SparseIMU: Computational Design of Sparse IMU Layouts for Sensing Fine-Grained Finger Microgestures

ADWAIT SHARMA[1], CHRISTINA SALCHOW-HÖMMEN[2,3], VIMAL SURESH MOLLYN[1], ADITYA SHEKHAR NITTALA[1], MICHAEL A. HEDDERICH[1], MARION KOELLE[1], THOMAS SEEL[3,4], and JÜRGEN STEIMLE[1], [1]Saarland University, Saarland Informatics Campus, Germany, [2]Department of Neurology, Charité-Universitätsmedizin Berlin, Germany, [3]Control Systems Group, Technische Universität Berlin, Germany, and [4]Department of AI in Biomedical Engineering, FAU Erlangen-Nürnberg, Germany

Gestural interaction with freehands and while grasping an everyday object enables *always-available input*. To sense such gestures, minimal instrumentation of the user's hand is desirable. However, the choice of an effective but minimal IMU layout remains challenging, due to the complexity of the multi-factorial space that comprises diverse finger gestures, objects and grasps. We present *SparseIMU*, a rapid method for selecting minimal inertial sensor-based layouts for effective gesture recognition. Furthermore, we contribute a computational tool to guide designers with optimal sensor placement. Our approach builds on an extensive microgestures dataset that we collected with a dense network of 17 inertial measurement units (IMUs). We performed a series of analyses, including an evaluation of the entire combinatorial space for freehand and grasping microgestures (393K layouts), and quantified the performance across different layout choices, revealing new gesture detection opportunities with IMUs. Finally, we demonstrate the versatility of our method with four scenarios.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; **Interaction techniques**; **Gestural input**.

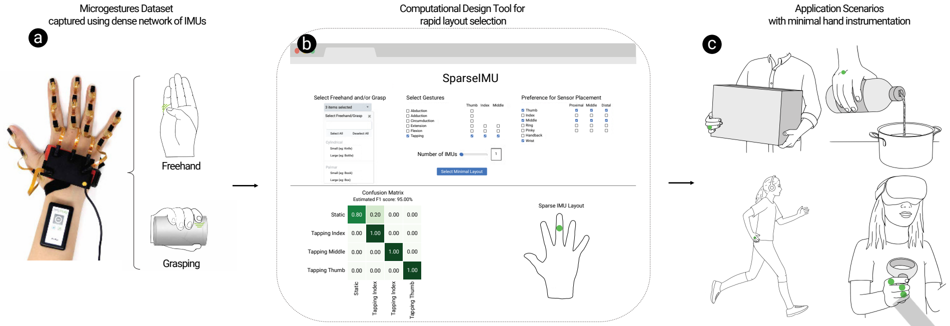
Additional Key Words and Phrases: Gesture Recognition; Hand Gestures; Sensor Placement; IMU; Objects; Design Tool

1 INTRODUCTION

In situations found in everyday life, people's hands can be free, but are often times also busy with objects they hold, carry, or use. Interaction techniques for always-available input [77] should be designed considering these settings. Prior work in HCI has established the design foundation of freehand and grasping microgestures: subtle finger gestures that can be performed with free [12] and busy hands [82, 83, 102]. These gestural techniques enable eyes-free, always-available interaction in demanding situations. However, implementing such input solutions is challenging, and it becomes even more complex if the recognition system needs to recognize microgestures in both free-hand and busy-hand conditions. Apart from the numerous spatial configurations that are possible with the dexterous movement of multiple fingers, the recognition system also needs to account for occlusions that are typically created when hands are occupied. These challenges make the deployment of optical sensing techniques very demanding [66].

Authors' address: Adwait Sharma[1], sharma@cs.uni-saarland.de; Christina Salchow-Hömmen[2,3], christina.salchow@charite.de; Vimal Suresh Mollyn[1], ms123vimal@gmail.com; Aditya Shekhar Nittala[1], nittala@cs.uni-saarland.de; Michael A. Hedderich[1], mhedderich@lsv.uni-saarland.de; Marion Koelle[1], koelle@cs.uni-saarland.de; Thomas Seel[3,4], thomas.seel@fau.de; Jürgen Steimle[1], steimle@cs.uni-saarland.de, [1]Saarland University, Saarland Informatics Campus, Saarbrücken, Germany, 66123 and [2]Department of Neurology, Charité-Universitätsmedizin Berlin, Berlin, Germany, 10117 and [3]Control Systems Group, Technische Universität Berlin, Berlin, Germany, 10587 and [4]Department of AI in Biomedical Engineering, FAU Erlangen-Nürnberg, Erlangen, Germany, 91052.

1073-0516/2022/10-ART
https://doi.org/10.1145/3569894



Select Gestures

	Thumb	Index	Middle
□ Abduction	□		
□ Adduction	□		
□ Circumduction	□		
□ Extension	□	□	□
□ Flexion	□	□	□
☑ Tapping	☑	☑	☑

Preference for Sensor Placement

	Proximal	Middle	Distal
☑ Thumb	☑	☑	☑
□ Index	□	□	□
☑ Middle	☑	☑	☑
□ Ring	□	□	□
□ Pinky	□	□	□
□ Handback			
☑ Wrist			

Confusion Matrix
Estimated F1 score: 95.00%

	Static	Tapping Index	Tapping Index	Tapping Thumb
Static	0.80	0.20	0.00	0.00
Tapping Index	0.00	1.00	0.00	0.00
Tapping Middle	0.00	0.00	1.00	0.00
Tapping Thumb	0.00	0.00	0.00	1.00

Fig. 1. We present a data-driven method for designing effective microgesture recognition systems that only require a sparse set of IMUs. (a) The method builds on an extensive microgestures dataset that includes Freehand and Grasping conditions, collected using a customized dense IMU setup. (b) A design tool helps designers to rapidly select sparse IMU layouts for a desired set of gestures and optional constraints. (c) It informs effective sensing solutions with minimal instrumentation for a broad variety of applications.

