest classifier, the downstream performance of GPT-3.5 shows an overall improvement of 1.0% over GPT-4, 0.9% over LLaMa 2, 0.6% over Palm 2, and 0.8% over Gemini across all the datasets. Therefore, we will use GPT-3.5 for generating textual descriptions in subsequent experiments unless otherwise specified. We were surprised that Gemini and Palm 2 achieved competitive downstream performance (less than 1% difference compared to the other three LLMs), despite them producing qualitatively worse textual descriptions with less context information (as discussed in section 4.2.2). We discuss the effect of additional context information within the textual descriptions in section 5.1.

When using the Random Forest classifier, the generated virtual IMU data led to significantly better downstream performance for all datasets but one, HAD-AW. For the deep learning models, Deep ConvLSTM and Deep ConvLSTM with self-attention, the results were mixed. The virtual IMU data did not significantly improve model performance for USC-HAD, HAD-AW, and the MyoGym datasets. We attribute this to the deep learning models overfitting to the virtual IMU data, where the models learned features only applicable to the virtual IMU, hence, the deep learning models were unable to generalize to the testing dataset.

4.3 Impact of Motion-Synthesis Models on HAR

In addition to experimenting with different LLMs for generating textual descriptions of activities, we also experimented with using a variety of motion synthesis models to convert the textual descriptions into human

motion sequences in order to determine how different generated motion sequences affect downstream activity recognition.

4.3.1 Experiment Setting. We convert 1,000 textual descriptions, generated by GPT-3.5, to motion sequences by using the following four motion synthesis models: i) T2M-GPT [86]; ii) MotionGPT [30]; iii) MotionDiffuse [87]; and iv) ReMoDiffuse [88]. Both T2M-GPT and MotionGPT are Variational-Auto-Encoder (VAE)-based pipelines. In these VAE-based pipelines, the input textual descriptions are tokenized using a text tokenizer. The text tokens are then used by a transformer to autoregressively generate motion tokens, which are converted into motion sequences using a motion decoder. On the other hand, MotionDiffuse and ReMoDiffuse are diffusion-based models. In these models, the encoded textual descriptions are used in a reverse diffusion process, where Gaussian noise is gradually denoised to produce motion sequences.

The main difference between VAE-based and diffusion-based models lies in the length of the generated motion. In VAE-based models, a trained stopping token exists, and the generation process stops after this token is generated, allowing motion sequences to vary in length depending on the input textual description. However, diffusion-based models do not have a stopping token, so all generated motion sequences have a predefined fixed length. In our experiments, we set the motion length to be six seconds, following previous work [88], for the diffusion-based models. With a fixed length, we observed that some generated motion sequences ended prematurely, as six seconds was not sufficient to portray the motion defined in the input textual description. Conversely, for some motion sequences, six seconds proved too long, resulting in repeated or miscellaneous motions at the end of the sequence. We did not encounter such issues with VAE-based models, as the generated motion sequences ended naturally.

The sizes of the real IMU dataset and the virtual IMU datasets are shown in Table 12 of Appendix C. The classification models are trained on either real IMU data alone or on both the real IMU data and the virtual IMU data generated by IMUGPT using the four motion synthesis models.

4.3.2 Results.

Table 3 shows the downstream activity recognition results when different motion synthesis models are used in IMUGPT for virtual IMU data generation. Overall, we observe that T2M-GPT generally performs better for most of the datasets and across different classifier models. With the Random Forest classifier, the downstream performance of T2M-GPT shows an overall improvement of 0.7% over MotionGPT, 1.2% over MotionDiffuse, and 2.2% over ReMoDiffuse across all the datasets (all differences are statistically significant). In some instances, other motion synthesis models return better results mainly with the DeepConvLSTM model with self-attention. However, for MyoGym, ReMoDiffuse appears to be the better motion synthesis model across all the three classifier models. Due to the complex activities in HAD-AW, its results are worse than for real data.

4.4 Diversity as a Predictor for HAR Model Performance

We made two assumptions that form the basis for drawing a connection between the diversity of text and the downstream HAR performance. Our first hypothesis is that diversity in textual descriptions is correlated to diversity of the motion sequences generated by those descriptions. The second hypothesis is that diversity in motion sequences is correlated to performance for models trained on the virtual data obtained from those sequences. We validate these hypotheses by computing the correlation between the diversity of the textual descriptions and the motion sequences, and subsequently between motion sequences and final classifier performance. We also evaluate whether there exists a correlation between the textual descriptions and the model performance.

4.4.1 Correlations. Using comparative diversity (section 3.1.2), we compute the Pearson correlation coefficient between text diversity, motion diversity, and the change in the F1 score downstream. This process is similar to our saturation point identification algorithm (algorithm 1). For each activity, we start with a set of 50 text descriptions,

Table 3. Model performance (Macro F1) when trained on both the real IMU data and the virtual IMU data when various motion synthesis models are used to generate the motion sequences in IMUGPT. "Real Data" denotes the baseline experiments not including any generated, virtual IMU data.

	RealWorld	PAMAP2	USC-HAD	HAD-AW	MyoGym			
Random Forest Classifier								
T2M-GPT	79.70 ± 0.38	69.20 ± 0.29	49.72 ± 0.67	48.98 ± 0.11	47.93 ± 0.58			
MotionGPT	79.19 ± 0.71	68.00 ± 0.28	49.72 ± 0.37	48.74 ± 0.03	47.73 ± 0.62			
MotionDiffuse	79.23 ± 0.40	68.15 ± 0.43	48.91 ± 0.49	49.19 ± 0.09	46.77 ± 0.21			
ReMoDiffuse	74.32 ± 0.87	67.81 ± 0.11	46.72 ± 0.30	50.05 ± 0.06	49.31 ± 0.27			
Real Data	71.53 ± 1.07	67.08 ± 0.53	47.23 ± 0.08	52.77 ± 0.07	46.61 ± 0.09			
		Deep Cor	ıvLSTM					
T2M-GPT	82.25 ± 0.32	75.16 ± 0.82	61.44 ± 0.49	51.98 ± 0.28	49.46 ± 0.88			
MotionGPT	81.45 ± 0.75	73.39 ± 0.53	60.72 ± 0.73	50.49 ± 0.15	46.35 ± 0.48			
MotionDiffuse	82.03 ± 0.68	74.44 ± 0.75	61.07 ± 0.30	53.00 ± 0.30	46.78 ± 0.23			
ReMoDiffuse	78.61 ± 0.45	73.80 ± 0.58	61.11 ± 0.86	53.49 ± 0.36	51.93 ± 0.57			
Real Data	77.79 ± 0.85	69.26 ± 1.07	63.35 ± 0.67	56.16 ± 0.38	50.69 ± 0.61			
	Dec	ep ConvLSTM v	vith self attentio	on				
T2M-GPT	80.47 ± 0.49	73.45 ± 0.85	59.12 ± 0.62	47.67 ± 0.73	47.35 ± 0.41			
MotionGPT	80.31 ± 0.47	72.20 ± 0.97	58.82 ± 0.59	46.96 ± 0.73	44.99 ± 0.51			
MotionDiffuse	81.58 ± 0.22	73.29 ± 0.14	59.71 ± 0.58	51.19 ± 0.44	45.66 ± 0.09			
ReMoDiffuse	77.65 ± 0.16	72.17 ± 1.24	60.71 ± 0.44	50.08 ± 0.82	48.49 ± 0.80			
Real Data	77.50 ± 0.76	64.36 ± 0.52	61.82 ± 0.82	56.51 ± 0.19	50.63 ± 0.88			

adding 5% more textual descriptions at each step, and calculate the MMD between the two sets. Corresponding motion sequences and virtual IMU data are generated at each step. For the motion sequences, we calculate the MMD in a similar manner. We then use the virtual IMU data to train classification models, as in previous experiments, and compare the performance to models trained on only real IMU data to obtain the per-class change in F1 score. This process is repeated until reaching the saturation point (algorithm 1). Ultimately, we have a list of values for text diversity, motion diversity, and changes in F1 scores, and we compute the correlation between these three factors.

