[THIS IS A PREPRINT] The manuscript has been accepted at IROS23

Anytime, Anywhere: Human Arm Pose from Smartwatch Data for Ubiquitous Robot Control and Teleoperation

Fabian C Weigend, Shubham Sonawani, Michael Drolet and Heni Ben Amor

Abstract— **This work devises an optimized machine learning approach for human arm pose estimation from a single smartwatch. Our approach results in a distribution of possible wrist and elbow positions, which allows for a measure of uncertainty and the detection of multiple possible arm posture solutions, i.e., multimodal pose distributions. Combining estimated arm postures with speech recognition, we turn the smartwatch into a ubiquitous, low-cost and versatile robot control interface. We demonstrate in two use-cases that this intuitive control interface enables users to swiftly intervene in robot behavior, to temporarily adjust their goal, or to train completely new control policies by imitation. Extensive experiments show that the approach results in a 40% reduction in prediction error over the current state-of-the-art and achieves a mean error of 2.56 cm for wrist and elbow positions.**

I. INTRODUCTION

The relationship between humans and robots is a central question of artificial intelligence and robotics. As robots become increasingly capable, there is growing interest for human-robot collaboration in various domains, such as healthcare, manufacturing, and daily activities. Many scenarios in these fields envision humans to teleoperate, assist, or teach a robot counterpart. For example, a human expert may demonstrate to a robot how to perform a new task or how to manipulate a new object. Such scenarios, however, require intuitive and robust interfaces for capturing human body motion.

To date, motion capture cameras are the gold standard in capturing human motion [1]–[3]. A setup of multiple cameras can provide a high-fidelity recording of body postures and positions over time. However, motion capture requires an expensive and stationary setup. Easier consumer-grade hardware, e.g., Microsoft Kinect, provides only low-fidelity approximations of the body posture and is heavily affected by line-of-sight issues and a limited field-of-view [3]. Alternative motion capture approaches are based on Inertial Measurement Units (IMU) and allow tracking without line-of-sight issues. However, they typically require wearing two or more IMUs on different limbs, e.g., strapped around lower arm and upper arm or as a special suit [4]–[6]. Even though research has investigated human arm posture estimations from a single IMU, the authors in [7], [8] reported prediction accuracy is of low fidelity.

In this paper, we devise a machine learning approach to increase the accuracy for predicting human arm poses from the single IMU of a smartwatch. As observable in the top in Figure 1, our approach results in a distribution of predicted postures, which allows to estimate a measure of uncertainty and provides a range of possible solutions to pick from.

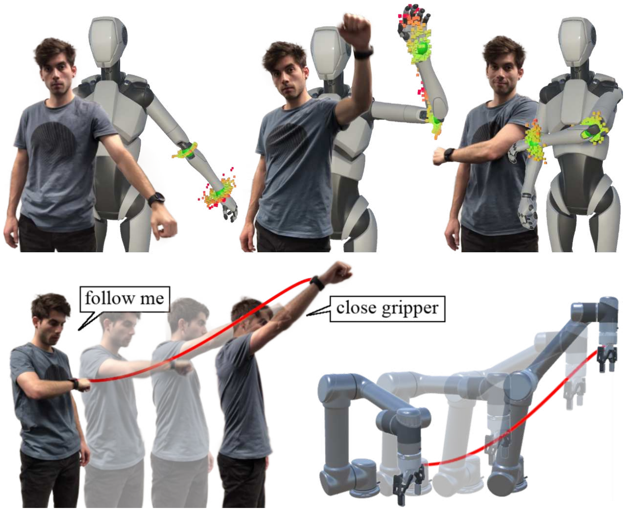

Fig. 1. **Top:** The avatar shows predicted elbow and wrist positions from smartwatch sensor data. Our approach results in a distribution of solutions. The mean of a distribution is depicted as a green sphere All individual predictions of a distribution are depicted as small cubes, colored according to their proximity to the mean. **Bottom:** We also stream microphone data to utilize speech recognition. This combination offers a versatile interface to interact with and to control robots anytime and anywhere.

By combining the increased accuracy of our approach with speech recognition, we turn the smartwatch into an ubiquitous robot control interface. Smartwatches are widely recognized as common consumer-grade devices that users are already familiar with [9]. Without the need for a complicated setup, a human expert can engage with the robot at any time and anywhere. As depicted in Figure 1, they may move the robot to a new target and issue commands via speech recognition. We summarize our contributions as follows:

- We present a machine learning approach for real-time estimation of upper and lower arm postures from a single smartwatch.
- Our approach results in a distribution of possible arm postures, which opens up opportunities for selecting optimal solutions.
- We identify solutions to calibration, data representation, and network design that yield higher accuracy than previously reported results in the literature.
- We combine human arm posture estimations with speech recognition and present two real-robot exam-

All Authors are with the School of Computing and Augmented Intelligence, Arizona State University {fweigend, sdsonawa, mdrolet, hbenamor}@asu.edu

ples that highlight the advantages of our smartwatch approach for robotics.

II. RELATED WORK

Tracking one or multiple parts of the human body is an essential step in approaches to robot control. For example, techniques for teleoperation build upon the accurate detection of human body pose [10]. In a similar vein, imitation learning [11] or programming-by-demonstration (PbD) [12] requires a human expert to provide one or more demonstrations of target motions. These are distilled into a policy that generalizes the observed behavior to new situations. Traditionally, a large number of works for PbD have relied on costly motion capture setups for recording high-fidelity data [13]–[15]. Other approaches try to strike a balance between the cost of data collection and the fidelity by leveraging Inertial Measurement Units (IMUs) or camera-based setups. For example, the works in [6] use multiple IMUs attached to different parts of the body to transfer human motions onto a robot. However, approaches based on multiple IMUs require a careful placement of sensors on the human body along with a (potentially time-consuming) calibration process.

