Synthetic Smartwatch IMU Data Generation from In-the-wild ASL Videos

PANNEER SELVAM SANTHALINGAM, Computer Science Department, George Mason University
PARTH PATHAK, Computer Science Department, George Mason University
HUZEFA RANGWALA, Computer Science Department, George Mason University
JANA KOSECKA, Computer Science Department, George Mason University

The scarcity of training data available for IMUs in wearables poses a serious challenge for IMU-based American Sign Language (ASL) recognition. In this paper, we ask the following question: can we "translate" the large number of publicly available, in-the-wild ASL videos to their corresponding IMU data? We answer this question by presenting a video to IMU translation framework (Vi2IMU) that takes as input user videos and estimates the IMU acceleration and gyro from the perspective of user's wrist. Vi2IMU consists of two modules, a wrist orientation estimation module that accounts for wrist rotations by carefully incorporating hand joint positions, and an acceleration and gyro prediction module, that leverages the orientation for transformation while capturing the contributions of hand movements and shape to produce realistic wrist acceleration and gyro data. We evaluate Vi2IMU by translating publicly available ASL videos to their corresponding wrist IMU data and train a gesture recognition model purely using the translated data. Our results show that the model using translated data performs reasonably well compared to the same model trained using measured IMU data.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing**; **Accessibility technologies**; • **Computing methodologies** → **Artificial intelligence**.

Additional Key Words and Phrases: Sign language recognition, Data collection, IMU, Wearables, Accessible computing

ACM Reference Format:
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.* 7, 2, Article 74 (June 2023), 34 pages. https://doi.org/10.1145/3596261

1 INTRODUCTION

With the increasing popularity of smartwatches and fitness trackers, developing IMU-based American Sign Language (ASL) recognition solutions will be highly valuable to the deaf and hard-of-hearing (DHH) community. Such solutions can enable human-computer interaction applications in home assistant systems (such as interacting with smart speakers), ASL-to-text transcription in video conferencing, etc. In addition, they can also bridge the communication gap between ASL speakers and non-ASL speakers. While the potential remains significant, the amount of research and progress in wearable IMU based ASL recognition appears relatively limited [1, 2] compared to camera-based solutions [3–10] which are extensively studied. Camera-based solutions incur privacy concerns as they need continuous monitoring and also suffer in performance in poor lighting conditions. Because of their continuous monitoring, camera-based solutions can also capture information about other humans who are not

Authors' addresses: Panneer Selvam Santhalingam, psanthal@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030; Parth Pathak, phpathak@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030; Huzefa Rangwala, rangwala@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030; Jana Kosecka, kosecka@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030.

2474-9567/2023/6-ART74
https://doi.org/10.1145/3596261

Table 1. Comparison of Vi2IMU with state-of-the-art.

System	Task	Method	Video diversity	3 axis gyro	3 axis accelerometer	No. of classes
IMUTube [11, 12]	Human activity recognition	Trajectory based	High	Yes	Yes	13
CNN based framework [13]				No	Yes	12
Let there be IMU [14]		Generative	Low	No	Yes	10
cVGAN [15]					Yes	N/A
Deep Inertial Poser [16]	Pose reconstruction	Trajectory based	Low	No	Yes	N/A
zeroNet [11]	ASL recognition	Trajectory based	Low	No	Yes	50
Vi2IMU			Medium - High	Yes	Yes	70

the intended users, which raises further privacy concerns. In contrast, IMU-based solutions are portable, low-cost, on-body, and do not require continuous camera-like monitoring. An important factor hindering research and the adoption of IMU-based solutions is the lack of large public datasets. Camera-based solutions take advantage of in-the-wild videos on streaming platforms like YouTube and scale the ASL recognition capability to orders of magnitude higher than existing IMU-based solutions. Collecting and labeling a large amount of IMU data require significant human efforts, slowing down the research and development. In this paper, we ask the following question: Can we "translate" the large amount of publicly available, in-the-wild ASL videos to their corresponding IMU representation? If yes, the translated IMU data can then be directly used for research and development of IMU-based ASL recognition solutions.

This paper attempts to address the ASL data availability problem using synthetic IMU data generation through *modality translation*. We present a novel framework (referred to as Vi2IMU) that takes as input publicly available ASL videos and estimates the IMU acceleration and rotation from the perspective of the user's wrist joint. The predicted IMU data can be thought of as the acceleration and gyro data that would have been observed by the IMU on the user's smartwatch, fitness tracker, or any other wrist-worn wearable device. Vi2IMU is developed on our insight that carefully tracking different hand and arm joints in videos can enable us to estimate displacement and orientation of the wrist. These combined with necessary transformations can be used to derive a model that can estimate wrist-based IMU's acceleration and gyro. The translated acceleration and gyro data obtained using Vi2IMU can then be directly used for research and development.

Understanding its importance, some recent works [11, 14, 17, 18] have attempted to solve the problem of video to IMU translation. Generative methods proposed in [14, 18] apply machine learning to learn a function that can derive IMU data from videos. On the other hand, trajectory-based methods [11–13, 16, 17, 19] first determine 3D joint positions from videos and then use forward kinematics to estimate joint orientations. The obtained orientation values are used to transform the 3D joint positions to IMU's frame-of-reference. The second order derivative of the transformed joint positions is then used for computing acceleration, while the first order derivative is used for computing gyro (angular velocity).

While the existing works have made important contributions, there are various limitations discussed below that render them unsuitable for translating data for wrist-worn IMUs which are key to ASL recognition. Table 1 provides a detailed comparison.

(1) A majority of existing work [12–15, 19] focuses on human activity recognition tasks. In contrast to ASL gestures, human activities involve the movement of the entire body's joints. This means that synthetic IMU data is created for multiple body joints where an error in IMU estimation for one joint can be compensated

by another IMU at a different body joint. In comparison, estimating the data for a single wrist-worn IMU requires *fine-grained prediction of orientation and hand movements*.

(2) Utilizing forward kinematics (as proposed in [12, 16, 17] for body joints) for wrist orientation estimation does not account for wrist rotations when resultant orientation change is not due to wrist displacement. While ignoring wrist rotations might not lead to significant errors in human activity recognition, such errors can lead to very poor performance in ASL recognition. For example, the major difference between two ASL signs `Mom`[20] and `Color`[21] is primarily the orientation of the wrist while the remaining arm joint movements are similar.

(3) Existing methods are not designed to account for acceleration changes that are observed at the wrist due to hand and finger movements. They are not capable of capturing the subtle accelerations that are observed at wrist when user performs hand/finger movements (e.g., open and close fist or change hand shape).

(4) Lastly, in-the-wild ASL videos can exhibit a large amount of diversity in terms of users, video backgrounds, lighting conditions, etc. Existing work [11] performs video to IMU translation for a finger-worn IMU, however, the approach was demonstrated on videos collected in controlled settings [22] with very little diversity.

As shown in Table 1, currently there exist no solution that can generate 3-axis acceleration and 3-axis gyro data for a wrist-worn IMU from hundreds of in-the-wild ASL videos.

We present Vi2IMU, a video to wrist IMU translation technique that enables us to create synthetic IMU acceleration data for wrist wearables directly from a diverse set of videos with the ability to scale to a large number of ASL gestures. Vi2IMU is built as a modular framework with two important modules of (i) orientation estimation and (ii) acceleration and gyro estimation. We use existing well-studied solutions to extract hand and arm joint positions from videos. Our orientation estimation module uses them to estimate the wrist orientation with respect to the camera's frame-of-reference. The orientation along with the displacement and positions of the hand and arm joints are then used for acceleration and gyro estimation. While the acceleration can be directly calculated by taking second order derivative of wrist displacement, such an approach can be very inaccurate as it ignores the wrist rotations and simply considers translational movements. Our key insight is that carefully accounting for hand joints in orientation and acceleration estimation enables us to capture wrist rotations as well as fine-grained hand/finger motion related accelerations observed at the wrist. We now present an overview of challenges and our solutions for both modules.

(1) Wrist orientation estimation. We demonstrate that incorporating hand joint positions along with wrist and arm joints can enable us to accurately estimate wrist rotations. This significantly improves wrist orientation estimation which would otherwise simply account for only arm movements resulting from gestures. We find that leveraging hand joint position information is not straightforward because in-the-wild ASL videos suffer from a range of issues including motion blur, poor lighting, etc. that result in inaccurate hand joint position estimation. We address this challenge by proposing a frame grouping strategy that groups together frames with common hand shape while ensuring that each group has one or more frames without motion blur. We then design a bi-directional LSTM based model that takes the frame groups and predicts the wrist orientation for each frame even when it contains motion blur by leveraging hand shape and joint information from other frames without blur in the group. This results in a continuous and highly accurate orientation estimation that can then be leveraged for acceleration and gyro predictions.

(2) Acceleration and gyro prediction. Through investigation of a large number of in-the-wild ASL videos, we find that simply using the displacement of the wrist joint (after transforming using orientation) is far from sufficient to accurately predict acceleration. This is because the movements of hand joints add a non-trivial amount of acceleration to the acceleration observed at the wrist. For example, when a user opens and closes her fist while keeping the wrist stationary, the movement of wrist muscles still results in acceleration. We also

find that even simply changing the hand shape while performing the same wrist and arm movement results in different acceleration observed at the wrist. These challenges make the direct computation of acceleration from wrist displacement highly inaccurate. Instead, we propose an LSTM-based multitask deep learning model that tries to learn the temporal dependencies between the movements of arm, wrist, and hand joints and the resultant acceleration. In doing so, the model pays attention to hand joint information when wrist displacement values are not significant and considerable rotational movements are perceived through gyro estimates. The two tasks then output the predicted acceleration and gyro for the wrist-worn IMU.

Vi2IMU evaluation. We extensively evaluate both modules of Vi2IMU individually as well as the entire end-to-end video to IMU translation.

For training and evaluating individual modules (orientation and acceleration/gyro), we collect our own dataset with 5 subjects and 1.6M frames. Our evaluation shows that (1) our orientation module achieves an orientation error (rotation angle between predicted and ground truth rotation matrices) of 12.49°, (2) the acceleration prediction achieves an average mean absolute error of 0.54 m/s^2 for three axes, (3) the gyro prediction achieves a average mean absolute error of f 0.39 $radian/s$ for three axes, and (4) compared to manual computation from displacement, Vi2IMU's models achieve on average 20.86% improvement in acceleration prediction, significantly advancing the state-of-the-art through careful accounting of impacts from hand joints and shape.

Furthermore, we train an IMU ASL gesture recognition model purely from IMU wrist acceleration and gyro data translated from videos without any collected IMU training data. We use a publicly available, in-the-wild ASL video dataset (MS-ASL [7]) with a large diversity in terms of users, video background, lighting conditions, etc., and calculate the wrist acceleration/gyro from videos. The videos in MS-ASL were curated from YouTube and labeled manually. We evaluate the accuracy of the recognition model using test IMU samples and find that our framework can accurately produce wrist IMU acceleration/gyro data that can be used to avoid laborious data collection efforts. Furthermore, we show that augmenting the translated data by incorporating gesture-specific and subject-specific attributes significantly improves performance. For 50 gestures, the model trained using real, measured data achieves a Top-1 accuracy of 91.6% while the model trained using our translated data achieves a Top-1 accuracy of 84.1%. With gesture-specific and subject-specific data augmentation, for 50 gestures, the Top-1 accuracies are 90.7% and 93.26% respectively. A model trained using real, measured IMU data for 70 gestures achieves Top-1, Top-3, and Top-5 accuracies of 86%, 100%, and 100%, while the same model trained using our translated IMU data achieves 66.6%, 86.2%, and 93.4% accuracies. For 70 gestures, with gesture-specific augmentation the Top-1 accuracy is 77.3%. This shows that Vi2IMU can accurately produce synthetic IMU acceleration and gyro data from in-the-wild videos with diverse conditions. We have made our translated IMU data (plots and raw acceleration and gyro data) from MSASL videos for 70 gestures along with corresponding measured IMU data available anonymously. Please refer to Appendix A.

We summarize our contributions as follows:

(1) We address the challenges in wrist orientation estimation by proposing a frame grouping strategy that is complemented by a bi-directional LSTM-based deep learning architecture that estimates wrist orientation while accounting for the inaccuracies in hand joint estimations.
(2) We demonstrate the importance of hand joints and the impact of hand shape in wrist acceleration estimation. We propose an LSTM-based multitask deep learning model that jointly estimates acceleration and gyro by learning common feature representations.
(3) We extensively evaluate different modules using a large amount of video and IMU data. We also translate a publicly available in-the-wild ASL video dataset to corresponding IMU data and evaluate the accuracy of gesture recognition using a model trained purely using the translated IMU data. Results suggest that our proposed translation framework is practically viable and useful in different applications.

2 SYSTEM OVERVIEW

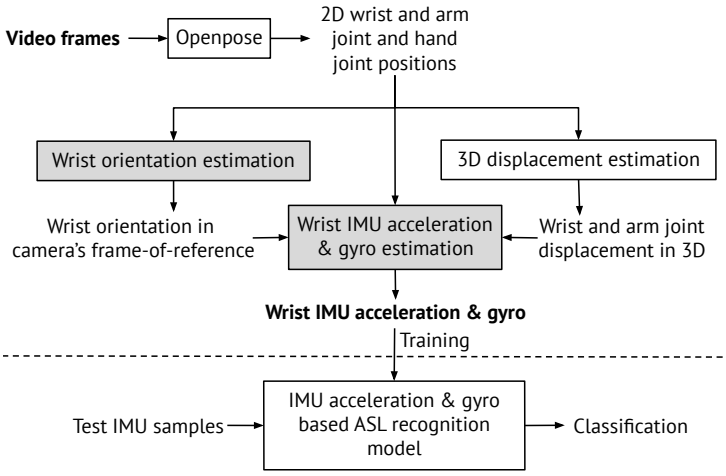


Fig. 1. Vi2IMU overview.

Fig. 1 shows the overview of Vi2IMU. First, we use a state-of-the-art 2D pose estimation model - OpenPose [23] - to extract the 2D hand (21 joints) and arm joints (3 joints) from the video. OpenPose uses part affinity fields to estimate the joint positions on an image and has been shown to achieve very good performance on several benchmark datasets [24, 25]. Next, the 3D displacement estimation module takes the extracted 2D arm joints and estimates the corresponding 3D displacement with respect to the camera. We use an existing 2D to 3D regression model [26] which utilizes a residual architecture for 3D displacement estimation.