For each dataset, we show the average correlation across all activities (Table 4). We find that text and motion diversity are highly correlated. For all but one dataset, HAD-AW, there is a negative moderate to strong correlation between the diversities and the downstream change in F1 score. We note that the correlation is negative because a lower MMD indicates higher diversity. This suggests that the diversity metric can serve as a downstream performance predictor, where higher diversity indicates better performance. Specifically, the RealWorld dataset showed the strongest correlation, as the virtual IMU data contributed to improvements in the per-class F1 score across all activities. In contrast, for the USC-HAD, PAMAP2, and MyoGym datasets, the virtual IMU data led to a decline in the per-class F1 score for some activities, resulting in a more moderate correlation. For the HAD-AW dataset, a positive correlation was observed, indicating that the virtual IMU data led to a drop in the per-class F1 score for more activities than it helped.

4.4.2 Saturation Point Identification Algorithm Evaluation. In order to evaluate our saturation point determination algorithm (algorithm 1), we use it to generate text descriptions for each activity across all the datasets. Our goal is to evaluate the set of descriptions generated by utilizing the algorithm, as opposed to the case when it is not used (i.e., textual descriptions are generated directly without accounting for diversity saturation). Therefore, we

Table 4. Correlations between comparative diversity of textual prompts, motion sequences, and the change downstream activity recognition performance (measured in macro F1 scores).

Dataset	Dataset RealWorld		USC-HAD	HAD-AW	MyoGym
Text V.S. Motion	$r = 0.92, p \le 0.001$	$r = 0.87, p \le 0.001$	$r = 0.91, p \le 0.001$	$r = 0.88, p \le 0.001$	$r = 0.91, p \le 0.001$
Text V.S. F1 $r = -0.77, p \le 0.001$		r = -0.31, p = 0.0052	r = -0.45, p = 0.004	r = 0.22, p = 0.173	r = -0.24, p = 0.136
Motion V.S. F1	$r = -0.76, p \le 0.001$	r = -0.32, p = 0.044	r = -0.46, p = 0.003	r = 0.21, p = 0.193	r = -0.24, p = 0.136

Table 5. Comparison of Model Performance (Macro F1 Score) Using Real and Virtual IMU Data, with and without the saturation Point Identification Algorithm.

	RealWorld	PAMAP2	USC-HAD	HAD-AW	MyoGym			
	Rand	lom Forest Cla	ssifier					
Without Saturation Point	79.70 ± 0.38	69.20 ± 0.29	49.72 ± 0.67	48.98 ± 0.11	47.93 ± 0.58			
With Saturation Point	80.39 ± 0.34	69.50 ± 0.54	50.07 ± 0.10	50.62 ± 0.05	48.73 ± 0.15			
	Deep ConvLSTM							
Without Saturation Point	82.25 ± 0.32	75.16 ± 0.82	61.44 ± 0.49	51.98 ± 0.28	49.46 ± 0.88			
With Saturation Point	82.70 ± 0.49	75.47 ± 1.15	61.36 ± 0.38	52.37 ± 0.27	48.74 ± 0.36			
Deep ConvLSTM with self attention								
Without Saturation Point $\mid 80.47 \pm 0.49 \mid 73.45 \pm 0.85 \mid 59.12 \pm 0.62 \mid 47.67 \pm 0.73 \mid 47.35 \pm 0.$								
With Saturation Point	81.11 ± 1.40	73.01 ± 0.45	59.79 ± 0.26	49.64 ± 0.32	47.53 ± 0.41			

use the results from section 4.2, where 1,000 textual descriptions were generated, as a baseline. Consequently, we obtain two sets of text descriptions for each activity across all datasets: one set consisting of 1,000 descriptions and another set generated using algorithm 1, which includes descriptions produced up to the saturation point.

Subsequently, we generate virtual IMU data from these sets of textual descriptions, combine them with the corresponding real datasets, and use this data to train downstream classifiers. Downstream activity recognition results are shown in Table 5. We observe that across all activities, datasets, and classifiers the performance in the Saturation Point case is similar to, if not better than, without using saturation point. Note that the saturation points are computed for each activity individually. Most of the saturation points across all activities and datasets fall within the range of around 400 to 600 textual descriptions, indicating that the algorithm stops generation after that point. This suggests that we can achieve equivalent performance while utilizing around 50% less data compared to directly generating 1,000 descriptions. Therefore, we can save at least 50% of the time and compute resources needed for data generation. This algorithm also provides a formal structure to guide the process of generation in a manner that makes it consistent and repeatable – in the absence of this algorithm, it would be impossible to determine how much textual descriptions should be generated, and at which point should generation be halted.

4.5 Motion Filter Evaluation

In this section, we evaluate the effectiveness of the newly introduced motion filter with regards to eliminating irrelevant motion sequences, and how it affects the downstream activity recognition performance.

4.5.1 Experimental Setting. We used the RealWorld dataset [72] for our evaluation. For each of the eight activities within the dataset, we first generated 50 textual descriptions of the activity using GPT-3.5 [2], and then used T2M-GPT [86] to convert these descriptions into human motion sequences. We visualized these motion sequences as

Table 6. Performance comparison of GPT-3.5, GPT-4, and human annotators in filtering out incorrectly generated motion sequences using motion captions. The percentage of incorrectly generated motion sequences before and after applying the motion filter is indicated as '% before' and '% after' the filter, respectively.

Annotator Precision		Recall	Accuracy	F1	% Before Filter	% After Filter
GPT-3.5	70.56 ± 0.84	71.49 ± 2.01	59.95 ± 1.01	67.93 ± 1.34	35.50%	$29.44\% \pm 0.84\%$
GPT-4	90.08 ± 0.63	60.55 ± 2.08	70.40 ± 0.88	71.12 ± 1.46	35.50%	$9.92\% \pm 0.63\%$
Human	91.63	72.28	77.00	79.47	35.50%	8.37%

3D animations. Some example visualizations are shown in Figure 5 of Appendix E. Using these visualizations, we manually annotated whether the sequences accurately portray the activity of interest. These manual annotations will serve as our ground truths for evaluating motion filtering performance.

The generated motion sequences are then processed with the motion filter to distinguish the relevance of the motion sequence. Specifically, we used GPT-3.5 and GPT-4 for our motion filter to further study the impact of LLMs on filtering performance. Besides using LLMs for labeling motion captions, we manually annotated the captions generated by our motion caption model [30] to compare human and LLM performance The performance of our motion filter was evaluated with precision, recall, accuracy, F1 score, and percentage of incorrectly generated motion sequences before and after filtering. For each LLM, we repeat the experiment five times on the same set of motion captions and report averages and standard deviations for each performance metrics.

4.5.2 Validation Results. Table 6 shows the results for the motion filter using GPT-3.5, GPT-4, and human annotators to filter out incorrectly generated motion sequences using motion captions. The goal of the motion filter is to maximize true negatives (correctly identified inaccurate motion sequences) and minimize false positives (inaccurate sequences that escape the motion filter). Therefore, an effective motion filter should have high precision. We observe that the choice of LLM greatly impacts the motion filter's performance. GPT-4, with a precision of 0.901, significantly outperformed GPT-3.5, which had a precision of 0.706, comparable to human performance. When GPT-4 is used for filtering, the percentage of incorrectly generated motion sequences within the remaining dataset is greatly reduced, from 35.5% before filtering to 9.9% after.

We note that both the LLMs and human annotators cannot achieve perfect performance in the filtering process using motion captions alone. This limitation stems from the fact that the motion captions generated by the motion caption model can sometimes be ambiguous and fail to accurately describe the input motion sequence.

4.5.3 Activity Recognition Performance. We now examine the impact of the motion filter on the downstream activity recognition performance. We start with the motion sequences generated by T2M-GPT based on the text descriptions generated by GPT-3.5. The captions of the motion sequences are input into GPT-4, which filters out incorrectly generated motion sequences. Table 13 in Appendix C shows the size of the virtual IMU datasets after filtering. The classification models are then trained on the filtered datasets. The classification results, as detailed in Table 7, indicate that using the motion filter leads to better downstream performance for the HAD-AW and MyoGym datasets, with relative performance improvements of 4.3% and 4.1% for HAD-AW and MyoGym, respectively, compared to not using the filter (all statistically significant differences).