More recently, consumer-grade hardware for virtual and augmented reality (VR/AR) is becoming an alternative for motion tracking in robotics [16]–[18]. For example, the work in [16] uses a HTC Vive VR system for robot teleoperation in a manipulation task. HTC Vive controller estimate their positions from infrared signals from so-called base stations, which have to be carefully placed and calibrated. In a similar vein, the work in [19] uses an Oculus Quest device for upper body tracking. However, Oculus controllers are tracked via cameras within the VR headset [20]. Headsets can cause ergonomic discomfort and reduce the situational awareness.

On the contrary, wearable devices like a smartwatch can only provide comparably low-fidelity position data. Instead, they offer a combination of low-cost, ease-of-use and a broad range of additional sensors, e.g., magnetometer, atmospheric pressure sensor, microphone or Photoplethysmography (PPG) sensor [9]. For example, these on-body sensors enable advances in emotion sensing [9]. In robot control, smartwatches are mostly used to control robots with roll, pitch, and yaw estimates from IMU and magnetometer [21]. Research has also investigated methods for human pose estimations from smartwatch data [7], [8], however, these are of low precision and have mostly been intended for recreational purposes or physical therapy [8]. To open up more opportunities to utilize the advantages of wearable devices in robot control, we propose a solution to improve real-time arm pose estimations in such settings.

III. METHODOLOGY

In this work, we address the problem of estimating human arm poses from a single smartwatch. We cast the process as a supervised learning task, in which postural information is predicted from a set of multimodal sensors. A challenging aspect is the inherent one-to-many mapping imposed by redundant human kinematics. Readings obtained from smartwatch sensors may not correspond to a single arm movement or position, but rather, can indicate various possibilities. Another challenge emerges from natural variability in the sensor data. Sensor readings for pressure and orientation need to be adjusted before usage. In the following section, we discuss how to train deep learning models that are particularly well-suited to the requirements of the task.

A. Data Collection

We collect motion capture data as ground truth prediction targets and match these with recorded smartwatch sensor measurements. To this end, we develop a Wear OS app to record and stream sensor measurements. The app is tested on a Samsung Galaxy Watch 5. It records data from a set of multimodal sensors. These include gyroscope measurements ($\boldsymbol{\phi}$) with $\boldsymbol{\phi} \in \mathbb{R}^3$, which represent the rotation angles with respect to the coordinate axes. Further, it records measurements of the gravity sensor ($\boldsymbol{\gamma}$) and linear acceleration sensor ($\boldsymbol{\alpha}$) with $\boldsymbol{\gamma}, \boldsymbol{\alpha} \in \mathbb{R}^3$, which represent the acceleration with respect to the X, Y and Z axis. Linear acceleration is the raw acceleration ($\boldsymbol{\alpha}_{\mathrm{raw}}$) minus the gravity measurements such that $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{\mathrm{raw}} - \boldsymbol{\gamma}$. In addition, the app records the virtual rotation vector sensor ($\boldsymbol{\theta}$), which is provided by Wear OS. The rotation vector sensor estimates the global smartwatch rotation from the magnetometer, accelerometer and gyroscope as a quaternion, thus, $\boldsymbol{\theta} \in \mathbb{R}^4$. Together with the reading from the atmospheric pressure sensor (ρ) with $\rho \in \mathbb{R}$, one observation s from the smartwatch consists of the following values $\boldsymbol{s} = [\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \rho]^\top$, with $\boldsymbol{s} \in \mathbb{R}^{14}$. In addition, the app also streams the microphone data, which we use for speech recognition. However, because we do not utilize microphone data for arm pose estimations, it is not included in $\boldsymbol{s}$.

As ground truth, we collect upper-body motion capture data. We use the research-grade optical motion capture system OptiTrack [2]. The motion capture environment features 12 cameras. We recorded data from 6 participants, who wore a 25-marker-upper-body suit along with the smartwatch on their left wrist (See Figure 2). We collect the hip rotation ($\mathbf{q}_{\mathrm{h}}$), lower arm rotation ($\mathbf{q}_{\mathrm{l}}$) and upper arm rotation ($\mathbf{q}_{\mathrm{u}}$) as quaternions. We further store the lower arm length (l_l) and upper arm length (l_u) of the participant to estimate wrist and elbow positions from recorded rotations. Therefore, a motion capture ground truth observation $\boldsymbol{g}$ contains $\boldsymbol{g} = [\mathbf{q}_{\mathrm{h}}, \mathbf{q}_{\mathrm{l}}, \mathbf{q}_{\mathrm{u}}, l_l, l_u]^\top$, with $\boldsymbol{g} \in \mathbb{R}^{14}$.

Once the motion capture system and our smartwatch app started recording, participants were instructed to keep their chest and hip stationary while moving their left arm in any possible way. The smartwatch recorded at around $\sim 50\,\mathrm{Hz}$ which resulted in a set of 381 535 observations. The motion capture system recorded at $\sim 120\,\mathrm{Hz}$ which resulted in a set of 926 164 motion capture observations. Data collection was conducted in accordance with Arizona State University (ASU) guidelines. Written informed consent was obtained under and approved by the institutional review board (IRB) of ASU under the ID STUDY00017558.