The 2D hand and arm joint positions are also input to the wrist orientation estimation module which estimates the wrist orientation with respect to the camera's frame-of-reference. The wrist orientation estimation module utilizes a bi-directional LSTM based architecture that enables orientation estimation even for frames with missing hand joints by leveraging the hand shape information from nearby frames with relatively more accurate hand joint predictions. The 3D displacement and wrist orientation along with the 2D hand joint positions become input to the wrist IMU acceleration & gyro estimation module that predicts the wrist IMU acceleration and gyro (angular velocity). The wrist IMU acceleration & gyro estimation module utilizes an LSTM based multi-task deep learning model that takes advantage of the correlation between acceleration and gyro to learn relevant feature representations and scale for diverse gesture classes and their videos. The translated IMU acceleration and gyro can be directly used for training IMU-specific ASL recognition models.

3 WRIST ORIENTATION ESTIMATION

While it should be possible to compute acceleration as a function of the obtained 3D displacements, we still cannot do so as the displacement values are in the camera's frame-of-reference, while the IMUs measure acceleration in their local frame-of-reference. To address this, we need to transform the displacement values to IMU's frame-of-reference before computing acceleration. Such a transformation requires estimating the orientation of wrist-worn IMU with respect to the camera as shown in Fig. 2a. Here, wrist orientation is nothing but the rotations required to align the camera's frame to that of IMU's. Additionally, the estimated orientation can also be used to calculate angular velocity (i.e., gyro).

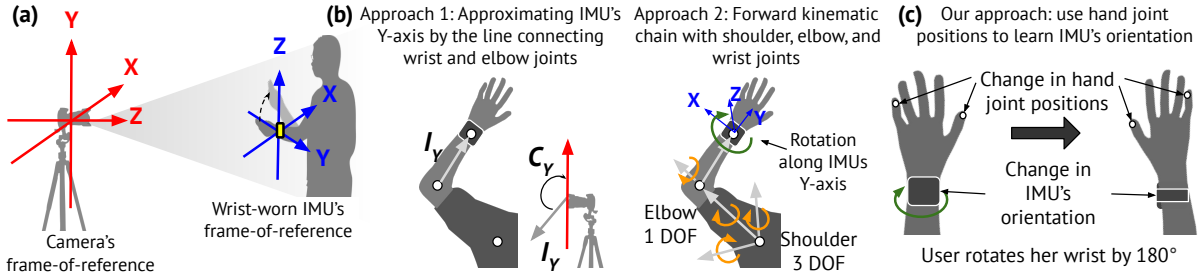


Fig. 2. (a) Transformation of displacement, (b) existing approaches and (c) our approach for orientation estimation

3.1 Accounting for Wrist Rotations

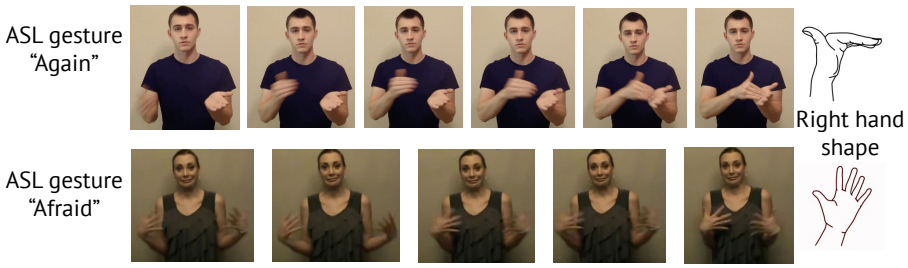
3.1.1 Limitations of existing solutions. We next consider two existing approaches that cannot be directly used for wrist orientation estimation due to various limitations.

The first approach (similar to the one presented in [11] for finger orientation estimation) is to approximate one of the IMU's axes using the limb connecting two arm joints. We can approximate the Y-axis of the IMU (I_y) using the line joining the wrist and elbow joints as shown in Fig. 2b (Approach-1). Since we have the joint positions in camera's perspective, the angle between the line (I_y) and the camera's Y-axis (C_y) can be treated as the one axis orientation. The main limitation of this approach is that this only provides us with one orientation value, and we cannot use this transform from the camera's frame-of-reference to that of IMU's.

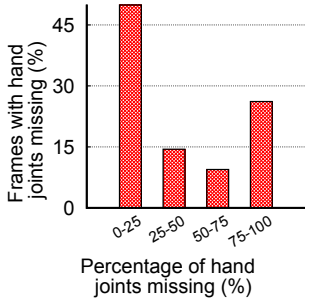
Another approach would be is to create a kinematic chain comprising the shoulder, elbow, and wrist joints, and utilize forward kinematics to estimate wrist orientation [17]. This is shown in Fig. 2b (Approach-2). Here, the orientation of the wrist is estimated as a function of change in position of the different joints in the kinematic chain. For example, when the user moves her arm up towards his head, there is rotational movement in the elbow which can be detected by the change in displacement values along the camera's Y-axis.

Both the above-mentioned approaches can work for wrist-orientation estimation when the considered gestures primarily involve arm joint movements. However, when a gesture only involves hand joints movement with wrist orientation change, the methods perform very poorly. For example, when a user rotates her wrist (as shown in Fig. 2b using a blue arrow) along the IMU's Y-axis, there is no considerable displacement of the arm joints, indicating no orientation change. As the forward kinematic chain estimates orientation as a function of change in positions of the joints in the chain, the corresponding orientation change will not be registered. Such gestures are common in ASL. In fact, wrist rotation is considered to be an important phonological property (a unique underlying characteristic that is common across many gestures) as per the ASL-LEX database [27]. Out of 2000 signs included in the database, approximately 20% of them are categorized under gestures with wrist rotations.

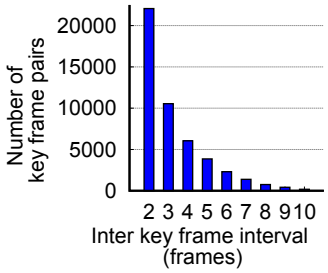
3.1.2 Understanding the role of hand joints. We claim that hand joints play an important role in determining wrist orientation especially in the presence of wrist rotations. As shown in Fig. 2c, as the user rotates her wrist, one important observable change in the body is the change in hand joint positions. This means that it should be possible to model the orientation change corresponding to wrist rotations (along the IMU's Y-axis) just as a function of hand joint positions. However, a critical problem in doing so is that publicly available gesture videos often have poor lighting and high amount of motion blur. This makes it extremely difficult to recover hand joint positions on a large number of frames. Fig. 3a shows a subset of frames for two ASL gestures. As it can be clearly seen, the hand joint positions are hard to identify in some of the frames due to poor lighting and motion blur. This results in low confidence hand joint estimates which we refer to as missing hand joints.



(a) Impact of motion blur on hand joint estimates



(b) Missing handjoints in frames



(c) Inter key frame intervals

Fig. 3. Motion blur and hand joint estimation.

3.1.3 Tackling missing hand joints. In addressing the issue of missing hand joints due to motion blur, we make two observations: (1) We note that although the hand joint positions vary in subsequent frames, the hand shape remains consistent over at least a sequence of frames. We analyze over 895 videos of 80 gesture classes in the MS-ASL dataset [7] and find that although arm position can change, the hand shape remains consistent typically for the duration of 200 to 400 ms duration, the hand shape can change due to minor variations, transitions to a different hand shape (for gestures with multiple hand shapes such as [28]) or at the start or end of the gesture. (2) We also observe that frames with missing hand joints vary in terms of the number of joints that are missing and they are either preceded or followed by frames that have less motion blur and more hand joints visible. For example, in the first row of Fig. 3a, the hand joints are not completely visible in the first two frames. However, the visibility improves in subsequent frames and all the joints are clearly visible in the last frame. To further validate this, we estimate the number of hand joints missing over different frames for 100 gestures (1.4 M frames) in MSASL. Fig. 3b shows the number of frames in the dataset with different percentages of hand joints missing. We find that for 64.32% of the frames, the amount of hand joints missing is less than 50%. These are frames with less motion blur and we will refer to them as key frames. Fig. 3c shows the distribution of the number of frames with more than 50% hand joints missing between two key frames i.e., the inter key frame interval. The inter key frame interval is computed over the frames for 895 gesture videos. We find that the average inter-key frame interval is 3.24 frames. This means that in a group of frames, a frame with zero hand joints will be close to a key frame either before or after it in the group.

3.2 Orientation with Key and Delta Frames

We leverage the observations we made above as follows. First, we categorize frames based on the number of missing hand joints. Along with the 2D joint positions, the OpenPose model also provides a confidence score

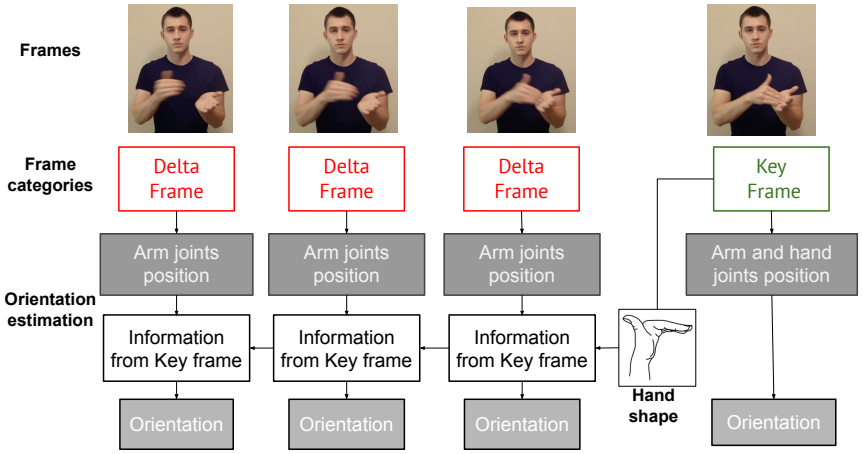


Fig. 4. Incorporating information from key frames into the delta frames for orientation estimation.

for different joints. Let E_f^j be the position estimate for joint j in frame f. Then the confidence score (P_f^j) is the probability that the estimate for the joint j in frame f is correct. Typically, frames with poor lighting or motion blur yield lower confidence scores for joint estimates. We use the confidence score to determine the validity of the obtained joint positions. Any joint with $P_f^j < \eta$ (a predefined threshold value) is considered invalid and treated as a missing joint. Frames that have more than ϵ hand joints missing are categorized as "delta" frames and the remaining are categorized as "key" frames. Fig. 4 shows the categorization of frames into key and delta for four frames. Our insight here is that we can export the knowledge in terms of the hand shape obtained from key frames to estimate the orientation for delta frames as shown in Fig. 4. However, incorporating the knowledge from key frames and combining it with the existing arm joint information in estimating the orientation for delta frames is nontrivial. We solve this challenge by designing a model that facilitates the transfer of knowledge between key and delta frames and estimates orientation as a function of the combined knowledge.

Fig. 5 shows our wrist orientation estimation model. The model takes as input a group of frames at a time where each frame consists of 2D arm and hand joint positions corresponding to it. We divide all frames of the video such as that there is at least one key frame in each group. This will enable the delta frames to learn the hand shape information from the key frames within that group as the model runs. We utilize a bi-directional LSTM architecture where two LSTM cells are used for modeling the input. We choose the bi-directional modelling as a key frame can either precede or follow a delta frame. We empirically choose ϵ and η to 50% and 25% respectively for categorizing the frames.

Before the inputs are passed to the LSTM cells, they are projected to higher dimensions through a linear block. The linear block is comprised of a linear layer followed by batch normalization for normalizing the input, dropout, and Rectified Linear Units (ReLU) for activation. ReLU activation function is used for avoiding the vanishing gradient problem [29] and batch normalization is used for stabilizing the input allowing for faster convergence. We utilize 2 layers of LSTM cells both for forward and backward LSTM with a dropout layer in between. The output of the forward and backward LSTM cells is concatenated and passed to another linear block which is comprised of two linear layers with dropout, batch normalization, and ReLU for activation. For the linear layers and LSTM cells, we use 1024 hidden units and set the dropout value to 0.5. We use Means Squared Error (MSE) as the loss function.