However, for the other three datasets, the motion filters did not significantly impact downstream performance. This was surprising and contrary to our hypothesis and the motion filter validation results. We identified that motion filtering tends to reduce the diversity of the generated sequences, favoring similar kinds of motion sequences, potentially due to the biases in LLMs, which likely contributed to the observed decrease in performance.

Table 7. Comparison of Model Performance (Macro F1) Using Real and Virtual IMU Data, with and without motion filter.

	RealWorld	PAMAP2	USC-HAD	HAD-AW	MyoGym					
	Random Forest Classifier									
Without Motion Filter	79.70 ± 0.38	69.20 ± 0.29	49.72 ± 0.67	48.98 ± 0.11	47.93 ± 0.58					
With Motion Filter	79.61 ± 0.44	70.22 ± 0.18	49.73 ± 0.14	51.09 ± 0.17	48.24 ± 0.19					
	Deep ConvLSTM									
Without Motion Filter	82.25 ± 0.32	75.16 ± 0.82	61.44 ± 0.49	51.98 ± 0.28	49.46 ± 0.88					
With Motion Filter	81.45 ± 0.83	74.89 ± 1.14	61.75 ± 0.37	52.36 ± 0.20	51.94 ± 0.65					
Deep ConvLSTM with self attention										
Without Motion Filter 80.47 ± 0.49 73.45 ± 0.85 59.12 ± 0.62 47.67 ± 0.73 47.35 ± 0.62										
With Motion Filter	81.66 ± 1.25	73.52 ± 0.46	60.67 ± 0.56	51.44 ± 1.10	50.50 ± 0.25					

5 DISCUSSION

Based on the investigations presented in this paper, we gain a range of insights into the overall process of language-based cross modality transfer and in IMUGPT 2.0 in particular. We detail these in the subsequent sections, highlighting some notable aspects of the system and providing potential directions for future investigations.

First, we study the effect of the presence of additional context information in the textual descriptions on the downstream classifier performance (section 5.1). Then, we investigate if the presence of multiple IMU sensors results in better performance gain from the addition of virtual IMU data (section 5.2). Subsequently, we evaluate the effect of the motion filtering mechanism on the diversity of virtual IMU data (section 5.3). Finally, we highlight the limitations of the extended system and offer some directions for future research (section 5.4).

5.1 How much impact does context information have on downstream HAR performance?

As reported in the results (Table 2), while GPT-3.5 had the best overall performance for the datasets, the other LLMs were not far behind. This raises the question: *To what extent does the context information within the generated textual descriptions affect downstream activity recognition performance?* To evaluate this, we modified the prompts for GPT-3.5 to generate text descriptions with additional context information. The intuition behind using context information is that these factors could affect the actual motions in the activity performed, thereby generating varying virtual IMU data. Arguably, a wider range of contextual factors can influence the generation of textual descriptions. While an exhaustive evaluation is challenging to conduct (and beyond the scope of this paper), we explore the, in our opinion, most important contextual parameters that may have an impact on generated textual descriptions, and thus, the generated movement data:

Age: The age of the person performing a particular activity could influence the motion (e.g., a toddler vs adult). **Physique:** Similar to age, the physique of a person could also affect the effort required and, hence, the motion of the activity being performed (e.g., skinny vs muscular).

Weather: Weather conditions on a particular day can affect activities performed outdoors. Especially in extreme conditions, it could make performing the activity more challenging (e.g., sunny vs thunderstorm).

We passed different combinations of these parameters as prompts to generate text descriptions, following the same pipeline as in our original experiments. Example textual descriptions can be found in Table 15 of Appendix D. The results of the experiments with varying context information are displayed in Table 8. We observe no significant difference in downstream performance compared to the experiments without context information. This could be because the motion synthesis model is trained on motion capture data that are recorded indoors, so the weather-related context may not be represented in the generated motion sequences. Similarly, the textual descriptions within the HumanML3D dataset [23], on which the motion synthesis models are trained, do

Table 8. Performance (Macro F1) of GPT-3.5 on RealWorld dataset with T2M-GPT motion synthesis model when parameters are added to generate text descriptions with context information

Model	No parameters	Age	Weather	Physique	Age &	Age &		Age, Weather
					Weather	Physique	Physique	& Physique
Random Forest	79.70 ± 0.38	79.09 ± 0.45	79.58 ± 0.87	79.27 ± 0.86	79.12 ± 0.43	79.48 ± 0.52	78.84 ± 0.30	79.32 ± 0.77
Deep ConvLSTM	82.25 ± 0.32	81.33 ± 0.50	80.64 ± 0.78	82.19 ± 0.56	81.20 ± 0.22	80.61 ± 0.25	81.00 ± 0.36	81.67 ± 0.60
Deep ConvLSTM self attention	80.47 ± 0.49	80.90 ± 0.42	79.72 ± 0.70	80.50 ± 0.44	79.89 ± 0.36	79.80 ± 0.73	80.50 ± 0.47	80.94 ± 1.00

not contain additional information related to age and physique. Consequently, the generated motion sequences may not accurately reflect the differences in motion that these context parameters would introduce.

5.2 Do multi-sensor setups benefit more from the use of virtual IMU data?

In our experiments (Table 2 and Table 3), we observe that the performance of downstream classifiers trained on the RealWorld and PAMAP2 datasets improves significantly with the addition of virtual IMU data. This trend holds across all three classifier models (i.e., Random Forest Classifier, Deep ConvLSTM and Deep ConvLSTM with self attention). Conversely, adding virtual IMU data does not equally improve the HAR performance for HAD-AW, and the performance for USC-HAD and MyoGym only improves for the Random Forest Classifier. One of the key differences between the datasets is the number of IMU sensors for each dataset. While RealWorld has six and PAMAP2 has three sensors, the USC-HAD, HAD-AW and MyoGym datasets only have a single sensor.

If multiple sensors are used to record an activity, certain relationships would exist between all the sensors, since all the data would be drawn from the same motion. We attribute the improved performance to this factor – adding virtual IMU data aids the downstream classifier in learning these relationships, which leads to better results. The datasets which contain just a single sensor are unable to benefit from this aspect since no such relationship can exist with just one sensor.

5.3 How does the motion filter affect data diversity?

In section 4.5, we observed that for three datasets–RealWorld, PAMAP2 and USC-HAD–the application of the motion filter does not lead to a significant improvement in downstream performance (Table 7), despite the fact that the motion filter is able to remove irrelevant motion sequences as shown in our evaluation (Table 6).

We hypothesize that while the filtering process does improve the relevance of the generated virtual IMU data, it also leads to a reduction in diversity. Both of which affect the downstream performance. As highlighted in section 4.4, a lower diversity would lead to poorer performance. Therefore, we investigate the effect of motion filtering on the diversity of the motion sequences by utilizing the absolute diversity metrics discussed in section 3.1.1. For each activity, we compute the standard deviation and centroid diversity metrics for the unfiltered motion sequences as well as the motion sequences retained after filtering.

The diversity scores are listed in Table 9. We observe that across all datasets and metrics, the diversity of the motion sequences decreases after the application of the filter. Thus, while the filtering process is able to improve the relevance of the virtual IMU data, it also brings down the diversity. This reduction in diversity counteracts the benefit provided by the increased relevance of the data. As a result, the downstream performance does not change significantly after motion filtering.