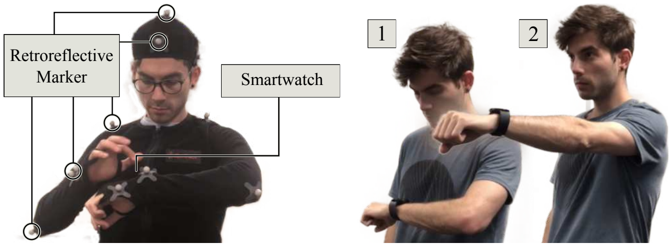

Fig. 2. **Left:** We collected ground truth data with an optical motion capture system and a 25-marker upper body suit. **Right:** Our two-step calibration process. First, the user holds the watch at chest height to estimate relative atmospheric pressure. Then, the user stretches the arm forward for an estimate of body orientation.

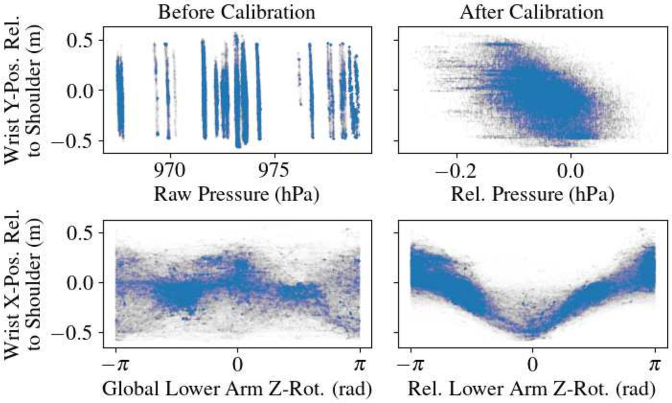

Fig. 3. This figure depicts two examples for data before and after calibration. Each plot contains all of our 381 535 data points.

B. Data Processing

Recorded smartwatch and motion capture data requires alignment and preprocessing since (a) motion capture data was recorded at a higher frequency, and (b) the data was collected in distinct coordinate systems. This subsection defines steps to merge smartwatch observations with our ground truth data. Additionally, we present a calibration procedure to further enhance correlations within the data and aid the training of predictive models for arm posture.

Merging data sets We retrospectively merged the set of collected smartwatch observations and the set of ground truth motion capture data by pairing observations according to their timestamps. Every smartwatch observation $\boldsymbol{s}$ was paired with the motion capture data observation $\boldsymbol{g}$ that was recorded closest in time.

Calibrating atmospheric pressure A critical smartwatch sensor provides measurements of the atmospheric pressure ρ. As depicted in the top left plot in Figure 3, these measurements suffer from day-by-day variations due to changing weather conditions and temperature. Data that was collected in the same experiment or on the same day are recognizable as vertical lines when plotted against the corresponding Y-position (elevation) of the wrist.

We propose a calibration procedure to remove the day-by-day variations and create a relative pressure measurement. It is depicted on the right in Figure 2. The user presses “calibrate” and holds the smartwatch at chest height. The watch records atmospheric pressure measurements for three seconds and then vibrates to signal that the step is completed. The average recorded pressure is saved as the atmospheric pressure at chest height (ρ_c) used to estimate relative atmospheric pressure (ρ^r) as $\rho^r = \rho - \rho_\mathrm{c}$.

The Kendall’s Tau correlation coefficient between the ρ and wrist Y-position in the top left plot of Figure 3 is -0.009. In contrast, the Kendall’s Tau correlation coefficient of ρ^r and the wrist Y-position is -0.308, confirming that there is a higher correlation between the variables. We asked our participants to perform this calibration step before data collection and replaced the ρ measurement in $\boldsymbol{s}$ with ρ^r.

Calibrating rotation Due to the kinematic structure underlying human anatomy, arm orientations are affected by the body orientation. However, no information about the body forward-facing direction is available from the smartwatch sensors. Although a universal solution is preferable, we introduce a constraint to overcome this hurdle: the forward-facing direction of the user must be constant and known. We explore future opportunities in this area in Section VI, but for now, we will highlight the advantages of this imposed constraint in our approach.

We incorporate the constraint of a constant body forward-facing direction with the second step of our proposed two-step calibration procedure. The step is depicted under number two in Figure 2. After completing the first step for the relative pressure measurement, the user stretches their arm forward. The watch records its rotation measurements for three seconds and saves their average as the calibration forward-facing direction ($\boldsymbol{\theta}_c$). This allows us to estimate the relative smartwatch rotation ($\boldsymbol{\theta}^r$) as the quaternion $\boldsymbol{\theta}^r = \boldsymbol{\theta}_c^{-1}\boldsymbol{\theta}$.

To transform our ground truth motion capture data into the same local coordinate system, we use the collected $\mathbf{q}_\mathrm{h}$ as the ground-truth forward-facing direction and estimate the relative lower arm rotation ($\mathbf{q}_\mathrm{l}^r$) and relative upper arm rotation ($\mathbf{q}_\mathrm{u}^r$) as $\mathbf{q}_\mathrm{l}^r = \mathbf{q}_\mathrm{h}^{-1}\mathbf{q}_\mathrm{l}$ and $\mathbf{q}_\mathrm{u}^r = \mathbf{q}_\mathrm{h}^{-1}\mathbf{q}_\mathrm{u}$. Together with the saved lower and upper arm lengths, this information also allows us estimate wrist and elbow positions from these orientations in the same local coordinate system and relative to the shoulder.

The example in the bottom plots of Figure 3 shows the benefit of using rotations relative to the forward-facing direction of the user. The rotation is denoted in Euler angles for easier interpretation. The body coordinate system in this example has the Z-axis tangential to the ground pointing forward and the X-axis along the right arm in T-pose. The Y-axis is orthogonal to the ground pointing upwards. As observable in the bottom-right plot of Figure 3, when the user extends their left arm wearing the smartwatch to the left, the lower arm Z-rotation from the T-pose is 0 and the distance from wrist X-position to shoulder X-position is around -0.5 m. In contrast, in the bottom-left plot, the global smartwatch rotation provides less information because users were not always facing the same direction during data collection. Thus, the relative rotation after our calibration allows to narrow down possible wrist positions from observed lower arm rotations.