One approach to address the challenges of hand occlusion includes extensive hand instrumentation. Prior work has demonstrated promising results for hand pose reconstruction while manipulating objects. For instance, Han et al. achieved this by employing deep learning combined with markers attached all over the hand [33]. Yet, extensive hand instrumentation is undesirable for practical use, as it will hinder the user going about their other everyday tasks. Other work has shown promising results by making use of sparser hand instrumentation, with only one or a few Inertial Measurement Units (IMUs) [28, 84, 103, 107]. IMUs are easy to deploy and can be ergonomically worn in a light-weight ring form factor. In addition, they are sensitive to subtle movements and do not suffer from occlusion problems.

However, the IMU layout, i.e., the specific locations where IMUs are placed on the hand and fingers, is crucial for accurate gesture detection. Designing an IMU layout that is sparse while capable of accurately detecting gestures is a challenging task and depends on multiple factors. These factors include the desired choice of microgestures, the hand conditions (free-hands v/s busy-hands or a combination of both), the grasp type (associated with holding an object), and the user-defined constraints for IMU placement. So far, IMU layouts for sparse instrumentation had to be chosen manually, in an ad-hoc manner, or using systematic trial-and-error [84, 103]. Considering the complexity of the multi-factorial design space, this manual process is time-consuming and may lead to far sub-optimal layouts. This work addresses this challenge by supporting designers of gesture recognition systems to make well-informed and rapid decisions.

We present *SparseIMU*, a computational design approach to assist interaction designers and engineers in creating gesture recognition systems, which effectively recognize a desired set of *freehand* and/or *grasping microgestures* with minimal hand instrumentation. A web-based design tool provides designers with the possibility to specify high-level requirements (e.g., desired set of gestures and grasps) and designer-specified constraints (e.g., locations on the hand and fingers that shall remain un-instrumented, and total number of IMUs to be deployed). It then automatically selects an optimal sparse IMU layout matching the given preferences as shown in Figure 1-b. In addition, the tool predicts the expected performance of gesture classification, including a confusion matrix. This allows the designer to assess the expected quality of a solution and to rapidly explore design alternatives in a well-informed manner. To the best of our knowledge, our computational approach and design tool are the first to enable the rapid iterative design of sparse IMU-based sensing solutions for microgestures.

The presented data-driven approach is based on our collection of an extensive microgestures dataset, captured with a customized hardware setup containing 17 synchronized IMUs placed all over the dominant hand. It comprises of 18 gestures and three non-gesture states performed with an empty hand as well as on 12 objects that cover all the six grasp types from Schlesinger's taxonomy [79], collected from 12 participants. Our dataset comprises fully annotated dense IMU data. This allowed us in computing models with all possible IMU layouts in Freehand, Grasping, and Both Combined conditions [in total $3 \times (2^{17} - 1) = 393{,}213$ models].

To investigate the potential of making conscious design choices when selecting a specific sparse IMU layout, we performed a series of empirical analyses looking into effects on recognition performance. Chiefly we have made the following observations: i) Sparse layouts with a very low number of IMUs achieve high recognition rates of 90% F1 score and above, ii) the choice of finger segment for IMU placement can be crucial, and iii) IMUs placed on a non-gesturing finger can be utilized to detect gestures from another finger. These findings reveal insights that uncover the great potential of sparse IMU layouts in gesture detection.

The collected microgestures dataset additionally serves as the building block for deriving a fast method to select sparse layouts. We employ a variant of a well-known metric from Machine Learning (ML), *Feature Importance*, to rapidly select optimized sparse layouts. We validate our SparseIMU approach with the classification results from the entire combinatorial space; the results demonstrate our method's efficacy. While generating results based on the entire combinatorial space is prohibitively time-consuming for a practical design task, our method generates results within minutes on a commodity laptop. Consequently, our approach can be used to enable rapid design iterations.

We demonstrate the benefits of the SparseIMU approach using four exemplary application cases. Finally, our user evaluation shows congruence in the tool's predictions and live gesture recognition. These show how the tool enables designers and engineers to rapidly determine optimal sparse IMU layouts, identify trade-offs, and fine-tune designs. Together, our rich microgestures dataset and computational design tool enable a rapid iterative design process in which designers can create, explore and modify custom sensor layouts in a well-informed manner.

In summary, the main contributions of this article are:

- **Microgestures Dataset**: Using 17 IMUs placed on the hand, we captured microgestures and hand manipulations with freehand and while holding 12 objects, performed by 12 participants. Overall, it consists of 13,860 trials, resulting in a total of 3,404,276 frames. We release our fully annotated dataset at: *https://hci.cs.uni-saarland.de/projects/sparseimu*.
  We hope it will be beneficial for the research community to gain insights into the subtle finger movements that happen during holding and manipulating objects, opening up a number of opportunities for future research in diverse areas such as gesture design or analysis of finger dexterity during object manipulation.
- **Computational Design Approach for Detecting Microgestures**: We present a method and graphical tool to rapidly select sparse IMU layouts that achieve a good trade-off between minimal instrumentation and high recognition accuracy, taking into account various user-defined preferences. We also release our computational tool code with the dataset at the aforementioned link.
- **Series of Empirical Analysis**: We quantified gesture recognition performance in different settings to thoroughly understand the effect of segment choice, the potential of detecting gestures from the IMUs on a non-gesturing finger, and generalizability across different users.
- **Application Scenarios**: Four application scenarios from diverse and representative domains illustrate how designers and engineers can leverage the potential of our approach for concrete design tasks.

2 RELATED WORK

Our work primarily lies at the intersection of microgestures, gesture sensing and gesture design tools.

2.1 Freehand and Grasping Microgestures

Microgestures (or Micro-interactions) refer to the subtle finger movements that are fast, easy to perform, and may not interrupt the other ongoing tasks [5]. They enable myriad applications in different scenarios [31, 34, 35]. Such microgestures are further interesting because they can be performed while holding an object (e.g., a steering wheel [2]) in hand. In such conditions, the physical constraints of each finger vary based on grasp and object type.