5.4 Limitations and Future Work

5.4.1 Lack of Expressivity of Motion Synthesis Model. As shown in our evaluation (Table 2), the generated virtual IMU data did not lead to improvements in downstream performance for the HAD-AW dataset. We identify two reasons for this: i) the motion synthesis models were unable to generate some of the activities within the

Table 9. Comparison of absolute diversity metrics for motion embedding sequences, with and without motion filter

Dataset	RealWorld	PAMAP2	USC-HAD	HAD-AW	MyoGym		
Standard Deviation Metric							
Without Motion Filter	0.0487	0.0393	0.0392	0.0308	0.0397		
With Motion Filter	0.0358	0.0305	0.0307	0.0259	0.0312		
Centroid Metric							
Without Motion Filter	0.0037	0.0027	0.0026	0.0014	0.0023		
With Motion Filter	0.0021	0.0017	0.0017	0.0011	0.0014		

HAD-AW dataset; *ii*) the used motion-to-sensor method, IMUSim [85], is unable to capture the subtle movement characteristics present in the real IMU data.

Since the motion synthesis models are trained on the HumanML3D dataset, they can only generate motion sequences for the activities that are present within this dataset. Many activities in the HAD-AW, such as 'washing hands' and 'drawing', are not included in the HumanML3D dataset. This led to all the motion sequences generated for these activities being irrelevant. While the motion filter was able to filter out most of these irrelevant motion sequences, some remained in the filtered dataset, leading to worse downstream performance.

Recently, a new motion capture dataset, Motion-X [47], has been introduced. Motion-X contains three times more motion capture data than HumanML3D and includes a more diverse range of activities, both indoor and outdoor. We expect a motion synthesis model trained on the Motion-X dataset would generate motion sequences for a broader range of activities, potentially alleviating the problem of generating irrelevant motion sequences.

5.4.2 Lack of Capturing Subtle Motions. Many daily activities in the HAD-AW dataset, such as 'writing on paper' and 'typing on a keyboard', involve subtle movements at the sensor location (here: the wrist). These subtle movements result in nuanced characteristics within the real IMU data that IMUSim cannot capture through simulation, leading to a distribution difference between the real and virtual IMU data. This discrepancy also contributed to the deep learning classifiers overfitting to the virtual IMU data for some datasets. The classifiers learned features specific only to the virtual IMU data, rendering them not generalizable to the real IMU testing dataset. This problem was also observed in some vision-based cross modality transfer approaches, such as IMUTube [42].

To address this challenge, Santhalingam et al. [68] trained a model to convert 3D hand poses into accurate virtual IMU data for sign language recognition. However, this model is specifically designed for wrist sensor locations and is not applicable to sensors located on other body parts. In our future work, we plan to collect a multi-modal dataset containing both motion capture and sensor data. This will enable us to train a model capable of accurately converting motion sequences to virtual IMU data for all body locations. With more accurate virtual IMU data, the deep learning classifiers can be trained to generalize better to real IMU data.

6 CONCLUSION

This paper presents a significant enhancement to the language-based cross modality transfer paradigm previously introduced and piloted in IMUGPT. We extend the IMUGPT system by introducing the motion filtering module to ensure the relevance of virtual IMU data and establishing metrics to measure the diversity of the generated data, which in turn helps determine how much data to generate. These improvements have been demonstrated to reduce the time and computational demands of data generation by over 50%.

We further consolidate the system with a comprehensive evaluation across various HAR datasets, LLMs, and motion synthesis models, confirming that using GPT-3.5 and T2M-GPT for virtual IMU data generation

leads to the best downstream performance. IMUGPT 2.0 highlights the potential to develop more complex and generalizable HAR systems, thus alleviating a major hurdle in the field: the lack of large labeled datasets.

REFERENCES

- [1] 2021. sentence-transformers/all-mpnet-base-v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2 (2024, Feb 1).
- [2] 2022. GPT-3.5. https://platform.openai.com/docs/models/gpt-3-5 (2024, Feb 1).
- [3] 2023. REDUCELRONPLATEAU. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html (2024, Feb 1).
- [4] Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. 2021. Vid2Doppler: Synthesizing Doppler radar data from videos for training privacy-preserving activity recognition. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–10.
- [5] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, et al. 2023. PaLM 2 Technical Report. arXiv:2305.10403 [cs.CL]
- [6] Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. 2023. SINC: Spatial Composition of 3D Human Motions for Simultaneous Action Generation. In ICCV.
- [7] Matthias Bächlin, Meir Plotnik, Daniel Roggen, Nir Giladi, Jeffrey M Hausdorff, and Gerhard Tröster. 2010. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. IEEE Transactions on Information Technology in Biomedicine 14, 2 (2010), 436–446. https://doi.org/10.1109/TITB.2009.2036165
- [8] Lei Bai, Lina Yao, Xianzhi Wang, Salil S. Kanhere, Bin Guo, and Zhiwen Yu. 2020. Adversarial Multi-view Networks for Activity Recognition. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4, 2, Article 42 (jun 2020), 22 pages. https://doi.org/10.1145/3397323
- [9] Lei Bai, Lina Yao, Xianzhi Wang, Salil S. Kanhere, and Yang Xiao. 2020. Prototype Similarity Learning for Activity Recognition. In Advances in Knowledge Discovery and Data Mining, Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan (Eds.). Springer International Publishing, Cham, 649–661.
- [10] Dmitrijs Balabka. 2019. Semi-supervised learning for human activity recognition using adversarial autoencoders. In Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers (London, United Kingdom) (UbiComp/ISWC '19 Adjunct). Association for Computing Machinery, New York, NY, USA, 685–688. https://doi.org/10.1145/3341162.3344854
- [11] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. OpenFace 2.0: Facial Behavior Analysis Toolkit. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). 59–66. https://doi.org/10.1109/FG.2018.00019
- [12] Sejal Bhalla, Mayank Goel, and Rushil Khurana. 2021. IMU2Doppler: Cross-Modal Domain Adaptation for Doppler-based Activity Recognition Using IMU Data. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5, 4 (2021), 1–20.
- [13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, et al. 2020. Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems, Vol. 33. Curran Associates, Inc., 1877–1901.
- [14] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput. Surv. 46, 3, Article 33 (jan 2014), 33 pages. https://doi.org/10.1145/2499621
- [15] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. 2013. The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042. https://doi.org/10.1016/j.patrec.2012.12.014
- [16] Wenqiang Chen, Shupei Lin, Elizabeth Thompson, and John Stankovic. 2021. SenseCollect: We Need Efficient Ways to Collect On-body Sensor-based Human Activity Data! *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–27.
- [17] Dongzhou Cheng, Lei Zhang, Can Bu, Xing Wang, Hao Wu, and Aiguo Song. 2023. ProtoHAR: Prototype Guided Personalized Federated Learning for Human Activity Recognition. *IEEE Journal of Biomedical and Health Informatics* 27, 8 (2023), 3900–3911. https://doi.org/10.1109/JBHI.2023.3275438
- [18] L. Cilliers. 2020. Wearable devices in healthcare: Privacy and information security issues. *Health information management journal* 49, 2-3 (2020), 150–156.
- [19] Richard O. Duda, Peter E. Hart, and David G. Stork. 2000. Pattern Classification (2nd Edition) (2 ed.). Wiley-Interscience.
- [20] Floyd Els and Liezel Cilliers. 2017. Improving the information security of personal electronic health records to protect a patient's health information. In 2017 Conference on Information Communication Technology and Society (ICTAS). 1–6. https://doi.org/10.1109/ICTAS. 2017.7920658
- [21] Siwei Feng and Marco F. Duarte. 2019. Few-shot learning-based human activity recognition. Expert Systems with Applications 138 (2019), 112782. https://doi.org/10.1016/j.eswa.2019.06.070