C. Predictive Models

Building upon presented data merging and calibration steps, we devise an optimized predictive model that benefits from previously presented data preprocessing steps. To this goal, we investigate two distinct neural network architectures and four distinct representations of prediction targets. This allows us to compare and choose among a range of design choices which we present in the following.

Architectures and Inputs We train two neural network architectures on two similar sets of inputs. The first architecture is a feedforward network, which receives as inputs $[\rho^r, \boldsymbol{\theta}^r, \boldsymbol{\alpha}, \boldsymbol{\phi}, \boldsymbol{\gamma}, l_l, l_u]^\top$. The second architecture is an Long Short-Term Memory (LSTM) network which receives the same input data with two additions: The data is stacked into a sequence of length 6 and it receives the time delta from each sequence step to the next.

Prediction Targets By human arm pose estimation from smartwatch data, more specifically, we refer to predicting ground truth relative lower and upper arm rotation, i.e., $\mathbf{q}^r_\mathrm{l}$ and $\mathbf{q}^r_\mathrm{u}$, or predicting ground truth wrist and elbow positions which were estimated from these rotations. The naive way to predict wrist and elbow positions is to train a network to generate positions in Cartesian XYZ coordinates. However, since lower and upper arm lengths are constants, i.e. l_l and l_u, we know that positions lie on a manifold, which allows to narrow down the search space. The elbow position has to lie on a sphere around the shoulder with radius l_u. The wrist position has to lie on the manifold defined by spheres with a radius of l_l around all possible elbow positions [7]. Therefore, as an alternative, we train our network architectures to predict upper and lower arm rotations and estimate positions from using known l_l and l_u. Intuitively, polar coordinates come to mind as a suitable representation. When using l_u as the radius, the position of the elbow relative to the shoulder is well-described by two angles. Further, rotations can be represented in quaternions.

However, these representation spaces do not have a continuous mapping to their the rotation space, e.g., Euler angles jump from 359 to 0 degrees, which can cause complications during the training process due to discontinuity [22]. A 6-dimensional rotation representation (6DRR) has been proposed by [22], with which the authors achieved promising results for training neural networks on a human pose inverse kinematics test. In the case of the 3-dimensional (3D) rotation group SO(3) in the 3D Euclidean space, their 6DRR space consists of the first two columns ($\mathbf{a}_1$ and $\mathbf{a}_2$) of the 3D rotation matrix. A mapping g from rotation matrix to 6DRR is therefore:

$$g\Bigg(\underbrace{\begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix}}_{\text{Rotation Matrix}}\Bigg) = \underbrace{\begin{bmatrix} | & | \\ \mathbf{a}_1 & \mathbf{a}_2 \\ | & | \end{bmatrix}}_{\text{6DRR}}. \tag{1}$$

The neural network is then trained to predict these two columns. For a mapping f to recover the full 3D rotation matrix, [22] propose to normalize and orthogonalize the predicted two columns and estimate the last one with the cross product as:

$$\begin{aligned} f\left(\begin{bmatrix} | & | \\ \mathbf{a}_1 & \mathbf{a}_2 \\ | & | \end{bmatrix}\right) &= \begin{bmatrix} | & | & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \\ | & | & | \end{bmatrix} \\ &= \begin{bmatrix} | & | & | \\ N(\mathbf{a}_1) & O(\mathbf{a}_2, \mathbf{b}_1) & \mathbf{b}_1 \times \mathbf{b}_2 \\ | & | & | \end{bmatrix} \end{aligned} \tag{2}$$

where $N(\mathbf{a}) = \frac{\mathbf{a}}{||\mathbf{a}||}$ and $O(\mathbf{a}, \mathbf{b}) = N(\mathbf{a} - (\mathbf{b} \cdot \mathbf{a})\mathbf{b})$. Note the repeated use of $N(\mathbf{a}_1)$ as $\mathbf{b}_1$ here.

We investigate prediction accuracy for all discussed position and rotation representations: elbow and wrist positions in polar coordinates (Polar) and Cartesian coordinates (XYZ) a well as upper and lower arm rotations in 6DRR and quaternions (Quat).

Activation Function Also the choice for the activation function of a network has an effect on performance. Normalization of our IMU, pressure or arm length inputs is cumbersome because of likely outliers. For example, extreme movements, like hitting an obstacle, can cause large spikes in accelerometer data. Additionally, it is difficult to define a minimum or maximum arm length since possible values vary between body proportions, children and adults.

To mitigate the possible impact of out-of-distribution observations, we opted to employ the scaled exponential linear units (SELU) activation function by [23], which is reported to induce self-normalizing properties. It is estimated as

$$\mathrm{SELU}(x) = \lambda \begin{cases} x & \text{if} x > 0 \\ \alpha e^x - \alpha & \text{if} x \leq 0 \end{cases}, \tag{3}$$

where [23] derived α as 1.6733 and λ as 1.0507. As summarized by [23], these values enable necessary properties of the SELU activation to allow for self normalization by, firstly, having positive and negative values for controlling the mean. Secondly, by featuring regions where the slope approaches zero and regions where the slope is larger than one. These regions allow to dampen the variance if it is too large or to increase the variance if it is too low. With these properties, [23] showed that there are upper and lower bounds on the variance, thereby making learning robust even under the presence of noise and perturbations.