Prior works have taken several paths for designing gestures that are possible with the same hand while holding an object: from interviewing experts [102] to using prototypes for understanding holding behavior [90]. Additionally, there is a rich body of prior work on the design of hand gestures (see [94] for a survey). These works adopted different design methods to develop gesture sets and focused on either empty hands or holding an object. A common technique to design gestures in HCI using Guessability-style elicitation studies was proposed by Wobbrock et al. [99]. We build on prior conceptual work that used this technique for deriving single-hand gestures in an empty hand [12], as well as for busy hands holding objects of different grasp types [83]. By consolidating a uni-manual gesture set from these two works, our goal is to enable a generic and scalable solution. In this article, we advance these conceptual foundations through a sensing approach, which makes their application in real-world deployments possible.

2.2 Sensing Technologies to Detect Microgestures

Various sensing techniques have been proposed to detect finger gestures. While each has its advantages and disadvantages, it is worth noting that the selected sensing type has a crucial role in the hardware's placement location and the enabled gesture set. A large body of pioneering work relies on optical sensing for detecting microgestures. CyclopsRing [13] proposed a finger-worn fisheye camera device to detect on-finger and in-air pinch and slide gestures, as well as palm-writing, FingerInput [85] demonstrated detection of thumb-to-finger gestures using a head-mounted or shoulder-mounted depth sensor. Sugiura et al. [87] have shown recognition of discrete finger-based gestures using an array of photo reflective sensors placed on the back of hand. A variety of other sensing approaches include ultrasonic [41, 63, 64], infrared [27, 44, 63, 108], pressure [16, 21, 98], magnetic [3, 37, 69], and capacitive techniques [7, 91]. Due to the advances in deep learning, researchers have also demonstrated the detection of fine finger movements using radar sensing [96]. These systems show some remarkable success in enabling gesture recognition in freehand conditions. However, due to the inherent property of such sensing technologies, these approaches can fail under occlusion caused by holding an object.

Although occlusion can be compensated by augmenting an object, the scalability issue can be a bottleneck to practical deployment. Another approach is based on data gloves that are instrumented with sensors [26, 33, 88]. Despite being able to capture high-fidelity information, they are often bulky and hence impede the dexterity of fingers. For a more detailed overview of the different vision-based and glove-based approaches, we refer to [14]. The most closely related approach to our goal of supporting gesture detection in both conditions, freehand and while grasping an object, is proposed using an electromyography band by Saponas et al. [77]. However, the selected grasp variations and the number of gestures are limited due to the lower resolution of the technique. Laput et al. used a smartwatch accelerometer to detect coarse freehand gestures and also demonstrated activity detection [52, 53]. Furthermore, placing an IMU on finger segments has been shown to be effective in capturing subtle finger movements and does not get affected if there is an object in hand [67, 92, 103]. Recently, DualRing [55] presented the usage of two IMUs placed on the thumb and index finger's proximal segment to detect four grippings postures but did not consider any gestures while holding objects. Bardot et al. [6] suggested the usefulness of a smart-ring (embedded with an IMU and a touchpad) for gestures in hands-busy situations. We take inspiration from these systems and selected IMUs as our sensing technique to simultaneously support gestures with freehand and while holding an object conditions.

2.3 Sparse Sensor Layouts

While the aforementioned works presented a viable technological solution to capture finger information while holding objects, these do not investigate the optimal sensor placement to fully harness the capability of IMU sensing. Yet, the placement of sensors is as crucial for gesture detection as selecting the appropriate sensing type. This is prominently shown by the findings from Gu et al. [28] and Shi et al. [84] who used a single IMU and determined that touch-contact recognition performance can be strongly increased by investigating the optimal position on different finger segments. Lin et al. [56] used an array of strain gauge sensors to detect finger gestures based on American Sign Language and reported the minimum accuracy of 70.8% can be increased to 95.8% for an identified optimal location. Kubo et al. [51] applied pizo-electric elements to detect thumb, thumb-to-finger, gestures, and palm touches and reported the change in accuracy from 90.6 to 96.6% for an optimal location. All these works employed the trial-and-error approach of moving the sensor at different locations, requiring considerable time and effort. We leverage our dense setup of 17 IMUs to avoid the process of repeating manual trials involving the movement of the single sensor at different locations. Using the principle of compressed sensing and other sophisticated techniques, a large body of work has demonstrated that human body pose can be reconstructed by a significantly reduced number of sensors [1, 20, 39, 68, 81]. However, as mentioned by Brunton et al. [10], reconstruction and classification are two different problems. While some work exists that uses sparse representation for gesture classification, it mainly uses visual data [62, 75]. To the best of our knowledge, our work is the first that presents a computational method for identifying a sparse layout for gesture classification using IMUs.

2.4 Gesture Design Tools

Gesture design and recognition have received a lot of attention in HCI. Wobbrock et al. [100] proposed $1 for rapid prototyping of gesture-based interfaces. Long's Quill [58], a pen gesture system, enables users to create pen gestures by example. Similarly, several design tools have been presented in the HCI literature for the design of various gestures. These include work from Ashbrook et al. [4] and Kohlsdorf et al. [49] that allows the designer to compare a gesture with a corpus of everyday activity data for false positive testing. EventHurdle [47], M.Gesture [46] and Mogeste [70] enable users to compose custom gestures on mobile devices. Gesture Coder [60] is a tool to help developers add multi-touch gestures by demonstrating them on a tablet's touchscreen. While there are existing machine learning-based frameworks and platforms for quickly prototyping and debugging various classifiers and implementing custom machine learning pipelines [36, 71, 72], they are targeted for programmers and do not consider aspects of interaction design. On the other hand, recent advances in technology have enabled novice users to train and classify custom ML models without the need for programming expertise [61]. However, these majorly address image or audio classification problems. Our main goal behind this work is to use machine learning as a design material [18] and enable designers to prototype custom microgestures without the need for having expertise in ML and programming. Motivated by the challenges of designing a sparse sensor layout, we strive to provide designers with a computational tool that abstracts from the complexity of multiple factors (choice of gesture, object, and location constraint), which are conventionally tuned by manual efforts and require technical skills.