- [22] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A Kernel Two-Sample Test. Journal of Machine Learning Research 13, 25 (2012), 723–773. http://jmlr.org/papers/v13/gretton12a.html
- [23] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022. Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5152–5161.
- [24] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. 2022. TM2T: Stochastic and Tokenized Modeling for the Reciprocal Generation of 3D Human Motions and Texts. In *ECCV*.
- [25] N. Y. Hammerla, R. Kirkham, P. Andras, and T. Ploetz. 2013. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *Proceedings of the 2013 international symposium on wearable computers*. 65–68.
- [26] Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L. Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz. 2020. Masked reconstruction based self-supervision for human activity recognition. In *Proceedings of the 2020 ACM International Symposium on Wearable Computers* (Virtual Event, Mexico) (ISWC '20). Association for Computing Machinery, New York, NY, USA, 45–49. https://doi.org/10.1145/3410531.3414306
- [27] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2021. Contrastive Predictive Coding for Human Activity Recognition. 5, 2, Article 65 (jun 2021), 26 pages. https://doi.org/10.1145/3463506
- [28] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2022. Assessing the state of self-supervised human activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–47.
- [29] Thorsten Hempel, Ahmed A. Abdelrahman, and Ayoub Al-Hamadi. 2022. 6d Rotation Representation For Unconstrained Head Pose Estimation. In 2022 IEEE International Conference on Image Processing (ICIP). IEEE. https://doi.org/10.1109/icip46576.2022.9897219
- [30] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. 2023. MotionGPT: Human Motion as a Foreign Language. arXiv preprint arXiv:2306.14795 (2023).
- [31] D. Jiang and G. Shi. 2021. Research on data security and privacy protection of wearable equipment in healthcare. *Journal of Healthcare Engineering* 2021 (2021).
- [32] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 22199–22213. https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf
- [33] Heli Koskimäki, Pekka Siirtola, and Juha Röning. 2017. MyoGym: Introducing an Open Gym Data Set for Activity Recognition Collected Using Myo Armband. In Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3123024.3124400
- [34] Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. arXiv:1808.06226 [cs.CL]
- [35] Hyeokhyen Kwon, Gregory D Abowd, and Thomas Plötz. 2019. Handling annotation uncertainty in human activity recognition. In *Proceedings of the 23rd International Symposium on Wearable Computers*. 109–117.
- [36] Hyeokhyen Kwon, Gregory D Abowd, and Thomas Plötz. 2021. Complex Deep Neural Networks from Large Scale Virtual IMU Data for Effective Human Activity Recognition Using Wearables. Sensors 21, 24 (2021), 8337.
- [37] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. 2020. Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–29.
- [38] Hyeokhyen Kwon, Bingyao Wang, Gregory D Abowd, and Thomas Plötz. 2021. Approaching the Real-World: Supporting Activity Recognition Training with Virtual IMU Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–32.
- [39] Julia Lahn, Heiko Peter, and Peter Braun. 2015. Car Crash Detection on Smartphones (iWOAR '15). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2790044.2790049
- [40] Yi-An Lai, Xuan Zhu, Yi Zhang, and Mona Diab. [n. d.]. Diversity, Density, and Homogeneity: Quantitative Characteristic Metrics for Text Collections. ([n. d.]).
- [41] Clayton Frederick Souza Leite and Yu Xiao. 2020. Improving Cross-Subject Activity Recognition via Adversarial Learning. *IEEE Access* 8 (2020), 90542–90554. https://doi.org/10.1109/ACCESS.2020.2993818
- [42] Zikang Leng, Yash Jain, Hyeokhyen Kwon, and Thomas Ploetz. 2023. On the Utility of Virtual On-body Acceleration Data for Fine-grained Human Activity Recognition. In Proceedings of the 2023 ACM International Symposium on Wearable Computers (ISWC '23). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3594738.3611364
- [43] Zikang Leng, Hyeokhyen Kwon, and Thomas Ploetz. 2023. 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*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3594738.3611361

- [44] Jiyang Li, Lin Huang, Siddharth Shah, Sean J. Jones, Yincheng Jin, Dingran Wang, Adam Russell, Seokmin Choi, Yang Gao, Junsong Yuan, and Zhanpeng Jin. 2023. SignRing: Continuous American Sign Language Recognition Using IMU Rings and Virtual IMU Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 107 (sep 2023), 29 pages. https://doi.org/10.1145/3610881
- [45] Dawei Liang, Guihong Li, Rebecca Adaimi, Radu Marculescu, and Edison Thomaz. 2022. AudioIMU: Enhancing Inertial Sensing-Based Activity Recognition with Acoustic Models. In Proceedings of the 2022 ACM International Symposium on Wearable Computers. 44–48.
- [46] Daniyal Liaqat, Mohamed Abdalla, Pegah Abed-Esfahani, Moshe Gabel, Tatiana Son, Robert Wu, Andrea Gershon, Frank Rudzicz, and Eyal De Lara. 2019. WearBreathing: Real World Respiratory Rate Monitoring Using Smartwatches. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 3, 2, Article 56 (jun 2019), 22 pages. https://doi.org/10.1145/3328927
- [47] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. 2023. Motion-X: A Large-scale 3D Expressive Whole-body Human Motion Dataset. Advances in Neural Information Processing Systems (2023).
- [48] Jiawei Liu, Xiaohu Li, Shanshan Huang, Rui Chao, Zhidong Cao, Shu Wang, Aiguo Wang, and Li Liu. 2023. A review of wearable sensors based fall-related recognition systems. Engineering Applications of Artificial Intelligence 121 (2023), 105993. https://doi.org/10.1016/j. engappai.2023.105993
- [49] Yilin Liu, Fengyang Jiang, and Mahanth Gowda. 2020. Finger Gesture Tracking for Interactive Applications: A Pilot Study with Sign Languages. 4, 3 (2020). https://doi.org/10.1145/3414117
- [50] Yucheng Liu, Naji Khosravan, Yulin Liu, Joseph Stember, Jonathan Shoag, Ulas Bagci, and Sachin Jambawalikar. 2019. Cross-Modality Knowledge Transfer for Prostate Segmentation from CT Scans. In Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data, Qian Wang, Fausto Milletari, Hien V. Nguyen, Shadi Albarqouni, M. Jorge Cardoso, Nicola Rieke, Ziyue Xu, Konstantinos Kamnitsas, Vishal Patel, Badri Roysam, Steve Jiang, Kevin Zhou, Khoa Luu, and Ngan Le (Eds.). Springer International Publishing, Cham, 63–71.
- [51] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. When Video Meets Inertial Sensors: Zero-Shot Domain Adaptation for Finger Motion Analytics with Inertial Sensors. In Proceedings of the International Conference on Internet-of-Things Design and Implementation (Charlottesvle, VA, USA) (IoTDI '21). Association for Computing Machinery, New York, NY, USA, 182–194. https://doi.org/10.1145/ 3450268.3453537
- [52] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. ACM Trans. Graph. 34, 6, Article 248 (nov 2015), 16 pages. https://doi.org/10.1145/2816795.2818013
- [53] MinYen Lu, ChenHao Chen, Shigemi Ishida, Yugo Nakamura, and Yutaka Arakawa. 2022. A study on estimating the accurate head IMU motion from Video. Proceedings of the Symposium on Multimedia, Distributed, Cooperative, and Mobile (DICOMO) 2022 2022 (07 2022), 918–923. https://cir.nii.ac.jp/crid/1050011771467456512
- [54] David Martin, Zikang Leng, Tan Gemicioglu, Jon Womack, Jocelyn Heath, William C Neubauer, Hyeokhyen Kwon, Thomas Ploetz, and Thad Starner. 2023. FingerSpeller: Camera-Free Text Entry Using Smart Rings for American Sign Language Fingerspelling Recognition. In Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3597638.3614491
- [55] Sara Mohammed, Reda Elbasiony, and Walid Gomaa. 2018. An LSTM-based Descriptor for Human Activities Recognition using IMU Sensors. 504–511. https://doi.org/10.5220/0006902405040511
- [56] F. Mohd-Yasin, C.E. Korman, and D.J. Nagel. 2001. Measurement of noise characteristics of MEMS accelerometers. In 2001 International Semiconductor Device Research Symposium. Symposium Proceedings (Cat. No.01EX497). 190–193. https://doi.org/10.1109/ISDRS.2001. 984472
- [57] OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, et al. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]
- [58] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. Sensors (2016).
- [59] Thomas Plötz and Yu Guan. 2018. Deep learning for human activity recognition in mobile computing. Computer 51, 5 (2018), 50–59.
- [60] Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. 2011. Feature learning for activity recognition in ubiquitous computing. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two (Barcelona, Catalonia, Spain) (IJCAI'11). AAAI Press, 1729–1734.
- [61] Thomas Plötz. 2023. If only we had more data!: Sensor-Based Human Activity Recognition in Challenging Scenarios. In 2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). 565–570. https://doi.org/10.1109/PerComWorkshops56833.2023.10150267
- [62] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html
- [63] Daniele Ravi, Charence Wong, Benny Lo, and Guang-Zhong Yang. 2016. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor

- Networks (BSN). 71-76. https://doi.org/10.1109/BSN.2016.7516235
- [64] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. http://arxiv.org/abs/ 1908.10084
- [65] Attila Reiss and Didier Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring (ISWC '12). IEEE Computer Society. https://doi.org/10.1109/ISWC.2012.13
- [66] Vitor Fortes Rey, Peter Hevesi, Onorina Kovalenko, and Paul Lukowicz. 2019. Let There Be IMU Data: Generating Training Data for Wearable, Motion Sensor Based Activity Recognition from Monocular RGB Videos. In Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers. Association for Computing Machinery, 699–708. https://doi.org/10.1145/3341162.3345590
- [67] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task Self-Supervised Learning for Human Activity Detection. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 3, 2, Article 61 (jun 2019), 30 pages. https://doi.org/10.1145/3328932
- [68] Panneer Selvam Santhalingam, Parth Pathak, Huzefa Rangwala, and Jana Kosecka. 2023. Synthetic Smartwatch IMU Data Generation from In-the-Wild ASL Videos. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (2023).
- [69] Allah Bux Sargano, Xiaofeng Wang, Plamen Angelov, and Zulfiqar Habib. 2017. Human action recognition using transfer learning with deep representations. In 2017 International Joint Conference on Neural Networks (IJCNN). 463–469. https://doi.org/10.1109/IJCNN.2017. 7965890
- [70] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in HuggingFace. arXiv:2303.17580 [cs.CL]
- [71] Satya P. Singh, Madan Kumar Sharma, Aimé Lay-Ekuakille, Deepak Gangwar, and Sukrit Gupta. 2021. Deep ConvLSTM With Self-Attention for Human Activity Decoding Using Wearable Sensors. IEEE Sensors Journal 21, 6 (2021), 8575–8582. https://doi.org/10.1109/ JSEN.2020.3045135
- [72] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-body localization of wearable devices: An investigation of position-aware activity recognition. In 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom). 1–9. https://doi.org/10.1109/ PERCOM.2016.7456521
- [73] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo. 2021. SelfHAR: Improving Human Activity Recognition through Self-training with Unlabeled Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (March 2021), 1–30. https://doi.org/10.1145/3448112
- [74] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, and Ioannis Antonoglou others. 2023. Gemini: A Family of Highly Capable Multimodal Models. arXiv:2307.09288 [cs.CL]
- [75] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]
- [76] Lena Uhlenberg and Oliver Amft. 2022. Comparison of Surface Models and Skeletal Models for Inertial Sensor Data Synthesis. In 2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN). 1–5. https://doi.org/10.1109/BSN56160. 2022 9928504
- [77] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural Discrete Representation Learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 6309–6318.
- [78] Vincent T. van Hees, Zhou Fang, Joss Langford, Felix Assah, Anwar Mohammad, Inacio C. M. da Silva, Michael I. Trenell, Tom White, Nicholas J. Wareham, and Søren Brage. 2014. Autocalibration of accelerometer data for free-living physical activity assessment using local gravity and temperature: an evaluation on four continents. Journal of Applied Physiology 117, 7 (2014), 738–744. https://doi.org/10.1152/japplphysiol.00421.2014 arXiv:https://doi.org/10.1152/japplphysiol.00421.2014 PMID: 25103964.
- [79] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. arXiv:2303.04671 [cs.CV]
- [80] Chengshuo Xia, Ayane Saito, and Yuta Sugiura. 2022. Using the virtual data-driven measurement to support the prototyping of hand gesture recognition interface with distance sensor. Sensors and Actuators A: Physical 338 (2022), 113463.
- [81] Chengshuo Xia and Yuta Sugiura. 2022. Virtual IMU Data Augmentation by Spring-Joint Model for Motion Exercises Recognition without Using Real Data. In Proceedings of the 2022 ACM International Symposium on Wearable Computers (ISWC '22). Association for Computing Machinery, 79–83. https://doi.org/10.1145/3544794.3558460
- [82] Fanyi Xiao, Ling Pei, Lei Chu, Danping Zou, Wenxian Yu, Yifan Zhu, and Tao Li. 2021. A Deep Learning Method for Complex Human Activity Recognition Using Virtual Wearable Sensors. In Spatial Data and Intelligence. Springer International Publishing. https://doi.org/10.1007/978-3-030-69873-7_19
- [83] Chenhan Xu, Huining Li, Zhengxiong Li, Xingyu Chen, Aditya Singh Rathore, Hanbin Zhang, Kun Wang, and Wenyao Xu. 2022. The Visual Accelerometer: A High-fidelity Optic-to-Inertial Transformation Framework for Wearable Health Computing. In 2022 IEEE 10th

- $International\ Conference\ on\ Healthcare\ Informatics\ (ICHI).\ IEEE,\ 319-329.$
- [84] Hyungjun Yoon, Hyeongheon Cha, Canh Hoang Nguyen, Taesik Gong, and Sung-Ju Lee. 2022. IMG2IMU: Applying Knowledge from Large-Scale Images to IMU Applications via Contrastive Learning. arXiv preprint arXiv:2209.00945 (2022).
- [85] A. D. Young, M. J. Ling, and D. K. Arvind. 2011. 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. 199–210.
- [86] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. 2023. T2M-GPT: Generating Human Motion from Textual Descriptions with Discrete Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [87] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2022. MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model. arXiv preprint arXiv:2208.15001 (2022).
- [88] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. 2023. ReMoDiffuse: Retrieval-Augmented Motion Diffusion Model. arXiv preprint arXiv:2304.01116 (2023).
- [89] Mi Zhang and Alexander A. Sawchuk. 2012. USC-HAD: A Daily Activity Dataset for Ubiquitous Activity Recognition Using Wearable Sensors. Association for Computing Machinery.
- [90] Shibo Zhang and Nabil Alshurafa. 2020. Deep Generative Cross-Modal on-Body Accelerometer Data Synthesis from Videos. In Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct). Association for Computing Machinery, 223–227.

A USING LLM FOR ACTIVITY FILTERING

In section 3.2.2, we used LLMs to filter out incorrectly generated motion sequences with motion captions as input. We provide the exact prompts that we used along with some example motion captions in Figure 4.

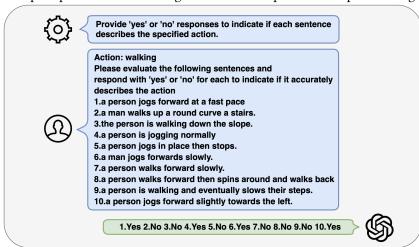


Fig. 4. The prompts passed to the LLM for it to determine whether the given motion captions accurately describe the specified activity.

B DATASETS

Our experimental evaluations are based on five benchmark datasets that are widely used in the HAR community. Table 10 provides an overview.

The RealWorld dataset contains recordings of 15 subjects performing 8 activities, with sensors placed on 7 different locations on the body. Each activity was recorded for 10 minutes each (except for jumping which was recorded for 1.7 minutes), under realistic conditions.

A total of 9 subjects performed 12 activities in the PAMAP2 dataset. Over 10 hours of data were collected from the subjects using 3 IMUs, and a heart rate monitor, sampled at 100Hz. The three IMUs were placed at the chest, the subject's dominant arm, and ankle of the subject's dominant side, while the heart rate monitor was placed at the chest.

For the USC-HAD dataset, 14 subjects recorded 5 trials for 12 activities each on different days at various indoor and outdoor settings. Given the location of the trails and the most common locations where people carry mobile phones, the front right hip was chosen as the location for the sensors.