Other Hyperparameters For both architectures all layers consist of 128 neurons. The feedforward network features five layers and the LSTM architecture four LSTM layers. Both networks are trained for 200 epochs with the Adam [24] optimizer, a learning rate of 0.001 and a Mean Absolute Error (MAE) loss function. Early stopping is applied when the minimal loss does not improve for 10 epochs.

D. Multimodality and Prediction Uncertainty

Even after incorporating the constraint of known body direction, still, the same smartwatch sensor recordings may have multiple possible arm posture solutions. To address this issue, we integrate dropout layers into our network architecture and utilize them for generating multiple stochastic

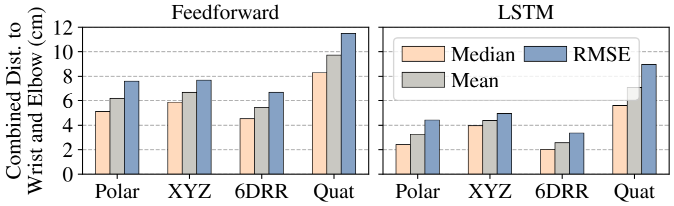

Fig. 4. A comparison of prediction accuracy for combined wrist and elbow positions on test data. Both network architectures are trained to predict wrist and elbow positions in polar coordinates or Cartesian coordinates (XYZ) as well as upper and lower arm rotations as quaternions or 6DRR.

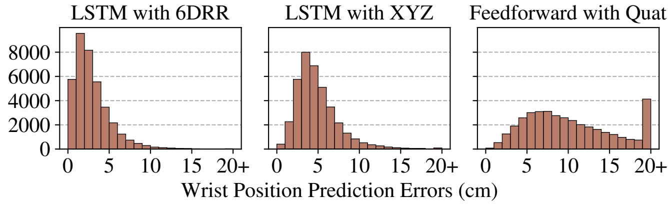

Fig. 5. Error histograms of wrist position predictions of three distinct combinations of network architecture and prediction targets.

forward passes through the network [25], i.e., Monte Carlo (MC) dropout predictions.

More specifically, MC dropout predictions involve keeping the dropout activated for predictions outside of the training process and repeating every prediction multiple times. Producing repeated outputs with dropout results in a distribution of predictions for the same input. The standard deviation of the distribution serves a measure of the prediction uncertainty [25]. Such a distribution allows us to identify cases where smartwatch sensor readings lead to multiple possible arm postures. In such instances, the distribution can become multi-modal, and we can detect and choose the most likely mode based on additional constraints, such as the safest trajectory for the robot.

E. Speech recognition

To further expand the teleoperation capabilities of our smartwatch approach, we incorporate the streaming of microphone data. The recorded audio signal is transcribed into voice commands utilizing the Google Cloud speech-to-text service[1]. This additional interface proves effective and detects commands even when the arm of the user is hanging down. We demonstrate the usability of the speech recognition interface in Section V.

IV. Results

This section discusses and compares overall prediction accuracy of trained models. Further, we relate our findings to reported results in previous related work.

A. Predictive Model Accuracy

The feedforward and LSTM network architectures are scrutinized by their prediction error on the test datasets via a 10-fold cross validation with each of our four introduced prediction targets Polar, XYZ, 6DRR and Quat. We derive the prediction error by calculating the combined distance from the predicted to the ground truth wrist and elbow positions divided by two. In Figure 4, we compare the mean, the median error and root mean squared error (RMSE) of those combined prediction errors.

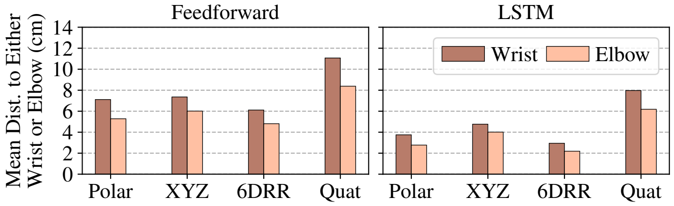

Fig. 6. A comparison of mean prediction errors when focusing on either wrist or elbow positions.

Overall, the LSTM models achieved lower errors than the feedforward models. This is an expected result given that arm movements are inherently a time series data set. The LSTM architecture has an advantage since it maintains an internal state of previous rotations or accelerations thereby providing additional information to the prediction step.

Both network architectures achieved the lowest errors when trained on the 6DRR targets. This finding confirms that continuous rotation representations are more suitable training targets when compared to quaternions or Euler angles [22]. Further, this finding validates our approach of optimizing the search space by utilizing the constraint that upper and lower arm lengths are constant. Using the Polar, 6DRR and Quat targets, predictions are limited to the value ranges of the respective rotation spaces. The confirmed findings of [22] together with fixed arm lengths are plausible reasons for why 6DRR prediction targets achieve better performance.

To investigate if reported average accuracy measurements hide extreme errors, Figure 5 depicts histograms of wrist position prediction errors. Each histogram summarizes the prediction errors for wrist positions of one fold during the conducted 10-fold cross validation. On the left in this comparison, the error distribution for the LSTM with 6DRR combination shows the highest peak at the lowest error. The LSTM with XYZ combination produces on average a higher prediction error, which is noticeable in a more right-shifted and wider error distribution. The combination of feedforward network with quaternion targets features a comparably flat error distribution and more than 4 000 predictions with an error above 20 cm. These observations coincide with our above findings that the LSTM with 6DRR combination makes the most accurate predictions while the feedforward with quaternion combination is the least accurate.

Figure 6 summarizes prediction errors for wrist and elbow positions independently. In general, it is observable that elbow predictions are more accurate than wrist predictions. This is plausible since the elbow has to lie on a sphere around the shoulder, while the wrist lies on a manifold defined by spheres around all possible elbow positions, allowing more room for error. Further, in case of the Polar, 6DRR and Quat targets, wrist positions are estimated by adding a vector with

[1] `https://cloud.google.com/speech-to-text`

lower arm magnitude and with the predicted rotation onto the predicted elbow position. Thus, the error of the predicted elbow position potentially adds to the error of the predicted wrist position.