3 MICROGESTURES DATASET

Researchers in the computer vision community have contributed various datasets comprising hand-object manipulations [8, 25, 89]. Yet, these do not include explicit *finger gestures*. Our dataset is the first attempt to collect hands-free and busy interaction along with finger microgestures. We use a dense network of 17 IMUs to capture high-dimensional sensor data with nearly full degrees of freedom (DOFs) of the hand/finger space. This is different from prior work wherein a single sensor has been shifted to different locations in different trials for finding the optimal placement [84]. Our high-dimensional data enables employing novel algorithmic approaches to uncover hidden phenomena; some of them are mentioned in the following sections. Overall, our dataset focuses on finger gestures – performed by different fingers – on objects with diverse grasp types, as well as with free hands. It also comprises hand-object manipulations with different intents, such as holding an object, using it as suggested by its primary purpose (e.g., writing with a pen), and handling it in an unscripted manner (e.g., fiddling). Although the dataset is intended to analyze microgestures, it can serve other purposes in future research, including enriching our understanding of finger movements during hand-object interaction, creating synthetic data, or pre-training neural networks.

3.1 Dense IMU Setup

Instead of utilizing commercially available gloves or marker-based solutions [23, 33], we performed the data collection with a customized hand sensor system that preserves the cutaneous properties of the hands, the sense of touch, and does not suffer from occlusion. The sensor system is shown in Figure 2. It offers an unobtrusive setup of 17 synchronized IMUs [76, 93] that provide detailed information about the full articulation of a human hand. It includes 9DOF inertial sensors with 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer (MPU9259, InvenSense Inc., CA, USA) with a footprint of 3×3 mm, deployed on all three segments of all five fingers using a medical-grade skin-friendly adhesive tape (Helvi Mogritz). The finger IMUs are mounted on flexible sensor strips and connected to a base unit attached at the hand's back, which includes an additional IMU. A customized fixture with a thin velcro belt is used to fasten the base unit on the hand, and the data is sent to the computer through a USB connection. We also attached a wireless IMU (RehaGait, Hasomed GmbH, Germany) on the distal forearm, to include data comparison from existing consumer devices like smartwatches or fitness trackers, resulting in a total of 17 IMUs. All IMUs are precisely time-synchronized, and the data is captured at a framerate of 100 Hz. We refer to Salchow-Hömmen et al. [76] for full details on formal hardware validation, which found that sensor readings are accurate enough to infer fingertip positions with errors < 2 cm. For the use of the raw IMU data, the hardware does not require any calibration, making it particularly practical and feasible for studies. However, we integrated an initial pose with the hand flat on the table and the straight thumb abducted at a known angle for a few seconds at the beginning of each subject's recording, in order to boost the dataset's versatility in light of potential future uses where a baseline or calibration pose might be desired. We also note that the framerate of

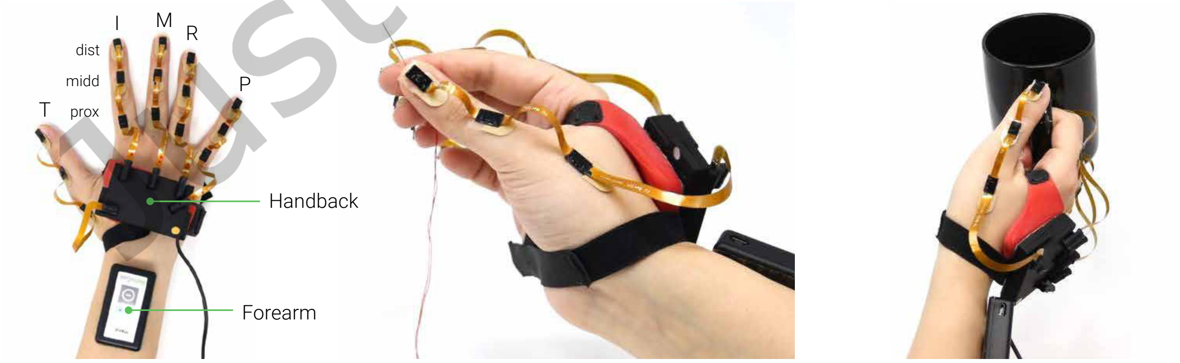


Fig. 2. Hardware setup with 17 synchronized IMUs placed all over the dominant hand. It preserves cutaneous properties and allows unobtrusive interaction with complex object geometries. The left image labels describe the spatial notation of each IMU used in our analysis.

our dense setup of 17 IMUs is in line with that of Xu et al.'s [104] recent work, which suggests that 100 Hz is sufficient for hand gestures' classification. Furthermore, prior studies have found that even the quick movements of the fingers are slower than 10 Hz [40, 42].

3.2 Objects Representing Grasp Variations

We collected data in Freehand and while Grasping an object conditions. For the latter, we selected a set of objects that are representative of real-world tasks. Specifically, we chose objects labeled in the *VLOG Dataset* [24] which is based on internet video logs of everyday activities. To ensure we have representatives for each type of grasp, we categorized the objects based on Schlesinger's Grasp Taxonomy [79]; this has been widely employed by prior works [19, 22, 77, 83]. For each grasp type, we focused on non-deformable objects with two size variations Small (S) and Large (L). The VLOG Dataset does not contain objects that correspond to *Small Tip* and *Spherical* grasps, which is presumably a result of not all grasp types being equally well-represented in everyday life [11]. Therefore, we added two additional objects, a Needle and Pestle, to obtain an exhaustive list of objects covering all grasp types [83]. The complete set of 12 objects and their corresponding grasp type is shown in Figure 3.