3.2.1 Ground truth for wrist orientation in camera's frame of reference. We need the ground truth wrist IMU orientation with respect to the camera's frame-of-reference in practice to train the supervised ML model. We

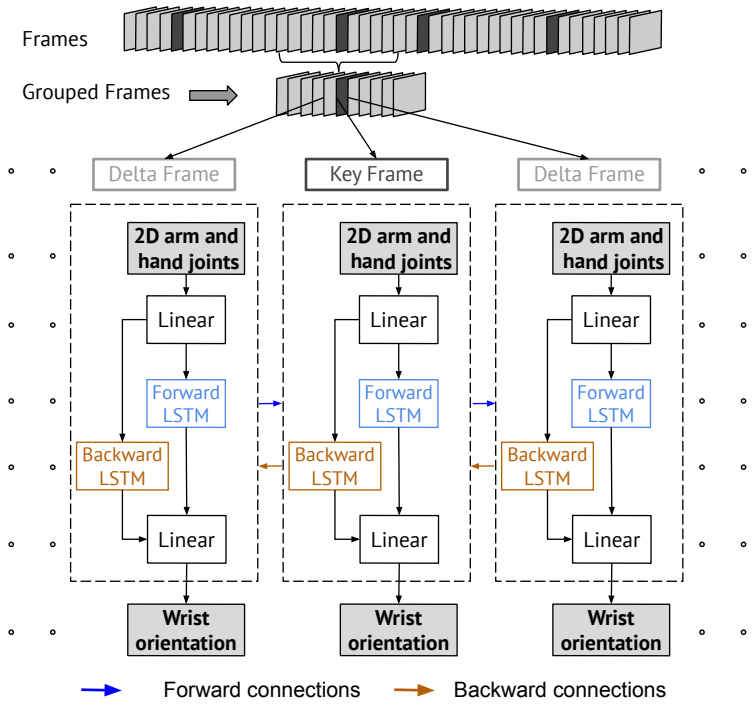


Fig. 5. Our orientation estimation model.

obtain this ground truth as follows: we first measure the camera's orientation with the respect to the earth's frame-of-reference (let's call it O_E^C) by aligning the frame of an (additional) IMU with the camera. We also measure the orientation of the wrist-worn IMU in the earth's frame-of-reference (referred as O_E^W). The rotation matrix R can be calculated between O_E^C and O_E^W. Furthermore, let O_C^W be the orientation of wrist IMU in camera's frame-of-reference. We are interested in calculating O_C^W from O_E^C and O_E^W, both of which are measured. The inverse of R can be used to calculate the O_C^W as $O_C^W = R^{-1}O_E^W$. *We note that this ground truth calculation requires the camera's orientation in the earth's frame-of-reference. However, this is only needed during the training of the orientation estimation model. When we use the model for orientation estimation for the in-the-wild videos, we do not need the camera's orientation. Our scheme assumes that the camera's orientation is not available while translating videos to IMU data.*

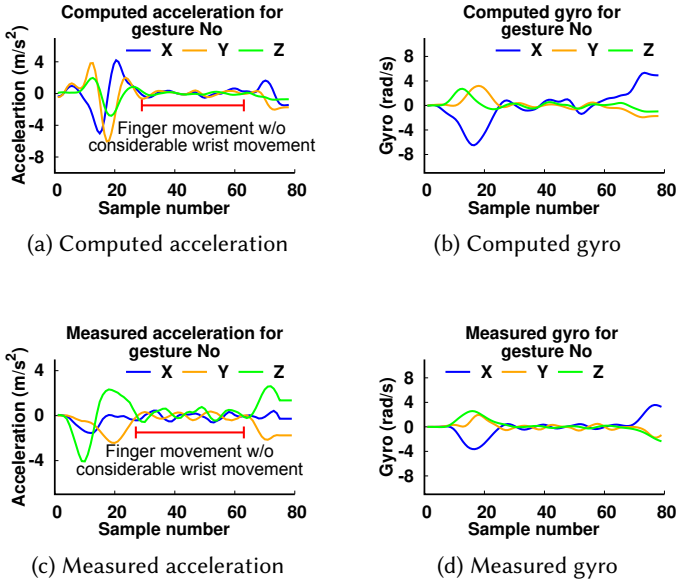
4 WRIST ACCELERATION & GYRO ESTIMATION

4.1 Impact of Hand Shape and Movements

After the calculation of displacement that accounts for IMU's orientation (i.e., in IMU's frame-of-reference), we can calculate the acceleration values by taking the second order derivative of the subsequent displacement samples. Similarly, we can take the first order derivative of the subsequent orientation values and obtain the angular velocity (gyro). For both first and second order derivatives, we can use finite differences following the prior works [11, 30, 31]. Fig. 6 compares the computed acceleration and gyro values with measured IMU acceleration and gyro for the gesture No [32]. We observe that while there are visible similarities in pattern between the computed and measured values, there are still clear gaps and the gaps are more pronounced in acceleration than gyro. Apart from the pattern, there is also a difference in the actual numeric values.

We claim that there are two underlying reasons due to which the computed and measured values are different.

(1) Contributions from hand/finger movements. First, we note that the difference between computed and



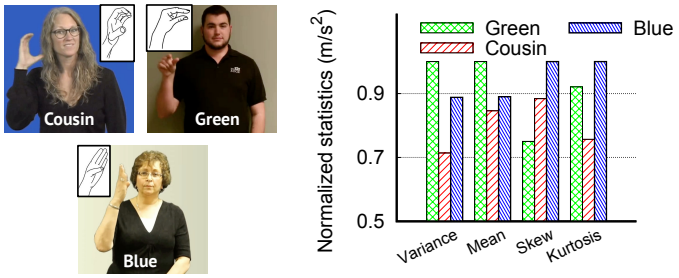
(a) Computed acceleration (b) Computed gyro

(c) Measured acceleration (d) Measured gyro

Fig. 6. Comparing manually computed and measured acceleration and gyro for an ASL gesture.

corresponding measured acceleration values is not consistent for different parts of the gestures. The samples in the highlighted area (marked in red Fig. 6a) for computed acceleration values are relatively lower than their respective measured acceleration values (marked in red Fig. 6c). The reason behind this difference is that the highlighted part of the gesture is only comprised of finger movements while the other not highlighted parts are comprised of arm movements. As established by multiple prior works [33, 34], IMUs also capture acceleration values corresponding to finger movements because of the connection between finger bones and muscles in the forearms. The computed acceleration does not account for the impact of finger movements in the acceleration as they are directly calculated from wrist displacements only.

(2) Same wrist movements but different hand shapes.



(a) Similar movements but different handshapes (b) Different observed acceleration due to different handshapes

Fig. 7. Impact of hand shape on observed acceleration.

As IMUs measure acceleration as a function of change in capacitance observed along a damp suspended mass, they are also impacted by the mass of the hand as observed in prior works [35, 36]. This means that the hand shape and its relative position to the wrist should have an observable impact on the measured acceleration while performing a gesture. We claim that just changing the shape of the hand in a gesture without any other change in wrist and arm movements should change the acceleration values observed at the wrist. We verify the claim by picking a set of gestures that vary only in the hand shape but have the same wrist and arm joint movements and comparing their acceleration values. Fig. 7a shows three such gestures and their corresponding hand shapes. The three gestures `Cousin`, `Green`, and `Blue` involve the same wrist rotations with different hand shapes. To compare the three gestures acceleration values, we use statistical features (first four movements - mean, variance, skewness, and kurtosis) that have been used in prior works for IMU-based gesture classification [33, 37]. Fig. 7b shows the acceleration statistics for the three gestures. The statistics were calculated for 100 instances per gesture (5 users each performing 20 instances per gesture). We compute an average over the three axis and normalize them for visual comparison. From Fig. 7b, it is clear that the statistics for the gesture are different. This difference in acceleration is due to the difference in the hand shapes given that all other characteristics of the three gestures remain the same. This clearly demonstrates the impact of hand shapes on the measured acceleration values and reaffirms the need to incorporate hand joint information in acceleration estimation.

4.2 Learning to Estimate Accel. & Gyro

Based on the above mentioned two reasons, our aim is to find a function that can take the transformed 3D wrist displacement values and incorporate the critically important 2D hand joint and shape information while computing the acceleration. Efforts towards deriving such a function is further complicated by the challenge of missing hand joints as we explained in Section 3. Given the complexity, a possible approach would be to use supervised learning where we can train a model to learn the function by supervising with the measured acceleration values. Care is needed in designing such a model as these models can overfit and disregard features crucial to the task. Here, we find an opportunity that enables us to address these challenges of overfitting by incorporating the gyro information during training.

4.2.1 Multitask learning of acceleration and gyro. Multitask learning [38] is a learning technique designed for simultaneous learning of multiple related tasks. By exploiting the commonality and differences across the related tasks, it generalizes well for all the tasks. Multitask learning forces *the tasks to focus attention* on only the relevant feature representations [39]. Specifically, in our case, it enables the acceleration model to pay attention to hand joint information when wrist displacement values are not significant. This could happen in gestures with wrist twists without considerable wrist displacement [40]. Here, training with gyro could assist the acceleration model in understanding if the acceleration is due to a rotation movement of wrist or due to change in hand shape (i.e., finger and hand movement). Additionally, the complementary nature of acceleration and gyro (they capture different representations of the same movement) help us reduce the overfitting. Fig. 8 shows the proposed model. Our model consists of two tasks, one for acceleration estimation and another for gyro estimation. The tasks share a common feature representation layer which is comprised of a linear layer followed by two layers of LSTM cells with dropout in between. The LSTM cells model the time-dependent nature of acceleration and gyro estimation. Additionally, LSTMs because of their ability to model short-term and long-term dependencies also help in addressing the problem of missing hand joints by incorporating the knowledge from key frames in the estimations for delta frames as explained in the Section 3. Following the shared layers, we have task specific layers for individual tasks. Each task specific layers is comprised of linear layers with batch normalization for normalizing the inputs, followed by ReLU for activation. This is followed by linear layer for IMU acceleration and gyro prediction. For both linear layers and LSTM cells we use 1024 hidden units and set the dropout to 0.5. We train the model using the mean square error loss.

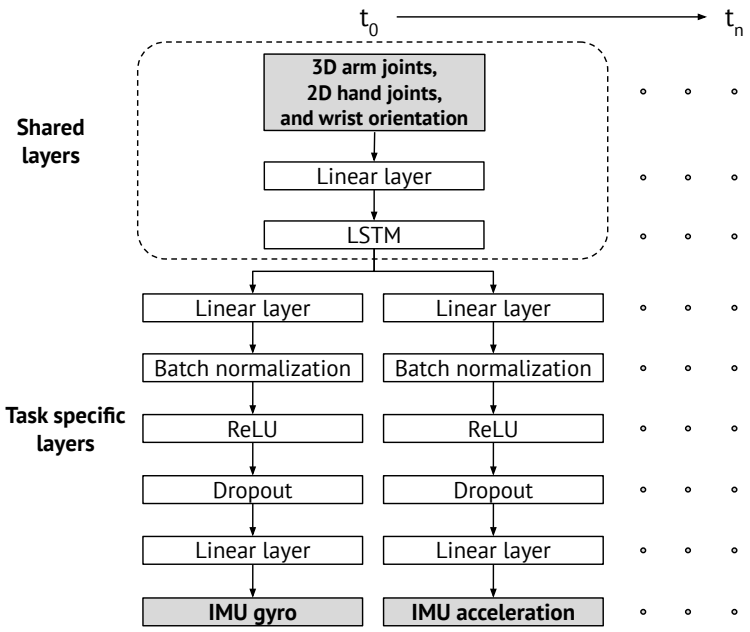


Fig. 8. Our multitask learning model for acceleration and gyro prediction.

5 EVALUATION AND NUMERICAL RESULTS

5.1 Datasets, Training and Implementation

We evaluate the two modules of Vi2IMU individually, in series, and in an end-to-end video to IMU translation. Table 2 and Fig. 10 summarize different datasets used, input and output for models, method of evaluation, and corresponding training and testing data. As we detailed in Section 2, we extract the 2D joint positions for the body and hand using openpose [23]. Following this, we use an existing 3D pose estimation model [26] for converting the obtained 2D body key points. We retrain the 3D pose estimation models using our collected data for Vi2IMU with only the upper body joint positions (with head as origin) to accommodate for the absence of lower body joints in many existing in-the-wild ASL video datasets.

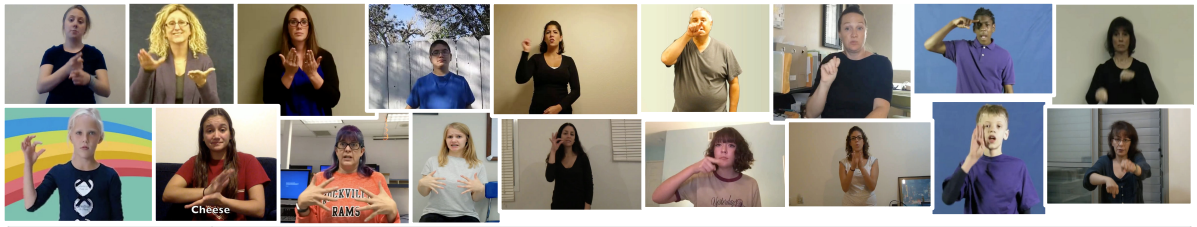
5.1.1 Vi2IMU dataset. Both orientation and acceleration/gyro estimation modules require a dataset that includes videos, ground truth/measured 3D joint positions, and orientation, acceleration, and gyro of the wrist IMU. To the best of our knowledge, there is no publicly available dataset that can provide all these. To address this problem, we embark on our own data collection effort referred as Vi2IMU dataset. Our dataset uses Azure Kinect camera Development Kit [41] and Google Pixel phones [42] with IMUs. We use Azure camera that provides RGB videos and corresponding depth information, which is then subsequently processed using Azure body tracking model [43] to estimate the 2D and 3D joint positions and displacement. We ask 5 subjects (the study is IRB approved) to perform different arm and hand gestures. The Pixel phone is also attached to the user's wrist to simultaneously collect the IMU data. We pick a sequence of gestures involving arm and hand movements as cue videos. The subjects are asked to perform the actions at varying speeds so that the model can learn acceleration as a function of displacement change at different rates. We collect data from 5 subjects for approximately 16 hours resulting in approximately 1.6M frames as summarized in Table 2.

Table 2. Summary of Vi2IMU dataset used for module evaluation

Evaluation	Dataset	Input	Output	Training and Testing
M2: Wrist orientation estimation	Vi2IMU	2D arm and hand joints from videos	Orientation of wrist IMU in camera's frame-of-reference	5 subjects, 1.6 M samples 1) 80% - 20%: Training on 1.4M samples Testing on 220K samples 2) Leave one out: Training on 4 users' data Testing on 1 user data
M3: Wrist acceleration & gyro estimation	Vi2IMU	3D arm joints, 2D hand joints and wrist orientation	Wrist IMU acceleration & gyro	

The orientation and accel/gyro estimation modules are trained using 1.4M frames (80% data) and corresponding IMU data and tested with 220K (20% data) frames for 5 subjects. We also evaluate the modules in a "leave-one-out" fashion where the model is trained on 4 users' data and tested on the remaining user. We do this for all 5 users.

5.1.2 MS-ASL dataset for translating in-the-wild videos. For the final evaluation of Vi2IMU, we take an in-the-wild ASL video dataset and convert the sign language gesture videos to corresponding wrist IMU and then train a sign language gesture recognition model using the translated IMU data. The model is then tested with IMU gesture samples (for the same gestures) collected by us. High accuracy in recognizing these gestures during the testing indicates that our video to IMU translation framework performs well in producing wrist IMU data that matches well with real measured data.



End-to-end evaluation method	- Use pre-trained models to translate IMU acceleration and gyro from sign language gesture videos. - Translate from videos for 70 gestures (with ≈ 72 subjects) and 20-25 video instances per gesture. - Train a sign language recognition model using the translated IMU acceleration and gyro data. - Use measured IMU test samples for 5 users to evaluate the recognition model.

Fig. 9. MS-ASL dataset: diversity of users, backdrop, lighting conditions, and part of user's body visible in videos.