Altogether, the combination of LSTM architecture and 6DRR targets outperforms other combinations with regards to prediction accuracy for wrist and elbow positions.

B. Comparison to Related Work

The work of [7] follows the the same objective as our paper, namely, the prediction of wrist and elbow positions from smartwatch data. Similar to our approach, they assume a fixed shoulder position and require the body facing direction to be known. In their evaluation they reported median errors of $9.2\,\mathrm{cm}$ for predicted wrist positions and $7.9\,\mathrm{cm}$ for elbow positions. Also [8] used a recurrent neural network to predict wrist and elbow positions. They predict wrist and elbow positions in Cartesian coordinates and report an error of $7.2\,\mathrm{cm}$ and $7.1\,\mathrm{cm}$ for wrist elbow.

The LSTM with 6DRR and coupled with MC dropout predictions presented in our work is also suitable for real-time applications. Our Wear OS app allows to stream sensor data from the smartwatch to any reasonably well equipped system via UDP at $50\,\mathrm{Hz}$. For example, with an Intel® Xeon® W-2125 CPU and a GeForce RTX 2080 Ti GPU we were able to make 150 MC dropout predictions targets at a rate of $\sim 40\,\mathrm{Hz}$. Regarding the prediction accuracy, as reported in Figure 4, our best performing model achieves a more than $4\,\mathrm{cm}$ reduction in median prediction errors compared to results reported by [7], [8]. Specifically, our model resulted in a median error of $2.33\,\mathrm{cm}$ for wrist position predictions and $1.61\,\mathrm{cm}$ for elbow predictions.

Another related approach was proposed in [4]. However, their approach used two IMUs; one IMU on the lower arm and the second on the upper arm. In their real-world experiment they reported a RMSE and standard deviation of $6.9 \pm 2.7\,\mathrm{cm}$ for wrist and $5.2 \pm 2.6\,\mathrm{cm}$ for elbow predictions. Their real-world experiment also required a short calibration procedure for their IMUs based on the work of [26].

As shown in Figure 4, our LSTM with 6DRR achieves a $\sim 40\%$ lower RMSE on our motion capture data while relying only on a *single* IMU. Specifically, it predicts wrist positions with an RMSE and standard deviation of $3.71 \pm 2.49\,\mathrm{cm}$ and elbow positions with an error of $2.99 \pm 2.19\,\mathrm{cm}$ on test data. In conclusion, our approach appears to result in a reduction of prediction error by at least $\sim 40\%$ when compared to previous works by [4], [7], [8] while it remains to be as real-time capable as the approach of [7].

V. USABILITY DEMONSTRATIONS

We combine the increased accuracy of our approach with speech recognition through the microphone of the smartwatch and present two tasks which highlight the advantages of our approach.

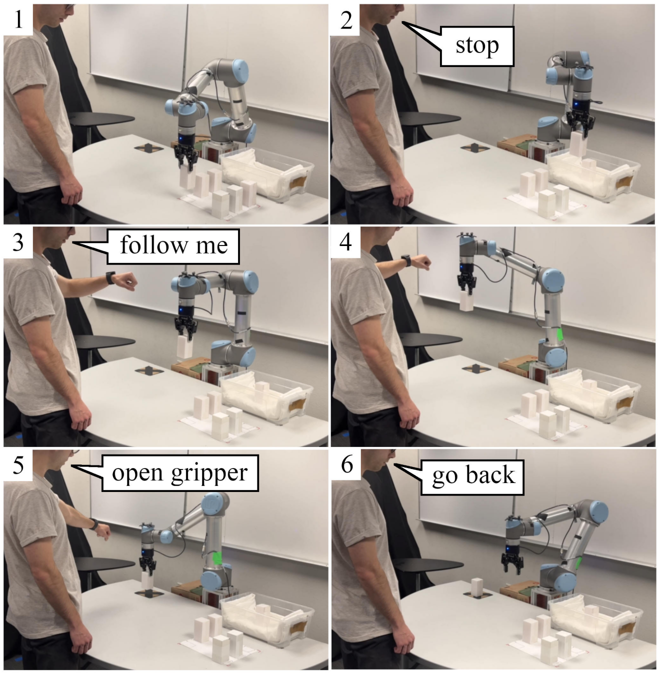

Fig. 8. **Step 1:** The robot picks up cubes and puts them into the tray. **Step 2:** The user says “stop”. The smartwatch recognizes the command and stops the robot mid-task. **Step 3:** The user raises their arm and says “follow me”. The robot moves its end effector to match the wrist position. **Step 4:** The user guides the end effector to a marked position. **Step 5:** The user says “open gripper” and the robot drops the cube. **Step 6:** The user says “go back” and the robot returns to Step 1.

A. Intervention Task

This tasks demonstrates that the smartwatch allows for swift and intuitive human-robot-interaction at any time. A schematic of the intervention task is depicted in Figure 7. The robot autonomously picks the blue cubes one-by-one and places them in the red area. The user can intervene at any time to place one of the cubes in the green area instead. Triggering an intervention is done via a voice command. Thereafter, the robot will mimic the human wrist motions.

The entire procedure is subdivided into six steps, which are depicted and summarized in Figure 8. The tray in this real-world example is the red area from Figure 7, the black square on the left of the robot is the green area and the white cubes are arranged in front of the tray as the blue cubes in Figure 7.