3.3 Gesture Set and Non-Gesture States

For Freehand and Grasping conditions, we collected finger movements while performing microgestures and non-gesturing states. For the microgestures, we focused on conscious subtle finger movements that do not require altering the grasp. We selected six primitive finger movements based on bio-mechanical characteristics [43, 95], shown in Figure 4: *Tap*, *Flexion*, *Extension*, *Abduction*, *Adduction*, and *Circumduction*. For consistency of gestures across different fingers, we use the Ring finger as the reference to define Abduction (away from the Ring finger) and vice-versa for Adduction gestures. Furthermore, the swipe gesture was recorded with the participant's

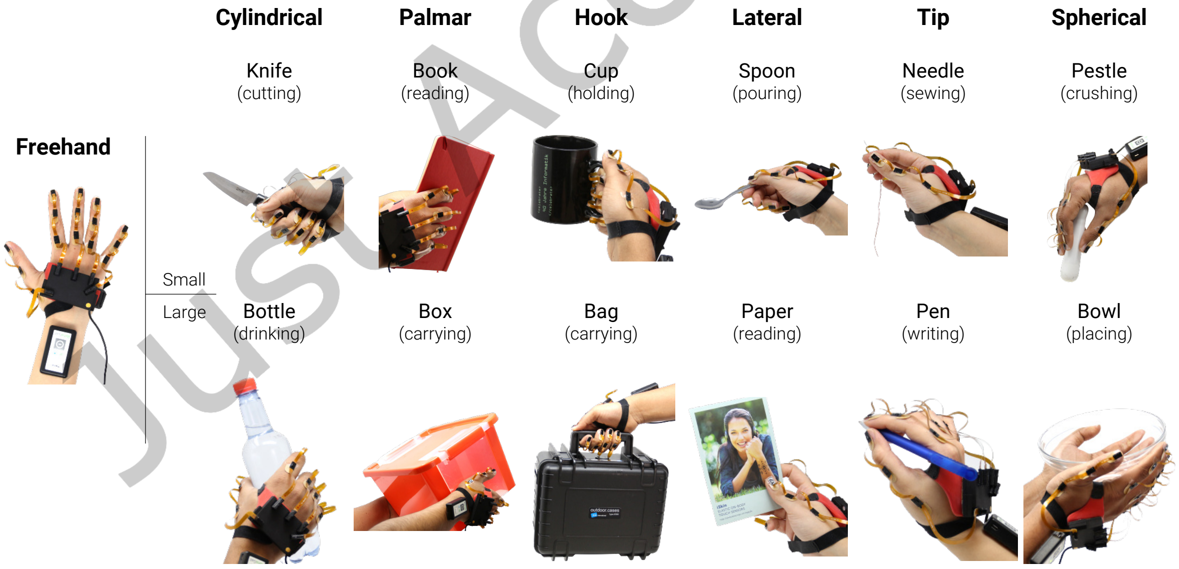


Fig. 3. Using a dense network of 17 IMUs placed on the hand, the microgestures dataset was collected for Freehand and while Grasping 12 objects covering each of the six grasp types with two variations.

finger motion from one extreme until it reached the opposite extreme. Following Ashbrook's definition of micro-interactions [5], we further limited our set to gestures with a short duration (4 seconds or less). Moreover, we centered our data collection on single-finger gestures because they promise to increase robustness [82]. In terms of gestural input, these movements translate to both - continuous and discrete gestures through directional sliding and tapping.

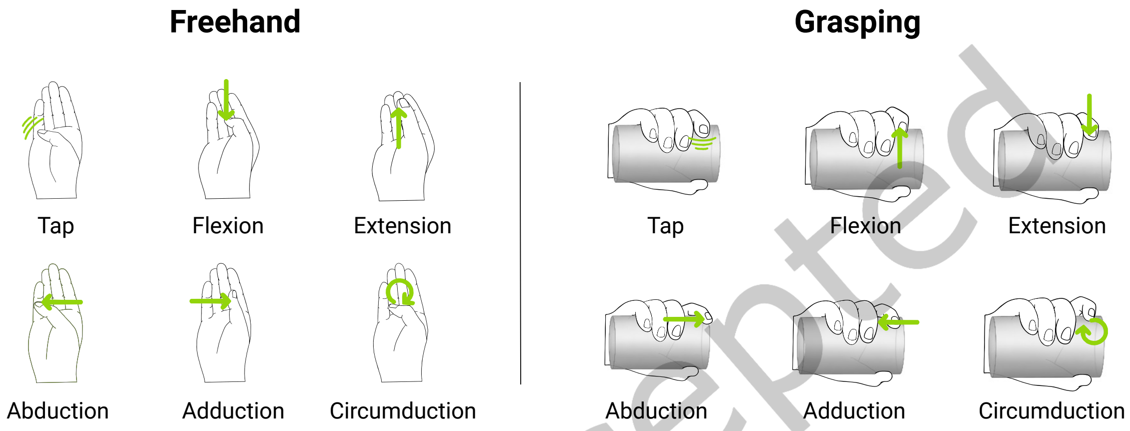


Fig. 4. The Dataset includes six gestures performed with three fingers - Tap, Flexion, Extension, Abduction, Adduction and Circumduction - resulting in a total of 18 gestures. Additionally, data was recorded for three non-gesture classes: Static hold (just holding the object), performing Primary action while holding the object, and an Unscripted action where the user was free to perform any custom movements.

The collected non-gesture states include a variety of finger movements that users perform consciously or unconsciously during conventional hand/object interaction. For instance, free hand movements while talking, adjusting the grip, turning the object for visual inspection, manipulating the object, or fiddling. For capturing non-gesture conditions, we recorded *Static hold*, *Primary action* (e.g., writing with a pen, drinking with a glass), and *Unscripted actions* (e.g., adjusting grip, fiddling). The participants were given no explicit instructions while the data for Unscripted action was recorded.

Since moving a finger while holding an object risks dropping the object, we empirically verified which fingers can be moved while holding objects. To consolidate our choice of finger movements, we conducted a pilot study. Two interaction design experts independently recorded their response on a 7-point Likert scale (1: impossible to perform and leads to dropping the object; 7: very intuitive and easy to perform). This resulted in a total of 360 gestures: 6 (gestures) × 5 (fingers) × 12 (objects) inspected by each expert. Of 720 Likert scale readings, 42 gestures received a rating of 1 by both the experts and these were marked as impossible. Consequently, we focus on the Thumb, Index, and Middle fingers as our main gesture fingers; a choice which is in-line with prior works [12, 82].