The HAD-AW dataset was created with 16 subjects wearing the Apple Watch Series One on the right wrist and recording 31 activities, but we are using only 25 activities since the other 6 activities are related to workouts in the gym and will be covered in the MyoGym dataset. The Apple Watch collects data from the subjects who repeat each activity 10 times, amounting to approximately 160 samples of each activity.

The MyoGym dataset is dedicated to 30 activities performed in the gym. 10 subjects recorded data performing a set of 10 repetitions of each activity using a Myo armband worn on the forearm. The exercises were done using free weights and targeted different muscle groups. For our experiments, we have downsampled the frequency to 20 Hz to match that of the virtual IMU data for all datasets.

Table 10. Summary of datasets. The percentage of samples for each activity is represented in the parentheses following the activities.

Dataset	Frequency	Sensor location	Subjects #	Activities #	Activities
RealWorld	50 Hz	Head, Chest, Upper arm, Waist, Forearm, Thigh, Shin	15	8	climbing down (11.3%), climbing up (13.2%), jumping (2.1%), lying (14.4%), running (15.7%), sitting (14.4%), standing (14.2%), walking (14.5%)
PAMAP2	100 Hz	Chest, Wrist, Ankle	9	12	ascending stairs (6.1%), cycling (8.5%), descending stairs (5.4%), ironing (12.3%), lying (10.0%), Nordic walking (9.7%), rope jumping (2.2%), running (5.1%), sitting (9.6%), standing (9.8%), vacuum cleaning (9.1%), walking (12.3%)
USC-HAD	100 Hz	Front Right Hip	14	12	elevator down (5.9%), elevator up (8.5%), jumping (3.8%), running (6.3%), sitting (9.3%), sleeping (10.7%), standing (8.4%), walking forward (13.6%), walking downstairs (7.0%), walking left (9.2%), walking right (9.8%), walking upstairs (7.5%)
HAD-AW	50 Hz	Right wrist	16	25	Bed making (3.8%), Cutting Components (4.2%), Cycling (2.3%), Dancing (3.0%), Drawing (4.2%), Driving Car (6.2%), Eat Sandwich with Hand (3.5%), Flipping (3.8%), Playing on a violin (4.9%), Playing on Guitar (3.6%), Playing on Piano (5.3%), Praying (5.2%), Put off clothes (3.5%), Reading (4.1%), Rowing (3.8%), Running (4.3%), Shaking the dust (4.4%), Showering (3.1%), Sweeping (3.7%), Typing on keyboard (4.1%), Washing dishes (4.2%), Washing hands (3.0%), Wearing Clothes (3.6%), Wiping (3.4%), Writing on paper (4.6%)
MyoGym	50 Hz	Right forearm	10	30	Bar Skullcrusher (3.7%), Bench Dip / Dip (2.8%), Bench Press (2.8%), Bent Over Barbell Row (2.5%), Cable Curl (3.1%), Car Drivers (2.7%), Close-Grip Barbell Bench Press (3.0%), Concentration Curl (3.1%), Dumbbell Alternate Bicep Curl (5.0%), Dumbbell Flyes (4.3%), Front Dumbbell Raise (4.7%), Hammer Curl (4.3%), Incline Dumbbell Flyes (4.2%), Incline Dumbbell Press (3.7%), Incline Hammer Curl (3.6%), Leverage Chest Press (3.1%), Lying Rear Delt Raise (2.9%), One-Arm Dumbbell Row (3.0%), Overhead Triceps Extension (3.1%), Pushups (2.6%), Reverse Grip Bent-Over Row (2.5%), Seated Cable Rows (3.4%), Seated Dumbbell Shoulder Press (3.1%), Side Lateral Raise (3.1%), Spider Curl (3.7%), Tricep Dumbbell Kickback (2.8%), Triceps Pushdown (3.0%), Upright Barbell Row (3.0%), Wide-Grip Front Pulldown (3.5%), Wide-Grip Pulldown Behind The Neck (3.6%)

C REAL AND VIRTUAL DATASET SIZES

In this section, we provide the sizes of the real IMU datasets and virtual IMU datasets that we used in our experimental evaluation (section 4). Table 11 shows the dataset sizes when different LLMs are used for the textual descriptions generation within IMUGPT. Table 12 shows the dataset sizes when different motion synthesis models are used to generate the motion sequences using the textual descriptions generated by GPT-3.5 as input. Table 13 shows the sizes of the virtual IMU datasets with and without using the motion filter.

Table 11. Real and virtual IMU data sizes when different LLMs are used for textual descriptions generation. "Real Data" denotes the baseline experiments not including any generated, virtual IMU data

	RealWorld	PAMAP2	USC-HAD	HAD-AW	MyoGym
GPT-3.5	848 min	1376 min	1400 min	3280 min	2231 min
GPT-4	932 min	1423 min	1397 min	3177 min	2862 min
LLaMa 2	923 min	1435 min	1424 min	3248 min	2549 min
Palm 2	917 min	1455 min	1350 min	3314 min	2392 min
Gemini	881 min	1413 min	1374 min	3200 min	2367 min
Real Data	1107 min	322 min	469 min	662 min	154 min

Table 12. Real and virtual IMU data sizes when different motion synthesis models are used. Motion sequences generated using textual descriptions generated by GPT-3.5. "Real Data" denotes the

	RealWorld	PAMAP2	USC-HAD	HAD-AW	MyoGym
T2M-GPT	848 min	1376 min	1400 min	3280 min	2231 min
MotionGPT	967 min	1560 min	1511 min	3424 min	3234 min
MotionDiffuse	798 min	1187 min	1184 min	2479 min	2975 min
ReMoDiffuse	798 min	1195 min	1195 min	2479 min	2975 min
Real Data	1107 min	322 min	469 min	662 min	154 min

Table 13. Comparison of virtual IMU dataset sizes with and without using motion filter

	RealWorld	PAMAP2	USC-HAD	HAD-AW	MyoGym
Without Motion Filter	848 min	1376 min	1400 min	3280 min	2231 min
With Motion Filter	330 min	517 min	417 min	872 min	425 min

D SAMPLE GENERATED TEXT DESCRIPTIONS

Table 14 and Table 15 gives an overview of some examples of textual descriptions generated through various LLMs. We use the following prompt template to generate textual descriptions of activities:

System Message: You are a prompt generator designed to generate textual description inputs for activities. Do not provide anything other than prompt

User Message: Prompts must be less than 15 words, contain description of the action with no description for environment. The prompt should only describe one person and an adjective describing how the person does the activity. Here are some example inputs: 'a person puts their hands together, leans forwards slightly, then swings the arms from right to left', 'a man is practicing the waltz with a partner', 'a man steps forward and does a handstand', 'a man rises from the ground, walks in a circle', 'a person jogs in place, slowly at first, then increases speed.', 'a man playing violin', 'a man playing guitar', 'a man doing skip rope', 'a man walks happily'." 50 prompts for prompts for "activity".