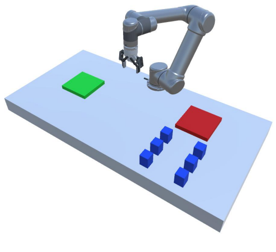

Fig. 7. The intervention task: The robot picks the blue cubes one after the other and places them in the red area. The user utilizes the smartwatch to stop the robot mid-task and to move one cube to the green area instead.

Three users performed the task 10 times each. We measured the times from when the participant said “stop” until the “open gripper” command was received. The distances from the placed cube to the target marked positions were measured with retroreflective markers and the

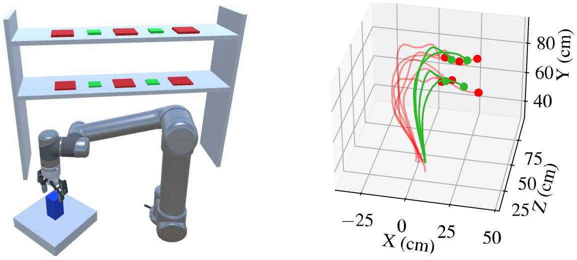

Fig. 9. **Left:** A virtual concept of our learning task. We record six smartwatch trajectories for placing the blue cube onto the red locations. Then, we train a policy to generate new trajectories for placing the cube on the highlighted green positions. **Right:** The red training trajectories are recorded smartwatch data. The green trajectories were generated using our trained policy for the green target positions on the shelf.

OptiTrack system which we used for our motion capture ground truth data. These distance and time measurements provided us with an estimate of how precisely the users could control the robot with the smartwatch and how quickly they could complete the task.

TABLE I
INTERVENTION TASK RESULTS

Part.	Time (s)	Dist. (cm)
1	19.5 ± 2.6	1.87 ± 0.62
2	28.9 ± 8.7	2.21 ± 1.11
3	20.1 ± 4.7	2.19 ± 1.07
All	22.8 ± 7.3	2.09 ± 0.97

The results are summarized in Table I. The average measured time from interrupting the robot until sending it back to its original task was 22.8 ± 7.3 s. On average, all participants placed their cubes within 2.09 ± 0.97 cm from the target position. Every run was successful, which confirms that our smartwatch approach is a suitable tool for the designed intervention task. Further, considering the time and position error, these findings confirm the reported accuracy and real-time capability of our smartwatch approach.

B. *Learning Task*

The goal of the second task is to show an application to the problem of learning from demonstration [12]. In particular, we learn a policy for placing a cube on a shelf, as depicted on the left in Figure 9. A human wearing a smartwatch demonstrates six training trajectories. The human holds the cube in heir hand at the start position and starts recording. Then, the human moves the cube in an arch to one of the six red marked positions on the shelf and repeats the procedure for the remaining goal positions. Since the human can demonstrate the trajectories without moving the robot, data collection is swift and uncomplicated. All training trajectories for this task were recorded within two minutes.

The smartwatch trajectories are depicted in red on the right in Figure 9. We then leverage these trajectories to train a movement policy using the Generative Adversarial Imitation Learning (GAIL) [27] method. As a result, we obtain a movement policy for letting a robot place cubes at any target position on the shelf. To visualize the generalization capabilities of the resulting policy, four generated example trajectories are depicted in Figure 9 on the right. They place the cube in-between the target positions, which are marked as green squares on the left in Figure 9.

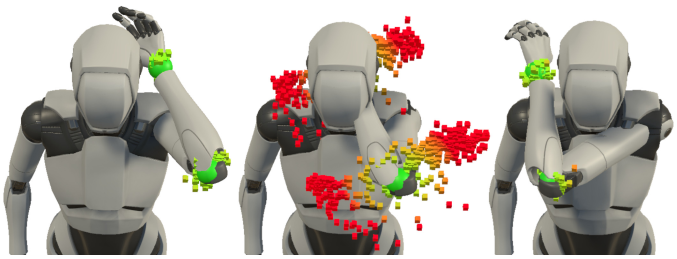

Fig. 10. **Left:** The user wears the smartwatch on their left arm and holds the left hand next to their head. The smartwatch predicts the correct position. **Middle:** The user rotates their wrist back and forth while keeping the hand in the same position. This causes the predicted positions to alternate between positions left or right of the head. **Right:** The predicted wrist position is at the wrong side of the head.

This use-case demonstrates that the smartwatch can be leveraged to train new movement policies to a robot at any time by swiftly recording a set of demonstrations in the same environment. The smartwatch trajectories in this example were collected within two minutes and enable a robot to place a cube anywhere on a shelf given a target placement position.

VI. LIMITATIONS AND FUTURE WORK

Our approach requires the completion of a two-step calibration procedure whenever there is a change in body orientation or location of the user. This is a limitation in comparison to the work of [4]. Their approach utilizes a second IMU which allows users to move and rotate their hip and chest. This limitation can be addressed by adding a second IMU to our smartwatch approach too. To maximize familiarity and ease-of-use, this work presents approaches to leverage the possibilities of the smartwatch to their full extend without adding additional devices. However, promising opportunities for future work can utilize the fact that a smartwatch is typically connected to a smartphone, which the user also wears on their body. The smartphone can serve as a second IMU and enable the tracking of arm movements while the user changes their body orientation or location.

A further limitation is that fast wrist rotations or unergonomic arm motions affect the accuracy of our approach. Figure 10 illustrates an example where the user wears the smartwatch on their left arm and estimated arm postures are visualized with an avatar. The final predicted wrist and elbow positions are the mean of 300 individual MC dropout forward passes. Individual predicted positions are marked as small cubes colored according to their distance to the mean.