3.4 Participants

We recruited 12 participants (6M, 6F, mean age: 26.1; SD: 3.4) with different professional backgrounds, including computer graphics researcher, firefighter, and kindergarten teacher. Ten were right-handed, and two reported themselves as ambi-dexterous. We measured their hand size from the Wrist to each finger's tip and found an average length to Thumb's tip: 137mm (SD:8mm), Index: 181mm (SD:12mm), Middle: 192mm (SD:12mm), Ring: 181mm (SD:10mm), and Pinky: 157mm (SD:9mm). For context, the average hand length (middle finger's tip to the wrist crease) is 193 mm and 180 mm for males and females, respectively [74]. Participation to our data collection was voluntary while adhering to the institution's COVID-19 rules and regulations, and each participant received a compensation of 30 Euros.

3.5 Task and Procedure

Before starting the data collection, we demonstrated the gestures on an abstract cylindrical object that was not used any further. Once the participants got familiarized with the gestures, we attached the hardware to their dominant hand, and they performed the initial pose by placing the hand on the table. For the Grasping condition, we asked the participants to perform gestures on the object (while maintaining the grasp), and use the palm as the surface for the Freehand condition. Of note, the same hand was used for holding the object and for gesturing. Furthermore, the directional orientation was kept constant across each participant. They performed all the gestures while sitting on a chair, except for Box and Bag, wherein we systematically added variation in posture and orientation for each participant by asking them to perform the gestures while standing and facing perpendicularly. We counterbalanced the two conditions (Freehand and Grasping) and further counterbalanced the order of objects (grasp variations). Once the Freehand or the Grasp variation was selected, we presented the gestures with the specific finger name and non-gesture states in a randomized order. We recorded five trials for each gesture. To collect data from non-gesture states without interruption, we recorded one long sequence of around 30 seconds and split it into five trials. The dataset collection took approximately 3 hours per participant with breaks in-between to avoid fatigue. The sessions were also video recorded. Using a custom MATLAB application, the experimenter manually annotated the trials during data collection with the participants orally communicating the start and stop of the gesture. The labels include information about the freehand or specific grasp variation, gestures along with the instructed finger, and the three non-gesture states. Overall, our dataset contains a total of 13,860 trials (1,155 trials × 12 participants) with 18 different gesture and three non-gesture states performed on 12 Grasp variations and with Freehand.

4 DATASET ANALYSIS TO UNDERSTAND IMU PLACEMENT

The usage of IMUs in HCI has been explored for gestural input; the most common approach is to place a single IMU on the gesturing finger [28–30, 84, 107]. However, very little is known about the relationship between the precise position of IMU(s) and its effect on classification performance. To understand the multitude of factors affecting the overall classification performance, we sought to systematically investigate different perspectives, including the quantity of IMUs, variation between different finger segments, alternative IMU placement location to simultaneously achieve higher recognition and usability, lastly, evaluate the feasibility of a user-independent recognition model. An in-depth understanding would not only enable taking full advantage of the IMU sensing capabilities and fine-tuning IMU placement to achieve the maximum performance for a given set of gestures, but also uncover hidden patterns to identify optimal designs of gesture sensing devices.

This section first describes our classification pipeline and a series of empirical analyses, which offers new insights into the design of sparse IMU layouts for hand microgesture recognition.

4.1 Feature Extraction and Classifier Selection

Aiming to understand the underlying factors affecting performance rate due to IMUs' location, we started off by creating a classification pipeline. Given the size of our search space has the large number of 393K layouts, we created a gesture detection pipeline with two essential requirements: scalable and rapid train-test time.

Feature Extraction. From a given trial and for each of the 9 axes of an IMU, we extract six statistical features: maximum, mean, median, minimum, standard deviation, and variance. In total, the number of features from all 17 IMUs × 9 axes × 6 features amounts to 918. To compile this list of features, we drew inspiration from the automatic feature extraction library, TsFresh [15], which has shown promising results in prior work on gesture and activity recognition [27, 45, 57]. Due to multiple sensors and reduced computational load, we used the minimum configuration of the library's functionalities. To further minimize the effect of different trial lengths, we removed the sum and length features. Due to the lower sampling rate of our 17-IMUs setup as compared to single-sensor approaches [53], we did not extract features from the frequency domain. However, we note that our released dataset will allow the research community to feed more features of TsFresh into the neural network [45], take advantage of a single feature, such as derivatives as input into the neural network [84], or further perform feature engineering for input in non-neural-network or neural-network classifiers to improve the recognition rate based on the optimal location. In Section 4.1, we show the correlation of our selected features and a different set of features from related work to show the correlation in the ranking of layouts.

Method. We selected 10 random participants as training set and the remaining two as test set (80:20 split) and created grasp-independent models, i.e., the class labels do not include any grasp information. We also performed a leave-one-person-out analysis in Section 4.5. For our multi-class classification, we used 19 classes: (3 fingers × 6 gestures) + 1 Static hold. Different IMU layouts may contain different amounts of IMUs (from 1–17); therefore, to compare different state-of-the-art classifiers and estimate the classification time required for the full combinatorial classification, we evaluated randomly selected 100 layouts for a given IMU count of 1–17, totaling 1,435 layouts. Note, for count = 1, 16, and 17, the total possible layouts are slower than 100.

Classifier Selection. We fed our extracted features into multiple commonly used classifiers to evaluate their recognition rate and training time. Specifically, we used scikit-learn's implementation of Support Vector Classification (SVC), Logistic Regression (LR), k-nearest neighbors (KNN), Random Forest (RF) with max_depth = 30; and PyTorch implementation for Neural Network (NN) with 4 fully connected layers of decreasing hidden layer size (n = 1024, 512, 256, ReLU activation) and a final softmax activated classification layer. Only NN models were trained on a GPU machine and others on a 40-core CPU. We used the default parameters for all the classifiers to perform trial-by-trial basis classification. As a performance metric, we used the macro average of the F1 score because it considers both precision and recall.