Table 14. Example generated textual descriptions using various LLMs

	GPT-3.5	The person takes brisk steps, swinging their arms energetically; Walking on a tightrope, the person carefully balances their body in mid-air; A scientist paces back and forth in their lab, deep in thought, pondering a breakthrough; Someone leisurely strolls down the tree-lined path, enjoying the serene surroundings; A detective walks stealthily, investigating a crime scene for clues				
walking GPT-4		A woman strolls, pausing to admire the flowers; A jogger cools down, slowing to an easy walk; A bride walks down the aisle, eyes locked on her partner; She power-walks on the sunlit track field; A janitor ambles around the school after hours				
	LLaMa 2	A woman strolls casually, her hands in her pockets, enjoying the sunny day; A woman power walks, her strides long and confident; A man carries a heavy load, his steps slow and labored; A child skips along the sidewalk, laughing with joy; A person ambles through a museum, taking in the artwork				
	Palm 2 (Bard)	A person walks to the store; A child skips down the sidewalk; A person walks in a park, enjoying the grass; A person walks alone; A dog walks alongside its owner, wagging its tail				
	Gemini	A person walks slowly with a determined stride; A soldier walks in formation, serving their country; A relaxed man strolls in the park; A child skip happily, a smile on their face; A child skips down the sidewalk				
GPT-3.5		A person crouches down, then springs upward with both feet leaving the ground; A child jumps on a trampoline, bouncing up and down with sheer delight; Leaping over a stream, a hiker successfully clears the water, continuing their trek through the forest; The person jumps with precision and accuracy, landing softly on the ground; Performing a series of jumps, the person maintains a steady rhythm and flawless technique				
	GPT-4	A woman crouches low, then springs up into a high jump; A soccer player jumps up to head a high ball; A frog enthusiast mimics a frog, leaping around a pond; She jumps in place, warming up before her run; Launching off one foot, he soars with ease				
	LLaMa 2	A man jumps onto a fitness class, his legs spread wide apart; A person jumps over a puddle, avoiding a splashy landing; A woman jumps rope, her arms swinging in perfect rhythm; A woman jumps over a hurdle, her legs long and lean; A child jumps off a swing, their arms spread wide in excitement				
	Palm 2 (Bard)	A person jumps up and down in excitement; A person jumps to be alive; A person jumps to celebrate; A person jumps to get exercise; A person jumps to be confident				
	Gemini	A girl with arms up and legs straight leaps skywards; A girl jumps up and down to a catchy song; A boy jumps to catch a falling object; A girl jumps to show off; A cat jumps out of a window, thirsty for a taste of freedom				
cycling	GPT-3.5	A person mounts the bicycle, grips the handlebars, and pushes off with one foot; A cyclist stands up on the pedals, exerting power to gain speed; A cyclist maintains a steady pace, finding solace in the rhythmic motion; Balancing on one wheel, the cyclist effortlessly rides a unicycle; A cyclist accelerates, increasing their speed with each push of the pedal				
	GPT-4	A cyclist weaves expertly through city traffic; The biker uses hand signals when approaching a turn; A racer crouches low, speeding on the track; On a country road, the biker enjoys a solo ride at dawn; He shifts gears, cycling up the steep incline				
	LLaMa 2	A woman pedals up a steep hill, her face determined; A person races against time, speeding through an empty park on their bike; A man in a cycling competition takes a sharp corner, his tires screeching; A person rides a bicycle through a scenic park, enjoying the views; A person cycles through a bustling park, dodging pedestrians				
	Palm 2 (Bard)	A cyclist rides a bike down a winding road; A cyclist rides a bike to improve their fitness; A cyclist rides his bike in the heat; A man rides a bicycle down a busy street; A cyclist rides for freedom				
	Gemini	Man balances and propels forward on a bicycle with confidence; A cyclist pedals vigorously up a steep hill; An elderly man cycles slowly on a rural path; A man in a suit cycles briskly to work; A young boy cycles through the park, wind in his hair				
pushups	GPT-3.5	A person begins in a plank position, lowers their body with controlled movement; A person lowers their body towards the ground, then pushes themselves back up; A person's biceps contract as they lower their chest towards the floor; A person maintains a steady pace, counting each pushup as they go; On a mat, a woman extends and bends her arms, repeatedly				
	GPT-4	Her triceps work hard as she raises up; Arms quaking, he pushes through the burn; He lowers his chest to just above the floor; A hiker does pushups beside the trail; A senior does modified pushups against a wall, building strength safely				
	LLaMa 2	A person starts in a plank position, lowering their body until their chest nearly touches the ground; A man uses a kettlebell to add weight to his pushups; A man in a spin class does pushups, his legs still pedaling as he works his upper body; A man in his 20s does pushups during a break at school or work; A man grunts with effort as he completes a set of pushups				
	Palm 2 (Bard)	A person lowers their body until their chest touches the ground, then pushes back up to the starting position; A person does a pushup with their feet on a chair; A person does a pushup with their feet elevated; A person does a pushup, then stands up; A person does a push-up with their legs bent				
	Gemini	A young girl smoothly transitions from plank to pushup; A woman with a fit physique does clapping pushups with ease; A group of friends do pushups together as a challenge; A woman does pushups on a bench; A boy with short hair grunts as he does pushups				

 $Table\ 15.\ Example\ generated\ textual\ descriptions\ with\ context\ information\ using\ GPT-3.5\ for\ `walking'$

No parameters	The person takes brisk steps, swinging their arms energetically; Walking on a tightrope, the person carefully balances their body in mid-air; A scientist paces back and forth in their lab, deep in thought, pondering a breakthrough; Someone leisurely strolls down the tree-lined path, enjoying the serene surroundings; A detective walks stealthily, investigating a crime scene for clues
Age	A teenager walks with confidence, swinging their arms rhythmically; An elderly woman uses a walking cane to steady herself as she slowly moves forward; A middle-aged woman walks briskly, multitasking by talking on her phone and checking her watch; A baby takes independent steps, their face full of determination and triumph; A school-aged child skips and hops, finding joy in every step they take
Weather	A person walks with an umbrella, shielding themselves from pouring rain; A jogger perseveres through blistering heat, their body glistening with sweat; A woman walks leisurely on a cool autumn evening as leaves softly carpet her path; A man walks steadily through a gusty wind, his jacket billowing behind him; In a snowdrift, a person walks hesitantly, lifting their legs with each step
Physique	A slender woman strides gracefully, her limbs moving in perfect synchronization; A sturdy hiker navigates rough terrain, carefully placing each step to maintain balance; A petite gymnast tumbles and flips while walking, turning the pathway into a personal gym mat; A tall woman walks gracefully, elongating her strides with elegance; A muscular man power-walks, propelling himself forward with each determined step
Age & Weather	A young child takes small steps, holding their parent's hand while walking in the rain; A woman in her forties walks briskly on a treadmill at the gym, headphones in; A person walks carefully on an icy sidewalk, taking small steps to avoid slipping; n elderly woman walks slowly with a cane, determined to stay active despite the rain; A person walks on a sandy beach, feeling the warmth of the sun on their skin
Age & Physique	A fit and athletic teenager strides confidently, covering a long distance in each step; A young athlete jogs lightly in between walking, maintaining an energetic and athletic stride; An elderly woman holds onto a walking stick, taking slow and measured steps; An overweight man waddles slowly, his gait slightly unstable, breathing heavily; A person with a prosthetic limb walks determinedly, their movements adapted but resilient
Weather & Physique	A slender woman effortlessly glides through the rain, walking with graceful determination; A person determinedly walks through a muddy field, pushing through the resistance; A hunched elderly man shuffles slowly with the aid of a cane during a snowstorm; A lean man power walks through a snowstorm, his breath visible in the freezing air; A petite man walks briskly, his coat flapping in the strong breeze
Age, Weather & Physique	An elderly woman strolls slowly, using a cane, under a clear blue sky; A petite girl skips merrily, twirling her colorful umbrella on a rainy day; A slim and athletic woman hikes uphill, sweat glistening on her forehead, on a sunny day; A pregnant woman takes gentle walks around the neighborhood, supporting her growing belly; A middle-aged man strolls leisurely along the beach, enjoying the warm breeze

E SAMPLE GENERATED MOTION SEQUENCES

Activity: climb down stairs
Textual description: A person descends stairs, one step at a



Activity: running Textual description: A man runs for exercise



Activity: lying Textual description: A young boy lies on his bed, daydreaming.

Activity: sitting
Textual description: A person sits comfortably while reading a book.



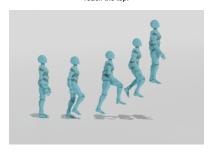
Activity: standing Textual description: A person balances on one leg while standing.



Activity: climb up stairs
Textual description: A person sprints up the stairs, eager to reach the top.



Activity: walking Textual description: A boy walks clumsily with crutches.



Activity: jumping Textual description: A man jumps over a small gap in the road, safely landing.

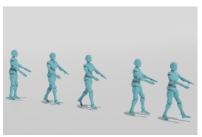




Fig. 5. Example visualized motion sequences for activities in the RealWorld dataset.