The user raised their hand to their ear and, as shown on the left in Figure 10, the position was predicted correctly. Then, the user rotated their wrist back and forth while keeping their wrist position constant. The resulting unusual wrist angles and rapid movements caused predicted positions to alternate between the left and right side of the head. In the middle of Figure 10 it is observable that predictions manifested in bimodal distributions with their modes on the left an right side of the head. The mean, and therefore the predicted elbow and wrist positions, moved into the middle causing the arm of the avatar to go through its head.

The detection and handling of such scenarios shapes promising opportunities for future work. The distributions obtained through the MC dropout predictions allow to detect such scenarios and to dynamically adjust estimated joint positions. If a multimodal distribution occurs, we can consult additional cost functions, i.e., distance to previous positions or risk for the teleoperated robot. It will also be possible to determine the most likely arm posture by consulting additional predictive models, which were trained on different inputs. Having a measure of uncertainty and a distribution of possible solutions is a promising base to improve prediction accuracy in the future.

VII. CONCLUSIONS

This work presents a solution to the problem of estimating human arm poses from a single smartwatch. We propose a simple yet effective two-step calibration procedure to mitigate variability in sensor data and to leverage information about the forward-facing direction of the user. This allows us to devise an optimized model architecture, which achieves a $\sim 40\%$ reduction in prediction error compared to results reported in previous works. Furthermore, our approach generates a distribution of posture predictions, which allows to estimate a measure of uncertainty and to select the best solution from several options in cases of multimodal distributions. By combining arm posture estimations with speech recognition we turn the smartwatch into a ubiquitous, low-cost and versatile robot control interface.

REFERENCES

[1] A. M. Aurand, J. S. Dufour, and W. S. Marras, "Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume," *Journal of Biomechanics*, vol. 58, pp. 237–240, 2017.

[2] G. Nagymáté and R. M. Kiss, "Application of optitrack motion capture systems in human movement analysis: A systematic literature review," *Recent Innovations in Mechatronics*, vol. 5, no. 1., p. 1–9., Jul. 2018.

[3] M. do Carmo Vilas-Boas, H. M. P. Choupina, A. P. Rocha, J. M. Fernandes, and J. P. S. Cunha, "Full-body motion assessment: Concurrent validation of two body tracking depth sensors versus a gold standard system during gait," *Journal of Biomechanics*, vol. 87, pp. 189–196, 2019.

[4] V. Joukov, J. Cesic, K. Westermann, I. Markovic, D. Kulic, and I. Petrovic, "Human motion estimation on lie groups using IMU measurements," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1965–1972.

[5] S. Li, J. Jiang, P. Ruppel, H. Liang, X. Ma, N. Hendrich, F. Sun, and J. Zhang, "A mobile robot hand-arm teleoperation system by vision and IMU," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 900–10 906.

[6] G. Ates, M. F. Stolen, and E. Kyrkjebo, "Force and gesture-based motion control of human-robot cooperative lifting using IMUs," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2022, pp. 688–692.

[7] S. Shen, H. Wang, and R. Roy Choudhury, "I am a smartwatch and i can track my user's arm," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016, pp. 85–96.

[8] W. Wei, K. Kurita, J. Kuang, and A. Gao, "Real-time limb motion tracking with a single IMU sensor for physical therapy exercises," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 7152–7157.

[9] K. Yang, B. Tag, C. Wang, Y. Gu, Z. Sarsenbayeva, T. Dingler, G. Wadley, and J. Goncalves, "Survey on emotion sensing using mobile devices," *IEEE Transactions on Affective Computing*, pp. 1–20, 2022.

[10] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, 2023.

[11] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Foundations and Trends in Robotics*, vol. 7, no. 1, pp. 1–179, 2018.

[12] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1371–1394.

[13] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, 2002, pp. 1398–1403 vol.2.

[14] C. Ott, D. Lee, and Y. Nakamura, "Motion capture based human motion recognition and imitation by direct marker control," in *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 399–405.

[15] L. Hasenclever, F. Pardo, R. Hadsell, N. Heess, and J. Merel, "CoMic: Complementary task learning; mimicry for reusable skills," in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., vol. 119. PMLR, 2020, pp. 4105–4115.

[16] D. Rakita, B. Mutlu, and M. Gleicher, "A motion retargeting method for effective mimicry-based teleoperation of robot arms," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2017, pp. 361–370.

[17] J. S. Dyrstad, E. Ruud Øye, A. Stahl, and J. Reidar Mathiassen, "Teaching a robot to grasp real fish by imitation learning from a human supervisor in virtual reality," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7185–7192.

[18] M. Hirschmanner, C. Tsiourti, T. Patten, and M. Vincze, "Virtual reality teleoperation of a humanoid robot using markerless human upper body pose imitation," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 259–265.

[19] J. DelPreto, J. I. Lipton, L. Sanneman, A. J. Fay, C. Fourie, C. Choi, and D. Rus, "Helping robots learn: A human-robot master-apprentice model using demonstrations via virtual reality teleoperation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 226–10 233.

[20] L. S. Yim, Q. T. Vo, C.-I. Huang, C.-R. Wang, W. McQueary, H.-C. Wang, H. Huang, and L.-F. Yu, "WFH-VR: Teleoperating a robot arm to set a dining table across the globe via virtual reality," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4927–4934.

[21] V. Villani, L. Sabattini, G. Riggio, A. Levratti, C. Secchi, and C. Fantuzzi, "Interacting with a mobile robot with a natural infrastructure-less interface," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12 753–12 758, 2017.

[22] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023, pp. 5738–5746.

[23] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[25] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[26] D. Tedaldi, A. Pretto, and E. Menegatti, "A robust and easy to implement method for IMU calibration without external equipments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3042–3049.

[27] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.