We use the MS-ASL dataset [7] which is a benchmark dataset proposed for word-level ASL recognition. We pick 70 gestures from the dataset with 20-25 videos per gesture and convert them into IMU data. The dataset was created by curating YouTube videos and manually labelling them. As seen in Fig. 9, the dataset is comprised of a diverse set of users of different age groups and physical builds. The dataset is also diverse in terms of the video resolution, backdrop, lighting conditions, user posture (sitting/standing), and part of the user's body visible in the videos. The video resolutions range from 360p to 1080p with approximately 35% videos having resolution lower than 480p. There are 72 different subjects in our 70 gestures. Translating from such a diverse dataset should help in addressing the user diversity problem inherent in IMU datasets. Fig. 10 shows the phonological properties

of the chosen ASL signs. The signs include an approximately equal number of repetitive and non-repetitive signs, 40% of them are one-handed, and 22% are with wrist twists. In repetitive signs, part of the sign is repeated multiple times within the sign (like knocking movement repeated twice in knock-knock). Fig. 10 shows the wrist movement type for the chosen ASL signs. Wrist movement type denotes the translational displacement of the wrist while wrist twist denotes the rotational movement of the wrist. Here, 55% signs have straight wrist movements, 18% involve curved wrist movements, 17% have no wrist movement, and the remaining ones have circular wrist movements. The phonological properties and movement information were obtained from [27]. Once we translate the data, we use it for training an IMU sign gesture recognition model and test with the IMU gesture samples collected by us.

Wrist movement type	No. of gestures (out of 70)
Circular	6
Curved	13
Straight	39
None	12

Phonological property of gestures	No. of gestures (out of 70)
Gesture involves a **repetitive** motion	34
Gesture involves a **wrist twist**	16
Gesture is **one-handed**	28

Fig. 10. MS-ASL dataset: phonological properties and wrist movement type for the 70 signs considered.

5.1.3 Model implementation. All the proposed models are implemented using Pytorch [44] and optimized using Adam optimizer [45]. The models are trained for 100 epochs, where one epoch is the time taken by the network to perform one iteration of training (feed forward, compute the loss, and backpropagate the losses) on the entire training set. We start with a learning rate of 0.001 and use an exponential weight decay to reduce the learning rate beyond 75 epochs. We set the batch size to 24 for the orientation estimation model and 5 for the acceleration/gyro estimation model. We pick the best model using the early stopping [46] approach, where we use a validation set to evaluate the performance of the model every 3 epochs and stop at the point where there is no observable improvement in performance on the validation set. All the models were trained using NVIDIA Tesla K480 GPUs and the training time for the orientation model was approximately 3 hours and for the acceleration/gyro estimation model was approximately 2.5 hours.

5.2 Orientation Estimation Results

5.2.1 Comparison models and error metric. We compare our orientation prediction model with two other models: (1) *Non-deep learning model:* We use Ridge regression [47] trained using the same training data (Table 2) as a baseline for comparison. Ridge regression utilizes linear least squares as the loss function in determining the parameters and uses L^2-norm for regularization. Regularizing with L^2-norm constraints the parameter values and has shown to achieve better generalization [48]. We use a regularization strength of 1 in training the model. (2) Vi2IMU *model without hand joints:* To understand the impact of the hand joints, we train the proposed deep learning model without hand joints and only with the arm joints (shoulder, elbow, and wrist). Comparing with such a model should give some insight into the role that hand joints play in orientation estimation. Also, the orientation values estimated by the model without hand joints (just arm joints) is similar to the orientation values obtained through forward kinematics. We use *mean absolute error* (MAE) in degrees which gives the per-axis orientation error highlighting the contributions in error along the different axes as the error metric. Let N be the number of test samples, then the MAE is given by $MAE = (1/N)\sum_{i=1}^{N}|v_p - v_g|$ where v_p and v_g are predicted and ground truth orientations, respectively.

5.2.2 Comparing with baseline model and Vi2IMU without hand joints. Fig. 11a shows the per-axis orientation error in degrees for Vi2IMU and Ridge regression. We observe that Vi2IMU's deep learning model achieves on

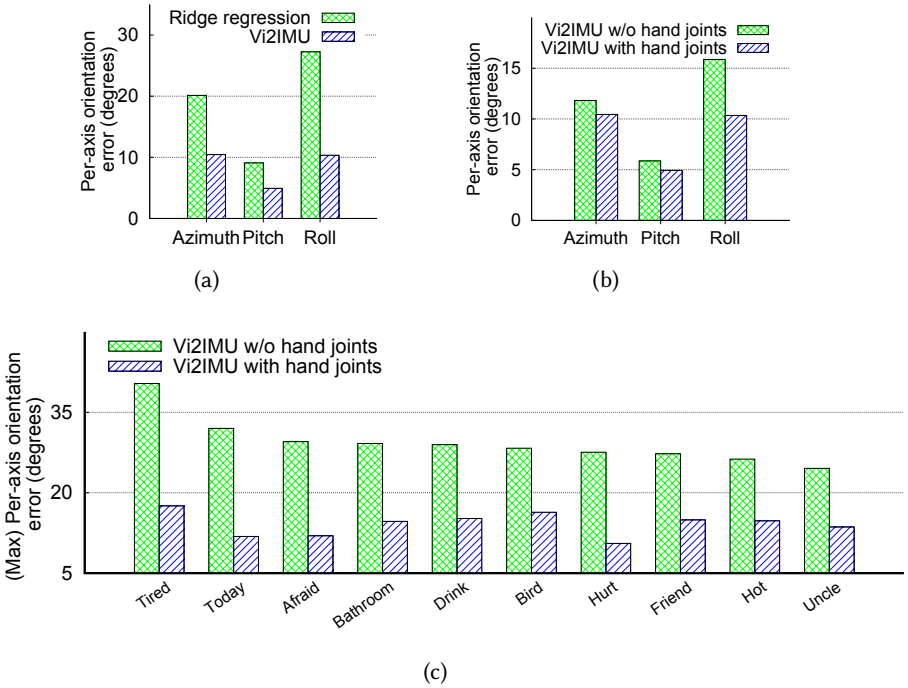


Fig. 11. Orientation estimation performance: (a) Comparison with regression, and (b,c) performance with and w/o hand joints

average 54.48% less error (8.57°) compared to the ridge regression (18.83°). This is expected because of the ability of deep learning models to better learn the problem-specific representations compared to other models. Of the three axes, the error is the highest for roll (IMU's Y-axis). As we detailed in Section 3, hand joints offer the required information to estimate the rotations along the IMU's Y-axis (roll) as it is possible to rotate the wrist along the Y-axis without moving the arm. Although the ridge regression model has hand joint information, unlike Vi2IMU, it is not able to account for the impact of inaccurate hand joint estimates (missing hand joints) which requires incorporating the knowledge from key frames in the orientation estimation for delta frames.

While comparing the deep learning models with and without hand joints, we find that for the model without hand joints, the average increase in error in the azimuth and pitch is 13.65% (Fig. 11b). Here, we observe that the error is relatively less in azimuth and pitch as they can be estimated using arm joint information. In contrast, the error for roll increases considerably (34.7% increase) when Vi2IMU is trained without hand joints. This clearly establishes the significance of hand joints. Fig. 11c shows the maximum (among three axes) orientation error for 10 gestures for Vi2IMU with and without hand joints. The gestures vary in the amount of wrist displacement and rotations. The average and maximum decrease in error is 52.02% and 61.71%, respectively, when Vi2IMU is trained with hand joints. The gesture with maximum decrease in error (`Hurt`[49]) predominantly involves wrist rotations, leading the model with hand joints to have a better estimation of orientation.

5.2.3 Impact of untrained users. Fig. 12a shows the per-axis orientation error for orientation estimation while the models with and without hand joints are evaluated in a leave one out fashion. Here, we train the models on 4 users and test on the remaining user. We repeat this for all the 5 users and provide the average of the obtained results. This should give us some insight into the generalizability of the proposed models. We observe that the deep learning model trained with hand joints has 13.04% less error on average. The difference is highest for roll

where the model with hand joints has an error of 17.8° and the model without hand joints has an error of 24.26° (26.6% increase in error). This further emphasizes the need for hand joint positions in estimating orientation.

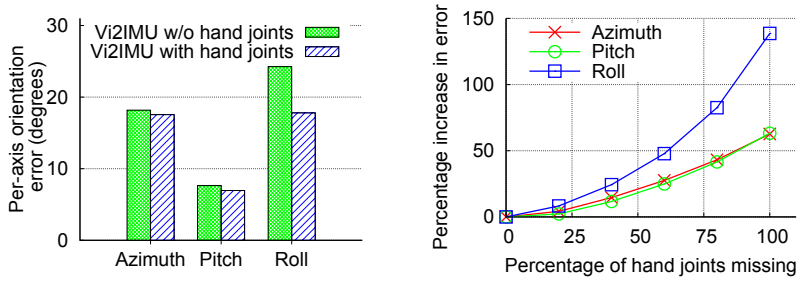


Fig. 12. Orientation estimation performance: (a) impact of untrained users and (b) impact of missing hand joints

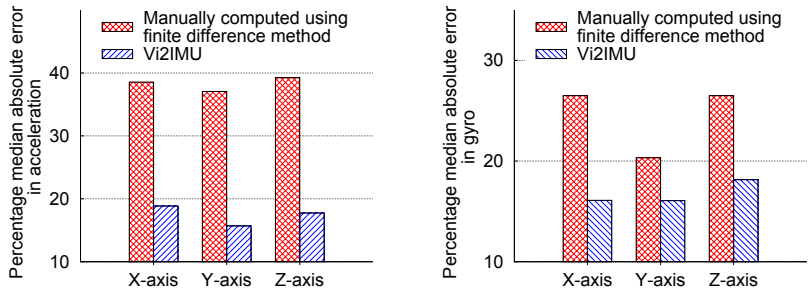
5.2.4 Impact of missing hand joints. We investigate the robustness of the proposed deep learning model in coping with inaccurate hand joint estimates by testing the model with different number of hand joints being available (accurately estimated). We do this by randomly masking some percentage of the hand joints to zero in the test data and estimating orientation with the masked hand joints (considered missing hand joints by the model). This is done on top of the existing missing hand joints which the model already has to accommodate for in orientation estimation. Fig. 12b shows the observed percentage increase in error along the three axes as we increase the amount of missing hand joints. We observe that until a 50% increase in the number of missing hand joints, the percentage increase in error is less than 25% for roll (still less than the model without hand joints) and less than 15% for azimuth and pitch. Here, Vi2IMU's ability to incorporate the knowledge from the key frames (like hand shape) into the orientation estimation for frames with missing hand joints (delta frames) makes it resilient. However, as we increase the missing hand joints beyond 50%, the percentage increase in error along the roll increases exponentially. This is expected, as the model does not have any key frames (all the frames have missing joints) to depend on, and the observed error in roll (24.32°) reaches close to that of the baseline model.

5.3 Acceleration & Gyro Estimation Results

5.3.1 Comparison and error metric. We compare our multitask learning model with two other schemes. (1) *Manual computation using finite difference:* The manually computed acceleration/gyro (also the approach used in [11]) is calculated as follows: we use the predicted wrist orientation to transform the obtained 3D wrist displacement values into IMU's frame-of-reference. We then compute wrist acceleration as the second-order derivative of displacement change between every subsequent sample. For the calculation of gyro, we calculate the first order derivative of orientation change between every subsequent sample of the predicted orientation values. (2) Vi2IMU *model without hand joints:* To evaluate the contributions from hand joints, we train another model using the same training data where we do not input the 2D hand joints to the model. Here, we input orientation predictions as predicted by the orientation estimation model trained without hand joints.

We use *percentage median absolute error* (PMAE) as the error measure for both acceleration and gyro evaluation. We obtain PMAE by computing the median of the absolute difference between the predicted and ground truth acceleration (or gyro) vectors and dividing it by the mean acceleration (or gyro) value observed in our collected dataset. The PMAE is separately calculated for all three axes. *PMAE provides insight into how the observed errors stand compared to typically observed acceleration and gyro values during gestures.*

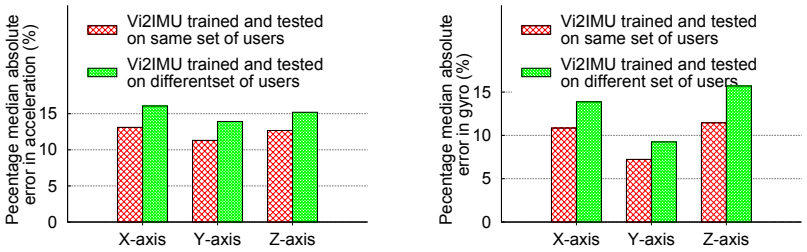
5.3.2 Comparison with manual computation. Figs. 13a and 13b show the comparison between manual computation using finite difference method and Vi2IMU's proposed model with multitask. We observe a significant decrease



(a) Accelerometer (b) Gyro

Fig. 13. Vi2IMU's multitask learning model vs. direct manual computation of acceleration and gyro.

in error for both acceleration and gyro along the three axes when manual computations are compared with Vi2IMU trained with multitask. Comparing manual computation with Vi2IMU, we observe a decrease in error from 1.34 m/s^2 to 0.54 m/s^2 for acceleration and from 0.55 rad/s to 0.39 rad/s for gyro when averaged over the three axes. As stated earlier, the major reason behind this is that the manual computation does not take the hand joint information into account for acceleration estimation. Similarly, since the manual computation does not incorporate the knowledge of domain difference between videos and IMUs in terms of bias and noise, the error for gyro is also considerably higher compared to that of Vi2IMU.