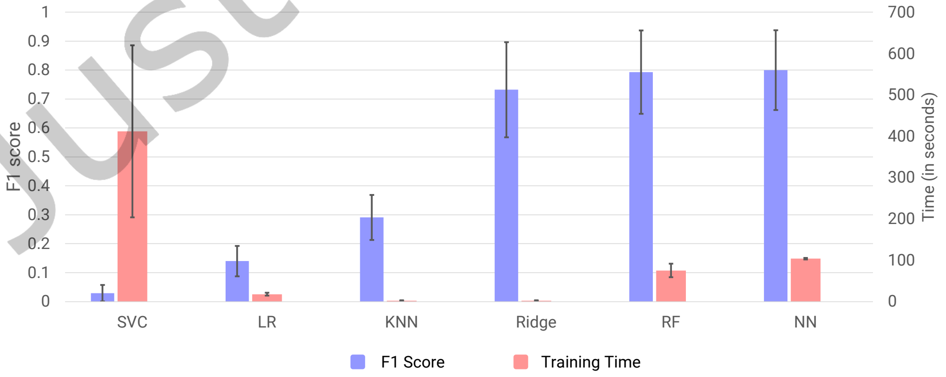


Fig. 5. Comparison between average F1 score obtained by different classifiers and their training time for 1,435 IMU layouts. The error bars depict one standard deviation.

Results. As shown in Figure 5, the F1 score and training time largely depend on the choice of classifiers. Since we wanted to use the same classifier for multiple settings in the following analyses, as well as the later-described computational design tool (see Section 6) – we opted for Random Forest. This classifier achieves an average F1 score close to the highest one obtained by Neural Network while having a lower training time than Neural Network. Furthermore, RF models can be easily computed on a consumer-grade CPU machine. In-line with findings from prior work [101], our results show that Random Forest Classifier has superior performance than KNN.

As shown above, our released dataset allows generating results with various classification models techniques. Through our analysis, we found that, while different models may yield different accuracy levels, the order of performance of individual layouts is very similar. Specifically, to understand our results' dependence on a particular classifier, we used F1 scores of all layouts with sensor count = 1 from the top-performing classifiers, namely KNN, Ridge, RF, and NN. Following that, we sorted the results alphabetically by IMU labels. Then, using a pairwise Spearman correlation (as used by Guzdial et al. [32] for comparing ranked lists), we obtained a correlation of 0.919, 0.975, and 0.919 with $p<0.001$ for RF vs. KNN, NN, and Ridge, respectively.

In addition, we conducted a similar analysis to understand the change in the ranking of IMUs for different sets of features. We selected five features (maximum, minimum, mean, skewness, and kurtosis) used in the existing literature on IMU sensing [28] and trained 17 models with RF. Subsequently, similar to the analysis comparing different classifiers, we calculated the Spearman correlation on the F1 score of alphabetically-sorted IMU's list from both feature sets. Our results show a high correlation of 0.995 with $p<0.001$ between the layout ranking produced by 2 different set of features, indicating that while selecting other features may result in a different F1 score, the order of IMUs remains very similar.

4.2 Identifying Sparse Layouts for a Given IMU Count

The large count of IMUs offers the possibility of creating vast layout combinations. However, not every count and layout may produce a similar recognition performance. Therefore, an important aspect that we examined was identifying the best performing sparse layout for a given number of IMUs. This analysis provides three major insights: Firstly, it allows us to understand how the recognition performance varies with the number of IMUs. Secondly, it gives insights into the interval in which F1 scores fall for any given number of IMUs. Lastly, the results inform the optimal IMU placement location with a fixed budget of sensors [10]. Of note, we use the term *IMU Count* to refer to any given amount of IMUs from 1–17.

Method. To explore the full combinatorial space, we trained models with all possible layouts from 1 to 17 IMUs on our initial train-test split as described in Section 4.1. Moreover, to systematically understand the variation in performance for both types of microgestures, we performed this analysis for three conditions: Freehand, Grasping, and Both Combined. This totals to $3 \times (2^{17} - 1) = 393{,}213$ models. For each model, we performed multi-class classification with 19 classes: (3 fingers × 6 gestures) + 1 Static hold. Note, Grasping and Both Combined conditions utilized grasp-independent models; therefore, we did not encode grasp information in the class labels. In Section 4.6, we compare our results with grasp-dependent models.

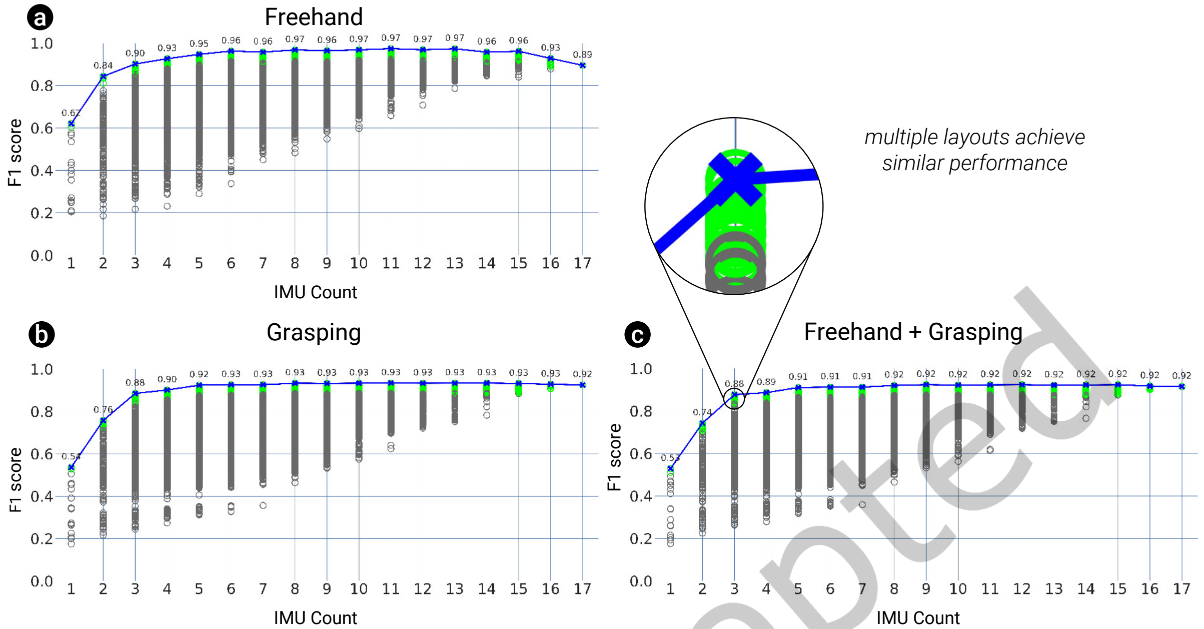


Fig. 6. **Full Combinatorial Results**: Each circle represents the F1 score for each of the 393K models classifying 19 classes in Freehand, Grasping, and Both Combined (Freehand+Grasping) conditions. The blue shows the maximum F1 score, and the green depicts the top 5% layouts in a particular IMU count.