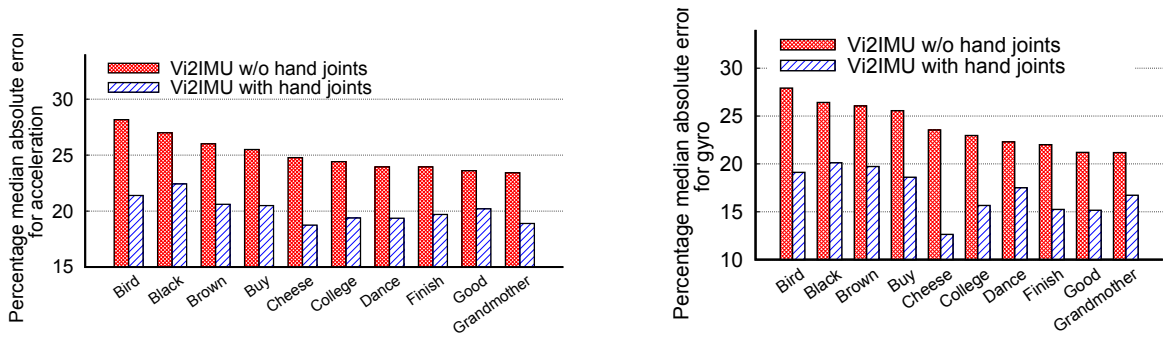


(a) Accelerometer (b) Gyro

Fig. 14. Vi2IMU's model trained and tested on the same set of users vs. different sets of users.

5.3.3 Training and testing on different users. Figs. 14a and 14b show the results for Vi2IMU when trained and tested on different set of users. Here, we train Vi2IMU on a subset of four users and test it on the remaining one. We repeat this for a different subset of users and provide the average of the observed errors. We find that although there is some increase in error when compared to trained and tested on the same set of users, the increase is still not significant (2.69% and 3.03% on average for acceleration and gyro). This shows that the multitask learning model generalizes well by learning more user-independent feature representations as originally intended. We will further evaluate this cross-user generalization capability of Vi2IMU in Section 5.4 with in-the-wild video evaluation.

5.3.4 Importance of hand joints in accel./gyro prediction. Figs. 15a and 15b show the comparison for two models (with and without hand joints). For the sake of brevity, we present the sum of PMAE for all three axes for both models and show the results for the top 10 worst performing gestures (in terms of error achieved by the model without hand joints for both acceleration and gyro). The maximum increase in acceleration error is 6.77% and the average increase in acceleration error across all the gestures is 4.67%. We find that 6 out of the 10 worst-performing gestures actually involve a considerably small amount of wrist displacement where correct



(a) Accelerometer (b) Gyro

Fig. 15. Impact of including hand joints on the performance of Vi2IMU for different gestures.

Table 3. Cumulative error of displacement and orientation estimation on accel. & gyro predictions.

3D arm joint positions / Wrist orientation	Percentage median absolute error (%)					
	Acceleration			Gyro		
	X	Y	Z	X	Y	Z
Ground truth/ Ground truth	13.11	11.31	12.68	10.85	10.83	11.45
Predicted/ Ground truth	13.45	11.68	13	11.1	11.06	11.85
Ground truth/ Predicted	18.74	15.4	17.57	15.95	16.1	18
Predicted/ Predicted	**18.85**	**15.68**	**17.74**	**16.1**	**16.06**	**18.15**

acceleration and gyro estimation requires relying on hand joints. For example, ASL gesture `Cheese` [50] involves a considerable amount of wrist rotation without displacement. Hence, the model without hand joint positions performs significantly worse with error increasing by 10.91%.

5.3.5 Understanding error accumulation. We now try and understand how errors in orientation estimation contribute to the error observed in acceleration and gyro estimation. In order to do this, we take the trained model for accel./gyro prediction and test it using three variants of input data: (i) use ground truth displacement of joints and ground truth wrist orientation as input, (ii) use predicted displacement of joints but ground truth wrist orientation as input, and (iii) use ground truth displacement of joints but predicted wrist orientation as input. Along with these, we use the 2D hand joint displacements extracted using OpenPose as input for all three variants. We compare these three with our model where both displacement and orientation are predicted (i.e., error accumulates). We summarize the results in Table 3. We observe that when the displacement is predicted but orientation is not, the error is similar to that of the model where both of them are ground truth. In comparison, we see an increase in error when orientation is predicted and displacement is not, and the observed error here is similar to that observed when both orientation and displacement are predicted. This means that the orientation

Table 4. Classification performance for different number of MSASL gesture classes when the model is (a) purely trained using Vi2IMU translated IMU data, (b) trained using translated IMU data with augmentation, (c) real measured IMU data and (d) real measured IMU data along with translated augmented IMU data.

Number of MS-ASL gestures	Training data											
	(a) Translated IMU data without augmentation			(b) Translated IMU data with augmentation			(c) Measured IMU data			(d) Measured + translated IMU data		
	Top-1 (%)	Top-3 (%)	Top-5 (%)	Top-1 (%)	Top-3 (%)	Top-5 (%)	Top-1 (%)	Top-3 (%)	Top-5 (%)	Top-1 (%)	Top-3 (%)	Top-5 (%)
50	84.1	96.8	98.8	90.7	100	100	91.6	100	100	**100**	100	100
60	74.7	93.2	98.1	84.2	99.9	100	89.6	100	100	**99.7**	100	100
70	66.6	86.2	93.4	77.3	98.2	100	86	100	100	**93.1**	100	100

estimation module introduces more errors in acceleration and gyro estimation compared to displacement. This can be explained by the fact that the displacement estimation module uses existing pose estimation approaches that have been researched upon for many years now and are expected to perform better than the fine-grained wrist orientation estimation problem which is relatively less explored.

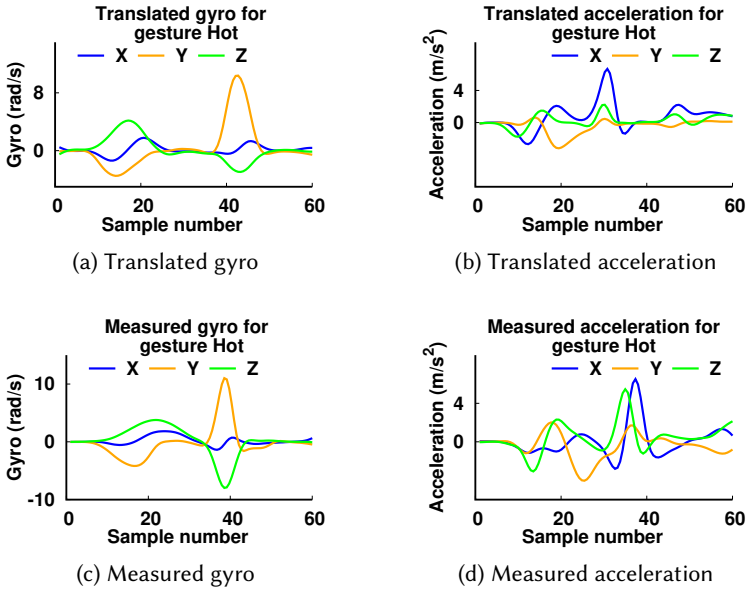
5.4 Translating in-the-wild Videos

Our final evaluation is to leverage Vi2IMU to translate videos to IMU and train an IMU-based ML model without the need for any actual IMU data for training. As mentioned earlier, we use an in-the-wild ASL video dataset referred to as MS-ASL [7] and arbitrarily pick 70 ASL gestures of varying characteristics. Next, we translate the videos corresponding to the chosen gestures using Vi2IMU. We then use the translated IMU data to train an ASL gesture recognition model without any IMU data collected specifically for training. We then collect test samples from wrist IMU for the same set of gestures (25 test instances for each of the 70 gestures). The model trained using the translated IMU data is then evaluated using the test IMU data.

5.4.1 Gesture classification model and metrics. We use the state-of-the-art deep learning model proposed in [1] for smart watch-based ASL gesture recognition as the machine learning model. The model is comprised of three bi-directional LSTM layers, which are followed by a fully connected linear layer and a softmax layer for classification. For evaluating the trained machine learning model, we use average accuracy as the metric [51]. In addition to the Top-1 accuracy, we also provide Top-3 and Top-5 accuracies. For Top-k accuracy, a sample is considered correctly classified if the label corresponding to the sample appears in one of the top k predictions.

5.4.2 Varying number of gestures. We start by understanding the Vi2IMU's ability to scale for a diverse set of gestures. Since the IMU data translated from videos can slightly differ from the ones obtained from real IMUs, domain adaptation between the source domain (videos) and the target domain (IMUs) is necessary. Existing works [11, 17] have addressed this through distribution matching, DTW-based stretching, and data augmentation techniques. To cope with such domain differences, we propose a simple augmentation technique in which we incorporate the attributes from the measured IMU data by scaling the translated data to the same range as the measured data. As most of the in-the-wild videos we obtained are educational videos, the gestures are often performed slowly to communicate the proper movement information. This can result in lower acceleration and gyro values in the translated data. Addressing this issue can minimize domain differences and improve the performance of the translated data.

In our evaluation, we consider three training strategies: (i) translated data without any augmentation, (ii) translated data with augmentation from 4 measured IMU samples per gesture, and (iii) a combination of measured



(a) Translated gyro (b) Translated acceleration

(c) Measured gyro (d) Measured acceleration

Fig. 16. Comparing translated and measured acceleration and gyro for an ASL gesture.

IMU data and translated data with augmentation from 2 measured samples. We separately evaluate the impact of the proposed data augmentation technique later in Section 5.4.8 to understand the benefits gained from increasing the number of measured samples used for augmentation and the importance of user and gesture specific attributes in augmentation.

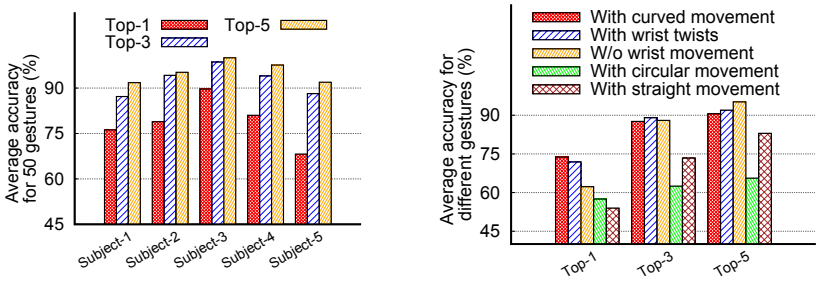
Table 4 shows the performance of the trained model for a different number of gestures. Comparing the models with and without augmentation of translated data, we find that there is on average 16.93% increase in accuracy when trained with augmentation. We note that explicitly addressing domain differences is important to improve the performance of the translated data. Our approach for domain adaptation is simple (only requires scaling) and can still provide a significant improvement in performance with just four measured samples per gesture. Additionally, while the model trained using the translated data achieves lower Top-1 accuracy, it does achieve better performance in terms of Top-3 and Top-5 accuracies. Of the three training strategies, the model trained with a combination of measured and translated IMU data significantly outperforms the other two models, and the model trained entirely with the measured IMU data. On average, there is an 8.53% increase in accuracy when the model is trained using a combination of measured and translated IMU data. The reasons are twofold: (1) the translated data incorporates information from a diverse set of users and (2) it increases the amount of training data, directly addressing the data scarcity problem.

Fig. 16 shows the translated acceleration and gyro values along with the corresponding measured and gyro values for ASL gesture `Hot`. We find that the pattern of translated and measured accelerometer and gyro data have considerable visual similarities. More such comparisons are available at our anonymous data repo [52].

5.4.3 Performance for different subjects. Figs. 17a shows the Top-1, Top-3, and Top-5 accuracies for the 5 subjects for 50 gestures. The observed standard deviations from the average Top-1, Top-3 and Top-5 accuracies for 50 classes are 7.70%, 4.72%, and 3.57% respectively. The reason behind this observed difference across subjects for Top-1 accuracy is that the gestures performed by Subject-5 were relatively different from the actual ASL

gestures (from which the data was translated) for a number of gestures. This is also evident in the Top-3 and Top-5 accuracies for Subject-5 which are relatively more consistent to that of the other subjects.

5.4.4 Performance for different gesture types. Here, we use the phonological properties of ASL that we explained earlier in Fig. 10. Fig. 17b shows the results for different sets of gestures grouped by their phonological properties. We note that the set of gestures without much wrist movement (the categories with wrist twist and w/o wrist movement) offer relatively good performance (Top-3 accuracy of 88% and 89.06% respectively) when compared to gestures with considerable wrist movement (curved movement with Top-3 accuracy of 87.6%). As Vi2IMU incorporates hand joints information in both orientation and acceleration estimation, there is no clear observable difference in translations for gestures with and without wrist movements. Compared to other gesture types, gestures with straight movement and circular movement have lower accuracies.



(a) Different subjects

(b) Different gesture types

Sensors considered for gesture recognition	Top-1 accuracy (%)	Top-3 accuracy (%)	Top-5 accuracy (%)
Accelerometer	48.8	74.6	84.0
Gyro	58.6	84.9	92.4
Accelerometer and gyro	84.1	96.8	98.8

(c) Different predicted IMU sensor data in classification

Fig. 17. Comparing performance of Vi2IMU's translated data for gesture recognition.

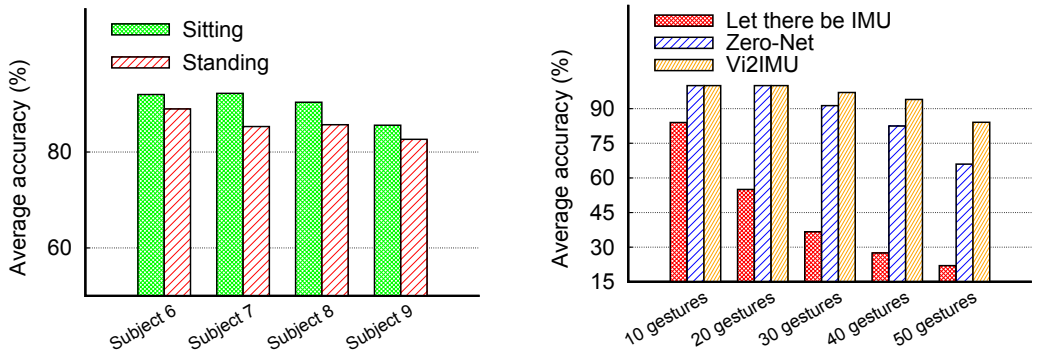
There are two reasons for this low performance. First, MS-ASL dataset is comprised of multiple paired words like father/mother, grandfather/grandmother, sister/brother, etc. for which the gestures have similar hand shape and movement except for the location. For example, the only difference between the gestures sister/brother is that while brother[28] is performed near the forehead sister [53] is performed near the chin. There are 10 such pairs among the gestures for which the wrist movement is straight causing significant confusion which results in low accuracy. The other reason is that some gestures like slow [54] and please [55] (gesture with circular wrist movement) are performed slowly which translates into low acceleration/gyro values. The low acceleration/gyro values make it difficult for the model to differentiate between the different gestures.

5.4.5 Contributions of translated acceleration and gyro in classification. We study the impact of the translated acceleration and gyro values on the observed classification results by training two separate models: one with the acceleration data and the other with the gyro data for classifying 50 gestures. Fig. 17c shows the results. We find that the impact of the predicted gyro values is relatively high. There is a 9.8% increase in Top-1 accuracy for the model trained only with the translated gyro values when compared with the model trained only with

the translated acceleration values. The rotational information obtained from the gyro provides the orientation change that is crucial in detecting wrist rotations. As the MS-ASL dataset includes several gestures that involve wrist twists without any wrist movement, the model trained only with translated gyro performs relatively better than the one trained with translated acceleration alone.

5.4.6 Performance under diverse settings. To understand the ability of the synthesized data to scale for diverse settings and users, we collect an additional IMU data set with four subjects. The subjects perform 50 gestures, each for 10 instances per gesture while assuming two body postures, sitting and standing. The dataset comprises three males and a female, aged between 26-33, with heights between 160-182 cm, and weights between 58-75 Kg. The users vary in their ASL proficiency ranging from no experience to non-native beginner's experience. Additionally, we do not impose any constraints on how the gesture is performed and encourage a casual posture to account for everyday usage. The collection time for each user was approximately 6 hours (excluding the time for breaks, instructions, etc.) with an approximate total time of 24 hours, spread over 6 days.

We test the collected data on the model trained using translated IMU data from MS-ASL dataset. Fig. 18a shows the results for different subjects in the two settings. The average accuracy for sitting and standing is 90.06% and 85.67% respectively. The observed performance reflects the benefit of translating from in-the-wild diverse videos with a large number of users performing gestures in different postures (approximately, 58% standing and 42% sitting), speeds, etc., and the ability of the translated data to scale for diverse unseen subjects and settings. We also find that on average there is a slight increase in accuracy (4.3%) when the subjects perform the gesture sitting than standing. As standing is a bit strenuous compared to sitting, there is occasional involuntary limb movements, which result in this difference in accuracy between the two postures. These involuntary limb movements are reflected as noise in the collected IMU data and result in this drop in accuracy.



(a) Performance under diverse settings (b) Comparison with existing works

Fig. 18. Vi2IMU's performance (a) under diverse settings and (b) compared to existing works

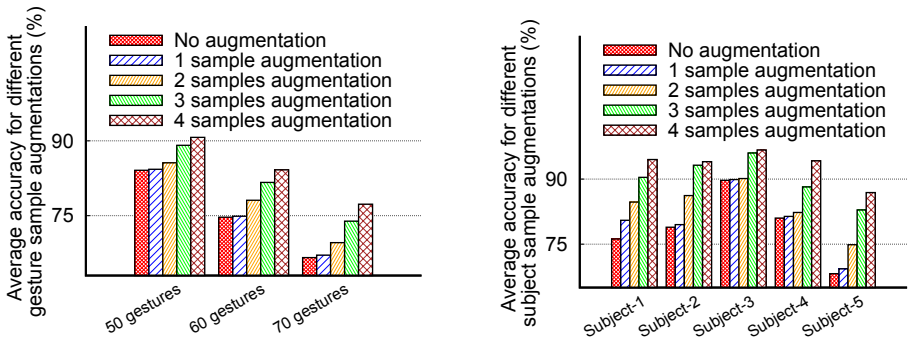
5.4.7 Comparison with existing works. We compare the performance of Vi2IMU with two existing works, "Let there be IMU" [14] and "Zero-Net" [11]. "Let there be IMU", proposed for human-activity recognition, uses a generative approach comprised of a multi-level CNN (convolutional neural network) architecture that takes as input a sequence of poses and directly regresses the corresponding IMU data. Instead of predicting the 3-axis acceleration and gyro data, the proposed approach predicts the norm of the acceleration and gyro data. We implement the proposed multi-level CNN architecture and use the upper-body joint pose sequences as input and the norms of acceleration, and gyro as output to train the model. "Zero-Net", which takes a trajectory-based approach, estimates the 3-D position of the finger joint corresponding to the ring position and computes the acceleration using finite differences. In place of the gyro data, their approach is to use the angle between the Y-axis

of the IMU and its projection on the X-Z plane of the TCN (torso coordinate system). To obtain this angle from the videos, they approximate the Y-axis of the IMU with the line joining two finger joints (metacarpophalangeal (MCP) and proximal interphalangeal (PIP) joints). In place of the 3D finger joint, we use the 3D wrist joint as an approximation for the smartwatch position from the videos and use finite differences to compute acceleration. For estimating the angle we use the line joining the wrist and elbow joints to approximate the Y-axis of the IMU and use its projection on the X-Z plane of the TCN.

We translate the videos from the MS-ASL dataset for 50 gesture classes using the implementations of "Let there be IMU", "Zero-Net", and Vi2IMU and use the translated data to train a model for ASL recognition. We repeat the training for a varying number of gestures from 10 to 50 and evaluate the trained model on the measured IMU data from five different subjects. Fig. 18b shows the results for the three approaches with a varying number of gestures. We find that compared to the generative approach ("Let there be IMU"), the trajectory-based approaches perform better when scaled for a large number of gestures. While this reinforces the importance of a systematic trajectory-based approach, we also acknowledge that the approach proposed in "Let there be IMU" was evaluated on ten human activities and depends on multi-sensor input from different joints for its prediction. It is expected that the method cannot scale for a large number of gestures with only the sensor information from a single joint. In comparing Vi2IMU and "Zero-Net", we find that as the number of gesture classes increases the performance gap also increases. On average, there is an 11.67% increase in accuracy when the model is trained using the data translated by Vi2IMU while scaling the number of gestures above twenty. As the videos we translate from are diverse, pose estimation on them often performs poorly, and manual computation of acceleration leads to considerable errors as we have highlighted in Fig. 13a. While "Zero-Net" was shown to perform well for a single source of high-quality videos, only depending on such videos reduces the number of videos available for translation. Additionally, gyro data is crucial for differentiating between different ASL gestures that have similar movement information but different orientations. As Vi2IMU can estimate 3-axis gyro data and also incorporate the information of missing joints in its models, it can better scale for a large number of gestures in comparison with "Zero-Net".

5.4.8 Understanding the impact of data augmentation. We study the benefits of data augmentation on the performance of the translated data through (i) gesture-specific augmentation and (ii) subject-specific augmentation. For both augmentations, we incorporate the attributes from the measured IMU data into the translated IMU data by scaling the translated data to the same range as the measured data. We propose this augmentation based on two observations. First, most of the in-the-wild ASL videos are educational videos, where the user performs the ASL gesture at lower speeds to communicate the proper hand movement and shape. This results in lower acceleration and gyro values in the translated data. Next, the translated data does not capture some of the subject-specific attributes which could hinder the performance of specific subjects as seen in Fig. 17a. Thus, we hypothesize that correcting for the gesture speed and incorporating subject-specific attributes could improve performance. For correcting the gesture speed, we perform gesture-specific augmentation, where we pick an arbitrary gesture sample and use its measured range to scale each of the gesture samples in the translated data. We repeat this for all the gestures. We increase the number of measured gesture samples used for augmentation and repeat the mentioned steps for each of the measured gesture samples. The number of translated samples for training linearly increases with the number of measured samples used for augmentation. We train separate models for the different number of gesture sample augmentations and evaluate their performance. For subject-specific augmentation, instead of picking arbitrary gesture samples we pick gesture samples from a single subject for augmentation and train separate models for different subjects.

Fig. 19a shows the results for gesture-specific augmentation for a different number of ASL classes in the MSASL dataset. For all the classes, there is a significant increase in performance with data augmentation. The average increase in accuracy is 9.28% as we move from no augmentation to augmentation with four samples.



(a) Gesture-specific augmentation (b) Subject-specific augmentation

Fig. 19. Impact of data augmentation on the performance of Vi2IMU's translated data.

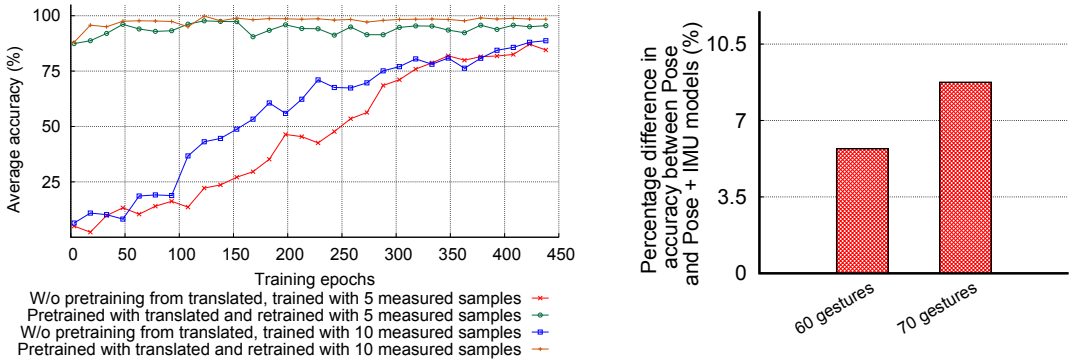
For 50 gestures, the model trained with four sample augmentation provides comparable accuracy (90.7%) to the model trained with measured IMU data. As we increase the translated samples by augmentation, the average accuracy gap between the translated and measured data reduces from 14.71% to 5.42%. We note that at least two gesture samples are needed for an observable difference in performance for all the gesture classes. In contrast to gesture-specific augmentation, in subject-specific augmentation, there is a significant difference in accuracy gain for the different number of samples used for augmentation. As seen in Fig. 19b, for subjects one, two, and five, just one or two samples were sufficient for an observable increase in accuracy. This confirms that the subject-specific attributes were not captured in the translated data and incorporating them leads to substantial improvement. The average increase in accuracy is 14.4% with a maximum increase in accuracy of 18.68% for Subject-5. Additionally, with subject-specific augmentation, the average accuracy for 50 classes is 93.26% which is better than the accuracy obtained from the measured data (91.6%).

5.4.9 Practical use cases for translated IMU data. We study the benefits of the translated IMU data under two scenarios: (1) using transfer learning, and (2) using multi-modal learning. The goal of these studies is to exhibit the potential of the translated IMU data in improving the research and development of IMU-based sensing and automatic ASL recognition.

(1) Transfer learning with translated IMU data. Research on transfer learning [56, 57] has shown that machine learning tasks with scarce training data for one domain (target domain), can benefit from pretraining with large data collected for a different domain (source domain). For example, the source domain on which pretraining happens could be pre-existing data collected for a set of subjects, and the target domain could be a new subject. Here, we explore if such benefits could be gained from our translated data for training with few measured samples. For this, we use two models: one is pretrained with the translated IMU data on 50 gestures, and another is not pretrained. We train both these models from scratch with a different number of measured IMU data samples from a single user and test it on gesture samples from four different users. The rationale behind this training/testing scenario is to highlight the importance of the translated data in better generalization with limited measured training data. We emphasize that the translated data we use for pretraining is not augmented with the characteristics of the measured data, and any benefits gained from the translated should be attributed only to the proposed system.

Fig. 20a shows the average accuracy for the two models (with pretraining and w/o pretraining from translated data) trained with 5 and 10 measured IMU samples for different epochs during the training. The model with pretraining outperforms the model without pretraining along two dimensions. First, the model with pretraining

yields maximum accuracy within 50 epochs of training which could translate to a reduction in the research and development cycle. Second, the model with pretraining yields maximum accuracy with a limited number of training samples. For example, there is a 15.9% difference in accuracy between the model trained with pretraining (97.7%) and the model without pretraining (81.8%) when trained with only five samples per gesture. This ability to better generalize from a few training samples is extremely important to tackle the data availability challenges. Additionally, the model with pretraining outperforms the model without pretraining in both scenarios further emphasizing the benefits of Vi2IMU and the translated IMU data.



(a) Transfer learning with translated data

(b) Multi-modal recognition using translated data

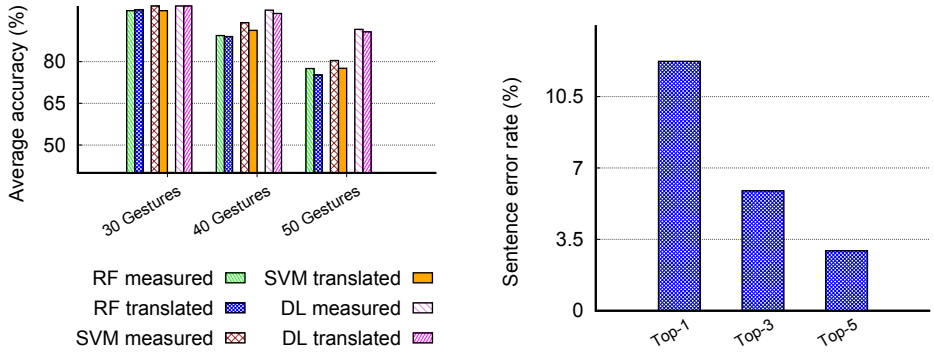
Fig. 20. Practical use cases for Vi2IMU's translated data.

(2) Multi-modal ASL recognition with synthesized IMU data. ASL is comprised of signs captured through hand and body movements, facial expressions, and mouthing. To accommodate for such complexity, multi-modal systems comprising multiple sensors like IMU, EMGs, cameras, etc. have been studied [58–60]. Such multi-modal systems can address the limitations of single-sensor solutions as each sensor captures a different representation of the same gesture and can compensate for the shortcoming of another. Here, we demonstrate one such multi-modal system using cameras and IMUs. Such a system can be used in a home environment where an ASL speaker wearing a smartwatch is interacting with a smart device or in an office environment where an ASL speaker is on a video conference while wearing a smartwatch. While existing works have studied standalone solutions with cameras [3, 4] and IMUs [1], they could complement each other to overcome their limitations. For example, cameras exhibit motion blurs and self-occlusions (like the left hand occluding the right) which result in missing joint information in pose estimates (as discussed in Section. 3 and also shown by existing works [12, 61]). IMUs, while capable of capturing fine-grained arm and hand movement, being local to the joint, do not capture complete body movements, facial expressions, and mouthing. A multi-modal system comprising cameras and IMUs can complement each other and enable robust ASL recognition.

We now consider a multi-modal camera and IMU ASL recognition system and demonstrate how our synthetic IMU data can augment camera data. For camera data, we use the upper body pose comprised of head, shoulder, elbow, and wrist joints from both hands similar to existing works [62, 63] for training and testing, and for the IMUs, we use the translated IMU data for training and the measured IMU data for testing. We train two models, one with only pose data (Pose) and the other with both pose and the translated IMU data (Pose + IMU). Fig. 20b shows the difference in accuracy between Pose + IMU and Pose models for different numbers of gestures. We find that the model trained with both pose data and IMU data can scale for a large number of gestures with an accuracy of 98.85% and 90.5% for 60 and 70 gestures, respectively. The average difference in accuracy between the two models is 7.24%. As explained in Section 3, due to varying factors like poor lighting, camera frame rate, etc. there is a considerable motion blur in the captured videos. IMUs can compensate for this motion blur and thus

offer better performance than only camera-based solutions. While we have demonstrated this with cameras, the translated data can also aid rapid prototyping of other multi-modal solutions such as RF-IMU, LIDAR-IMU, etc.

5.4.10 Feature engineering with translated data. Deep learning models learn task-specific feature representations, while non-deep learning models like random forests (RF), and support vector machines (SVM) can benefit from explicit feature engineering. Here, we study the difference between features obtained through feature engineering and deep learning for the translated and measured data. For this, we use high-level features used in existing works [34, 37] like kurtosis, skew, mean, absolute area, FFT coefficients, etc. We extract 29 features and train two machine learning models: random forests (RF) and support vector machines (SVM) with the extracted features and compare them with a deep learning model trained using the same data. We train models separately for the measured data and translated data (with augmentation).



(a) Feature engineering with translated data (b) Sentence-level ASL recognition

Fig. 21. Performance for (a) different machine learning models: Random forests (RF), Support vector machines (SVM), and Deep learning (DL), and (b) Sentence-level ASL recognition using Top-K accuracy and a language model.

Fig. 21a shows the results for the non-deep learning and deep learning models trained with features from the translated and measured data. The average difference in accuracy between the deep learning and non-deep-learning models for 30, 40, and 50 gesture classes is 1.67%, 6.05%, and 13.1%, respectively. This difference can be attributed to the well-understood shortcomings of hand-engineering features used in non-deep learning models compared to the features learned in the deep learning models. We also find that the average difference in accuracy for the models trained with the measured and translated data, for the non-deep learning models and deep learning model is 1.6% and 0.7% respectively. This shows that the application of our translated data is not necessarily limited to the deep learning models and it can achieve a comparable (to real, measured data) performance even using non-deep learning models.

5.4.11 Sentence-level ASL recognition. We find from Table 4 that the Top-3 and Top-5 accuracies for the translated data are reasonably high. Here, we answer the following question: can the Top-3 and Top-5 accuracies translate into better performance in sentence-level ASL recognition? For this, we create thirty sentences such as "time to work", "I have no college", etc., by combining the seventy words in our vocabulary. Next, we pick samples corresponding to these words and combine the predictions for Top-k accuracy as follows: for Top-1, if all the words were predicted correctly, we count it as a correct prediction. For Top-3 and Top-5, we first create sentences by considering all possible combinations of the words in the predictions. Next, we use a language model based on [64] to compute the probabilities of each created sentence and pick the sentence with the maximum probability as the prediction. The language model [64] provides the probability for different English sentences computed based on their frequency of occurrence in the Wikipedia text. If the sentence returned by the language model

matches the desired sentence, we treat it as a correct prediction. For example, if the top-3 predictions for two words are "black, pencil, and buy", and "finish, sunday, and fish" respectively, the language model returns the most frequently used combination of the nine possible combinations, which in this case is "buy fish". This way, a language model by incorporating conditional probabilities of word co-occurrences can compensate for the shortcomings of the word classification model. We use sentence error rate (SER) as the metric which is defined as the ratio between incorrect predictions to the total number of samples. Fig. 21b shows the results. We find that the Top-3 and Top-5 predictions when combined with the language model have lower SER compared to Top-1 predictions. The difference is 6.35% for Top-3 and 9.29% for Top-5. Thus, by incorporating language models, we can take advantage of high Top-3 and Top-5 accuracies for sentence-level ASL recognition.

6 RELATED WORK

Sensing using IMUs. Because of their general availability on smartphones and wearables, IMUs have been used in a variety of sensing applications like human activity recognition [65–68], gesture recognition [69, 70], sports analytics [71, 72], etc. They have also been used along with other mobile sensors (like microphones, GPS, etc.) for health applications [73, 74], user experience tracking [75], augmented reality [76], and more. In [1, 2], authors use IMUs in the smartwatch to perform both word and sentence level American sign language recognition. In [33], authors use wrist and finger worn IMUs to classify 37 gestures involving hand, finger and arm movements. In the above mentioned works, the authors establish the feasibility of using IMU sensors for a particular task by collecting data and evaluating the performance. Our work shows that the development of such schemes can considerably benefit from our proposed translation framework to create synthetic IMU data for training without laborious data collection. IMUs have also been used in arm tracking [77] and localization using dead reckoning [78–80]. Both arm tracking and dead reckoning solve the inverse problem where displacement is calculated from acceleration. On the other hand, we use displacement to calculate the acceleration while combining it with orientation.

Computer vision. Our work utilizes 2D and 3D pose estimation models developed in the field of computer vision. 2D and 3D pose estimation has been extensively studied with multiple proposed approaches [81–87]. In [88], authors propose a novel convolutional neural network (CNN) architecture that utilizes subsequent steps of pooling (downsampling) and upsampling that has shown to outperform existing approaches. In [23] authors propose a bottom-up approach where they first identify different body parts (using predictors) and use bipartite matching to estimate poses for multiple persons from a single image. In [89] authors decompose 3D pose estimation as a problem of 2D pose estimation and match the depth from a library of 3D poses to estimate the final pose. In [90] authors estimate 3D pose by fusing the information from videos and IMU sensors. We note that while our work builds on existing pose estimation frameworks, the problems of wrist orientation estimation and acceleration calculation directly from videos require addressing many challenges that are previously unaddressed.

Generation, transfer, and similar approaches. Multiple generative methods [14, 15, 91] have been proposed for generating IMU data. In [18] authors propose a deep learning based regression model to generate IMU data from 2D poses for human activity recognition. Directly regressing IMU data can lead to generalization issues and cannot provide a clear understanding of the model's shortcomings. In contrast, our modular approach enables the study of individual modules and any improvement of the underlying modules (studied separately) will also improve the entire framework. Other generative approaches have also been proposed to generate RF data [92, 93] directly from videos, simulate IMU data from motion captures [94, 95]. Unlike RF where data is dependent on the environment, IMU's data are local to the body joint and require careful modeling of displacement and wrist orientation. In contrast to generative approaches, trajectory based methods [13, 16, 19] compute IMU's acceleration by tracking the joint positions in the video. In [12, 17] authors propose a pipeline for IMU data translation for human activity recognition (HAR). Our focus is not on large, multi-joint movements (like human

activities) but is on fine-grained hand gestures which require attention to the impact of hand joints in orientation and acceleration/gyro estimation as discussed earlier. While authors in [11] have presented a pipeline for finger-worn IMU data synthesis from videos for gesture recognition, they do not consider videos of varying diversity in terms of resolution, motion blur, lighting condition, etc. Also, the proposed approach can only reconstruct one orientation axis. Other approaches such as projecting IMUs and videos to a common projected semantic space[96] for zero-shot activity recognition and cross-modal domain adaptation [97] for Doppler based activity recognition have also been proposed. In contrast to the proposed techniques which are designed for human activity recognition, our objective is to reconstruct realistic IMU data that can be used for any task in addition to gesture recognition.

7 DISCUSSION

We now discuss the various aspsects of Vi2IMU that can be improved through further investigation:

Better pose estimation approaches. OpenPose [23] is known to perform poorly in presence of motion blur [12, 61] which leads to missing pose estimates for different joints. While our orientation estimation module tackles this challenge, improvements in 2D pose estimation approaches can lead to better performance of all modules of Vi2IMU. Also, since we depend on existing 2D to 3D pose estimation approaches, improvements in 3D pose estimation can also improve the performance of Vi2IMU. Errors in 3D pose estimation often translate into inaccurate acceleration values. Accurate pose estimation can also enable Vi2IMU to better model the relation between the 3D displacement values and the acceleration values.

Improving orientation and acceleration/gyro estimation. The estimated orientation values affect the performance of the acceleration/gyro estimation module. Additionally, the absence of 3D hand joint positions impacts both the orientation and acceleration/gyro estimation modules as the models need to learn the relation between 2D and 3D inputs and the predicted output. As the arm and hand joints are presented in two different metrics, one in pixels and one in meters, the orientation and acceleration/gyro estimation modules have to either convert inputs in one metric to another or learn to project them to a common representational space where they can be modeled together. Accurately estimating 3D hand joint positions and incorporating them are open areas of investigation and any advancements in solving the problems can improve the performance of both orientation and acceleration/gyro estimation modules.

Improvement for specific gesture types. While Vi2IMU offers good performance for different types of gestures, gestures that predominantly depend on accurate depth estimates pose a significant challenge. For example, in ASL words like “father” and “mother”, the only observed displacement is along the Z-axis of the camera. However, as the observed displacements in the videos are relatively low, they translate into lower depth estimates and low acceleration values. We also note that in our current training data, the number of samples with lower displacement values is relatively less compared to higher displacement samples. This is further evident from the improvements we gain through our proposed augmentation technique. We believe that by incorporating better depth estimation approaches and better representation of short displacement samples in our training data, performance on gestures that heavily depend on accurate depth estimates could be improved.

Curating in-the-wild videos. While Vi2IMU benefits from a large number of in-the-wild videos, such videos often pose challenges in terms of their curation before incorporating them into our pipeline. For example, there exists a varying number of videos for different ASL signs on the web which could result in a data imbalance problem. This requires exerting some effort to compensate for this by going over YouTube videos, segmenting the videos into gesture instances, and labeling them with correct labels. In the future, such efforts can be avoided by automating the pipeline for curating videos (specifically, for platforms like YouTube) using web scrapping, segmentation, annotation, and labeling.

Increasing the translated data. We note that our proposed data augmentation could be extended with existing data augmentation techniques which can directly translate into performance gains. While we have utilized a large public dataset for translation, the number of translated instances can also be increased by adding more videos from YouTube to the pool of available videos. As the deep learning models are known to benefit from more data, increasing the number of translated instances will yield better performance.

Impact of user activity. The translated data is robust to different user postures like standing and sitting and involuntary limb movements. However, there are situations where a user might want to interact with an ASL recognition system, like a home assistant, while being in motion such as walking. In such cases, the resultant IMU data will contain signal representations for both walking and gestures. One way to address this challenge would be to take advantage of the fact that gestures and walking are performed at different frequencies, and we could do a frequency domain analysis to separate the signal components of the gesture from other user activity. Exploring the impact of such signal processing on the performance of the translated data requires further inquiry. Additionally, we could also study methods to incorporate such additional activity-related information into the translated data. For example, we have currently proposed an augmentation technique that converts the translated data to incorporate the amplitude characteristics from the measured IMU data. Similarly, approaches could be devised to incorporate additional frequency information like that of walking into the translated data.

Impact of non-manual markers. In ASL, non-manual markers such as torso shifts and head movements can be interleaved with manual markers like hand and arm gestures. When this happens, the measured IMU data will have signal components of both hand and body movements. In its current version, Vi2IMU does not take into account the acceleration due to body movements. For this, we can track other joints that are involved in non-manual markers like head and torso, and model their contribution to the measured acceleration and gyro values in the smartwatch. We can next integrate the contributions from different joints to estimate the composite acceleration and gyro values as observed by IMUs. Incorporating the impact of non-manual markers can make Vi2IMU more practically useful for expressive ASL recognition.

Sentence level ASL data synthesis. While word-level ASL data synthesis can enable various applications in HCI, continuous sentence-level synthesis is needed for applications like ASL-to-text transcribing, ASL-to-speech synthesis, and many more. There are existing ASL sentence datasets that we could use for sentence-level ASL data synthesis. In addition, public video platforms such as YouTube have several videos of ASL interpreters translating a speech or hearing along with the corresponding annotations. Given that sentence-level ASL recognition in itself is an active area of research, studying and extending Vi2IMU for sentence-level IMU data synthesis is a challenging and important research direction.

Extending Vi2IMU beyond ASL. Although Vi2IMU is proposed with a focus on in-the-wild ASL translation, the proposed framework can be directly adapted for synthesizing other hand gesture data. This could be used for applications like continuous arm and hand tracking for virtual reality, telerehabilitation, human-computer interaction (HCI), and many more. In contrast to gesture recognition, arm and hand joint tracking require integration over the synthesized acceleration and gyro data which is challenging as it leads to error accumulation over time. Accurate synthesis of IMU data is necessary to reduce the error accumulation over time. Extensions to the framework that account for such application-specific challenges could lead to better data synthesis. For example, during the acceleration and gyro synthesis, including loss values to indicate the efficacy of the synthesized data in arm and hand tracking could lead to better synthesis.

The proposed framework can be extended to applications that involve multi-body sensors like human activity recognition and human pose reconstruction. While existing works [11, 16] have shown the benefits of synthetic data in these problems, incorporating fine-grained synthetic IMU data that accounts for the impact of hand joints could complement the existing approaches and improve their performance. An important question to answer in these explorations is the difference between the measured IMU data for gestures and activities. A deep

understanding of this difference can inform changes to the model that account for the difference in movements between human activities and gestures. Specifically, as activities could vary in intensity and speed compared to the gestures, methods to incorporate this information in acceleration/gyro estimation models are needed. Additionally, with multi-body sensors, one body part movement can impact the IMU values observed in sensors at other body parts. Incorporating these indirect impacts into acceleration/gyro estimation is needed for an accurate IMU data synthesis.

8 CONCLUSIONS

Our proposed system Vi2IMU attempts to solve a challenging problem of video to IMU translation for ASL gestures. Vi2IMU is built on the insight that orientation and acceleration estimations are dependent on hand joint positions and it is necessary to carefully account them to produce realistic, synthetic IMU data from ASL videos. Our model is shown to be robust to missing hand joints due to the development of our key-delta frame based learning models. Using an in-the-wild ASL dataset and our own collected data, we trained and evaluated the models to show that our translation can be accurate and useful, especially in the ASL recognition problem.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments and feedback. This research is supported by NSF grants CNS-2045885 and CNS-1730083.

REFERENCES

[1] J. Hou, X.-Y. Li, P. Zhu, Z. Wang, Y. Wang, J. Qian, and P. Yang, "Signspeaker: A real-time, high-precision smartwatch-based sign language translator," in *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[2] Q. Zhang, J. Jing, D. Wang, and R. Zhao, "Wearsign: Pushing the limit of sign language translation using inertial and emg wearables," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, mar 2022.

[3] S.-K. Ko, J. G. Son, and H. Jung, "Sign language recognition with recurrent neural network using human keypoint detection," in *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, RACS '18, (New York, NY, USA), pp. 326–328, ACM, 2018.

[4] C. C. de Amorim, D. Macêdo, and C. Zanchettin, "Spatial-temporal graph convolutional networks for sign language recognition," *CoRR*, vol. abs/1901.11164, 2019.

[5] T. Yuan, S. Sah, T. Ananthanarayana, C. Zhang, A. Bhat, S. Gandhi, and R. Ptucha, "Large scale sign language interpretation," in *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pp. 1–5, May 2019.

[6] B. Fang, J. Co, and M. Zhang, "Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, SenSys '17, (New York, NY, USA), pp. 5:1–5:13, ACM, 2017.

[7] H. R. V. Joze and O. Koller, "MS-ASL: A large-scale data set and benchmark for understanding american sign language," *CoRR*, vol. abs/1812.01053, 2018.

[8] D. Li, C. R. Opazo, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1448–1458, 2020.

[9] C. Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 44–52, June 2015.

[10] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3d convolutional neural networks," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, June 2015.

[11] Y. Liu, S. Zhang, and M. Gowda, "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*, IoTDI '21, (New York, NY, USA), p. 182–194, Association for Computing Machinery, 2021.

[12] H. Kwon, B. Wang, G. D. Abowd, and T. Plötz, "Approaching the real-world: Supporting activity recognition training with virtual imu data," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, sep 2021.

[13] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li, "A deep learning method for complex human activity recognition using virtual wearable sensors," in *Spatial Data and Intelligence* (X. Meng, X. Xie, Y. Yue, and Z. Ding, eds.), (Cham), pp. 261–270, Springer International Publishing, 2021.

[14] V. F. Rey, P. Hevesi, O. Kovalenko, and P. Lukowicz, "Let there be imu data: Generating training data for wearable, motion sensor based activity recognition from monocular rgb videos," UbiComp/ISWC '19 Adjunct, (New York, NY, USA), p. 699–708, Association for Computing Machinery, 2019.
[15] S. Zhang and N. Alshurafa, "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, (New York, NY, USA), p. 223–227, Association for Computing Machinery, 2020.
[16] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll, "Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time," *ACM Trans. Graph.*, vol. 37, dec 2018.
[17] H. Kwon, C. Tong, H. Haresamudram, Y. Gao, G. D. Abowd, N. D. Lane, and T. Plötz, "Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, Sept. 2020.
[18] V. Fortes Rey, K. K. Garewal, and P. Lukowicz, "Translating videos into synthetic training data for wearable sensor-based activity recognition systems using residual deep convolutional networks," *Applied Sciences*, vol. 11, no. 7, 2021.
[19] S. Takeda, T. Okita, P. Lago, and S. Inoue, "A multi-sensor setting activity recognition simulation tool," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, UbiComp '18, (New York, NY, USA), p. 1444–1448, Association for Computing Machinery, 2018.
[20] "Asl mom." https://www.signingsavvy.com/sign/mom.
[21] "Asl color." https://www.signingsavvy.com/sign/COLOR/1136/1.
[22] "Asl slow." https://www.signingsavvy.com/.
[23] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," 2019.
[24] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.
[25] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.
[26] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
[27] N. K. Caselli, Z. S. Sehyr, A. M. Cohen-Goldberg, and K. Emmorey, "Asl-lex: A lexical database of american sign language," *Behavior Research Methods*, vol. 49, no. 2, pp. 784–801, 2017.
[28] "Asl brother." https://www.signingsavvy.com/sign/BROTHER/57/1.
[29] A. F. Agarap, "Deep learning using rectified linear units (relu)," 2019.
[30] S. Takeda, T. Okita, P. Lago, and S. Inoue, "A multi-sensor setting activity recognition simulation tool," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, UbiComp '18, (New York, NY, USA), p. 1444–1448, Association for Computing Machinery, 2018.
[31] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li, "A deep learning method for complex human activity recognition using virtual wearable sensors," *CoRR*, vol. abs/2003.01874, 2020.
[32] https://www.signingsavvy.com/sign/NO/291/1.
[33] C. Xu, P. H. Pathak, and P. Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, HotMobile '15, (New York, NY, USA), p. 9–14, Association for Computing Machinery, 2015.
[34] J. Gummeson, B. Priyantha, and J. Liu, "An energy harvesting wearable ring platform for gestureinput on surfaces," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, (New York, NY, USA), p. 162–175, Association for Computing Machinery, 2014.
[35] E. Simonetti, E. Bergamini, G. Vannozzi, J. Bascou, and H. Pillet, "Estimation of 3d body center of mass acceleration and instantaneous velocity from a wearable inertial sensor network in transfemoral amputee gait: A case study," *Sensors*, vol. 21, no. 9, 2021.
[36] W. Zijlstra, R. W. Bisseling, S. Schlumbohm, and H. Baldus, "A body-fixed-sensor-based analysis of power during sit-to-stand movements," *Gait Posture*, vol. 31, no. 2, pp. 272–278, 2010.
[37] E. Munguia Tapia, *Using machine learning for real-time activity recognition and estimation of energy expenditure*. PhD thesis, Massachusetts Institute of Technology, 2008.
[38] Y. Zhang and Q. Yang, "A survey on multi-task learning," *CoRR*, vol. abs/1707.08114, 2017.
[39] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017.
[40] "Asl finish." https://www.signingsavvy.com/sign/FINISH/149/1.
[41] "Azure kinect dk." https://azure.microsoft.com/en-us/services/kinect-dk/.
[42] "Google pixel 41." https://store.google.com/us/product/pixel_4a.
[43] "Azure body trakcing api." https://docs.microsoft.com/en-us/azure/kinect-dk/get-body-tracking-results.
[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.
[46] R. Caruana, S. Lawrence, and L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," *Advances in neural information processing systems*, pp. 402–408, 2001.
[47] "Ridge regression." https://en.wikipedia.org/wiki/Ridge_regression.
[48] "L2 norm." https://mathworld.wolfram.com/L2-Norm.html.
[49] "Asl hurt." https://www.signingsavvy.com/search/hurt.
[50] "Asl cheese." https://www.signingsavvy.com/search/cheese.
[51] "Accuracy metrics." https://en.wikipedia.org/wiki/Accuracy_and_precision.
[52] "vi2imu dataset." https://github.com/sensys2022/Vi2IMU.
[53] "Asl sister." https://www.signingsavvy.com/sign/SISTER/392/1.
[54] "Rotation matrix." https://www.handspeak.com/word/search/index.php?id=1992.
[55] "Rotation matrix." https://www.lingvano.com/asl/blog/please-in-sign-language/.
[56] N. Patricia and B. Caputo, "Learning to learn, from transfer learning to domain adaptation: A unifying perspective," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
[57] W. M. Kouw, "An introduction to domain adaptation and transfer learning," *CoRR*, vol. abs/1812.11806, 2018.
[58] J. Wu, L. Sun, and R. Jafari, "A wearable system for recognizing american sign language in real-time using imu and surface emg sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 5, pp. 1281–1290, 2016.
[59] J. Zhang, Q. Wang, Q. Wang, and Z. Zheng, "Multimodal fusion framework based on statistical attention and contrastive attention for sign language recognition," *IEEE Transactions on Mobile Computing*, pp. 1–13, 2023.
[60] M. Jebali, A. Dakhli, and M. Jemni, "Vision-based continuous sign language recognition using multimodal sensor fusion," *Evolving Systems*, vol. 12, no. 4, pp. 1031–1044, 2021.
[61] A. Duarte, S. Palaskar, L. Ventura, D. Ghadiyaram, K. DeHaan, F. Metze, J. Torres, and X. Giro-i Nieto, "How2sign: A large-scale multimodal dataset for continuous american sign language," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2735–2744, June 2021.
[62] D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1459–1469, 2020.
[63] M. Boháček and M. Hrúz, "Sign pose-based transformer for word-level sign language recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 182–191, 2022.
[64] "Wikipedia language model." https://nlp.cs.nyu.edu/wikipedia-data/.
[65] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, June 2017.
[66] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Human activity recognition with inertial sensors using a deep learning approach," in *2016 IEEE SENSORS*, pp. 1–3, 2016.
[67] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, Jan. 2014.
[68] H. Kwon, G. D. Abowd, and T. Plötz, "Complex deep neural networks from large scale virtual imu data for effective human activity recognition using wearables," *Sensors*, vol. 21, no. 24, 2021.
[69] N. Siddiqui and R. H. Chan, "Multimodal hand gesture recognition using single imu and acoustic measurements at wrist," *PloS one*, vol. 15, no. 1, p. e0227039, 2020.
[70] Y. Liu, F. Jiang, and M. Gowda, "Finger gesture tracking for interactive applications: A pilot study with sign languages," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, Sept. 2020.
[71] M. Gowda, A. Dhekne, S. Shen, R. R. Choudhury, L. Yang, S. Golwalkar, and A. Essanian, "Bringing iot to sports analytics," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, (Boston, MA), pp. 499–513, USENIX Association, Mar. 2017.
[72] M. Gowda, A. Dhekne, S. Shen, R. R. Choudhury, S. X. Yang, L. Yang, S. Golwalkar, and A. Essanian, "Iot platform for sports analytics," *GetMobile: Mobile Comp. and Comm.*, vol. 21, p. 8–14, Feb. 2018.
[73] V. Chandel, A. Sinharay, N. Ahmed, and A. Ghose, "Exploiting imu sensors for iot enabled health monitoring," IoT of Health '16, (New York, NY, USA), p. 21–22, Association for Computing Machinery, 2016.
[74] T. Ahmed, M. Y. Ahmed, M. M. Rahman, E. Nemati, B. Islam, K. Vatanparvar, V. Nathan, D. McCaffrey, J. Kuang, and J. A. Gao, "Automated time synchronization of cough events from multimodal sensors in mobile devices," in *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, (New York, NY, USA), p. 614–619, Association for Computing Machinery, 2020.
[75] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Trans. Sen. Netw.*, vol. 6, Jan. 2010.
[76] J. D. Hincapié-Ramos, K. Ozacar, P. P. Irani, and Y. Kitamura, "Gyrowand: Imu-based raycasting for augmented reality head-mounted displays," in *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, (New York, NY, USA), p. 89–98, Association for

Computing Machinery, 2015.
[77] S. Shen, M. Gowda, and R. Roy Choudhury, "Closing the gaps in inertial motion tracking," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, (New York, NY, USA), p. 429–444, Association for Computing Machinery, 2018.
[78] W. Kang and Y. Han, "Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2906–2916, 2015.
[79] S. Shen, M. Gowda, and R. Roy Choudhury, "Closing the gaps in inertial motion tracking," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, (New York, NY, USA), p. 429–444, Association for Computing Machinery, 2018.
[80] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, (New York, NY, USA), p. 197–210, Association for Computing Machinery, 2012.
[81] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
[82] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[83] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
[84] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network," in *Computer Vision – ACCV 2014* (D. Cremers, I. Reid, H. Saito, and M.-H. Yang, eds.), (Cham), pp. 332–347, Springer International Publishing, 2015.
[85] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3d hand pose estimation from monocular video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1793–1805, 2011.
[86] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," in *2018 International Conference on 3D Vision (3DV)*, pp. 120–130, 2018.
[87] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2006.
[88] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 483–499, Springer International Publishing, 2016.
[89] C.-H. Chen and D. Ramanan, "3d human pose estimation = 2d pose estimation + matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[90] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. P. Collomosse, "Total capture: 3d human pose estimation fusing video and inertial sensors.," in *BMVC*, vol. 2, pp. 1–13, 2017.
[91] O. Banos, A. Calatroni, M. Damas, H. Pomares, I. Rojas, H. Sagha, J. del R. Mill´n, G. Troster, R. Chavarriaga, and D. Roggen, "Kinect=imu? learning mimo signal mappings to automatically translate activity recognition systems across sensor modalities," in *2012 16th International Symposium on Wearable Computers*, pp. 92–99, 2012.
[92] H. Cai, B. Korany, C. R. Karanam, and Y. Mostofi, "Teaching rf to sense without rf training measurements," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, Dec. 2020.
[93] K. Ahuja, Y. Jiang, M. Goel, and C. Harrison, "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*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.
[94] A. D. Young, M. J. Ling, and D. K. Arvind, "Imusim: A simulation environment for inertial sensing algorithm design and evaluation," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 199–210, 2011.
[95] Y. Hao, B. Wang, and R. Zheng, "Cromosim: A deep learning-based cross-modality inertial measurement simulator," 2022.
[96] C. Tong, J. Ge, and N. D. Lane, "Zero-shot learning for imu-based activity recognition using video embeddings," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, dec 2022.
[97] S. Bhalla, M. Goel, and R. Khurana, "Imu2doppler: Cross-modal domain adaptation for doppler-based activity recognition using imu data," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, dec 2022.

APPENDIX A

As part of Methodological Transparency & Reproducibility Appendix (META), we have made our translated IMU data (plots and raw acceleration and gyro data) from MSASL videos for 70 gestures along with corresponding measured IMU data available anonymously at https://github.com/vi2imu/Vi2IMU.