Results. Figure 6 plots the F1 score on the test set from each 393K models trained in all three conditions (Freehand, Grasping, Both Combined), organized by the count of IMUs present in the model. We now discuss each condition in turn:

(1) Freehand microgestures: The results provide a complete overview of the large performance difference that depend on the IMU count and, for a given IMU count, on the specific location of IMUs comprised in a model. As shown in Figure 6-a, the highest F1 score for count = 1 is 0.62 $_{\text{(M-midd)}}$. Adding a second IMU increases the F1 score to 0.84 $_{\text{(T-midd, M-dist)}}$; the F1 score further increases to 0.90 $_{\text{(T-midd, I-prox, M-midd)}}$ and 0.93 $_{\text{(T-midd, I-prox, M-dist, R-prox)}}$ with 3 and 4 IMUs, respectively. On the contrary, the lowest F1 score for count = 1 was 0.2 $_{\text{(Forearm)}}$, and for count = 2 was 0.19 $_{\text{(R-prox, Forearm)}}$. Amongst all models, the maximum F1 score of 0.97 $_{\text{(T-prox, I-dist, I-prox, M-dist, M-midd, R-midd, P-midd, Forearm)}}$ is achieved with count = 8. It should also be noted that a F1 score of 0.90 can be achieved with as little as 3 IMUs, and henceforth only a maximum increase of 4% occurs with the addition of more IMUs. The F1 score drops to 0.89 when all 17 IMUs are included. To further investigate this drop, we trained 100 classifiers with random states from 0-99 for count = 17. We only change the seed values for this investigation, while training classifiers for other analyses have a constant seed value with default parameters to allow reproducible results. Out of 100 models, 4 models achieved the maximum F1 score of 0.96, which is close to the maximum F1 score of 0.97 achieved by some other higher counts. Overall, 93 out of 100 models achieved an F1 score of greater or equal to 0.90, and only 7 models have an F1 score in the range of 0.88 (lowest) and 0.89. This explains the reason for the drop we observed at count = 17.

(2) Grasping microgestures: Here, our classification setting is more challenging than Freehand microgestures due to the inclusion of all 12 Grasp variations. This results in a slight drop in overall performance (see Figure 6-b). For count = 1, the highest F1 score was 0.54 $_{\text{(I-midd)}}$. Adding an additional IMU (count = 2) gradually increased the performance to 0.72 $_{\text{(I-prox, M-midd)}}$, for count = 3 to 0.88 $_{\text{(T-dist, I-prox, M-prox)}}$, and for count = 4 to 0.90 $_{\text{(T-dist, I-midd, I-prox, M-prox)}}$. Similar to Freehand, the IMU located on the forearm achieved the lowest F1 score of 0.17 for count = 1. Across all models, the maximum F1 Score of 0.93 $_{\text{(T-dist, I-dist, I-prox, M-dist, M-prox, Handback)}}$ is first achieved at count = 6. Note, the general pattern of variation in the maximum and minimum F1 score is similar to the Freehand condition, and an F1 score of 90% can be observed with a small number of IMUs (count = 4). Afterwards, the maximum increment in F1 score is only 3%.

(3) Both Combined microgestures: As shown in Figure 6-c, we observed a similar overall trend when gestures in Freehand and all Grasp variations were classified together. The maximum performance achieved with one IMU was 0.53 $_{\text{(I-midd)}}$. Adding more IMUs resulted in an increase of F1 score to 0.74 $_{\text{(I-prox, M-midd)}}$, 0.88 $_{\text{(T-dist, I-prox, M-prox)}}$ and 0.89 $_{\text{(T-dist, I-prox, M-midd, M-prox)}}$ for IMU count = 2, 3 and 4 respectively. Conversely, the minimum F1 score for counts = 1, 2, 3 and 4 is 0.18 $_{\text{(Forearm)}}$, 0.23 $_{\text{(P-dist, P-midd)}}$, 0.26 $_{\text{(P-dist, P-prox, Forearm)}}$, 0.28 $_{\text{(P-dist, P-midd, P-prox, Forearm)}}$ respectively. The min and max difference of the F1 score within each IMU count shows a similar pattern as the other two conditions. Across all counts, the maximum F1 score of 0.92 $_{\text{(T-dist, T-midd, I-dist, I-prox, M-dist, M-midd, M-prox, R-dist)}}$ is first achieved with count = 8. At count = 5, an F1 score of 91% is obtained, and only a 1% increase is seen with more IMUs.

4.2.1 *Relevance of each IMU.*

Multiple layouts may achieve a performance close to the top-most layout in each count as shown in Figure 6. To better understand what locations on the hand and finger are more likely to contribute to top-scoring layouts, we analyzed the top 5% best-scoring layouts (marked in green color in Figure 6). Specifically, we introduce an *Occurrence Score* metric that quantifies the occurrences of each IMU in the top 5% layouts (see Eq. 1). Here, a higher score of an IMU indicates its frequent presence in the top layouts. For a set I of possible IMUs, the Occurrence Score of an IMU i is

$$occ_i = \frac{1}{|I|} \sum_{k=1}^{|I|} \frac{\text{occurrences of IMU } i \text{ in top 5\% layouts with } k \text{ sensors}}{\text{number of top 5\% layouts with } k \text{ sensors}} \tag{1}$$

where we calculate the mean of an individual IMU's occurrence over all IMU counts. It is important to note that this is not the overall occurrence in the total space of 393K models but rather how frequently it occurs in the top layouts.

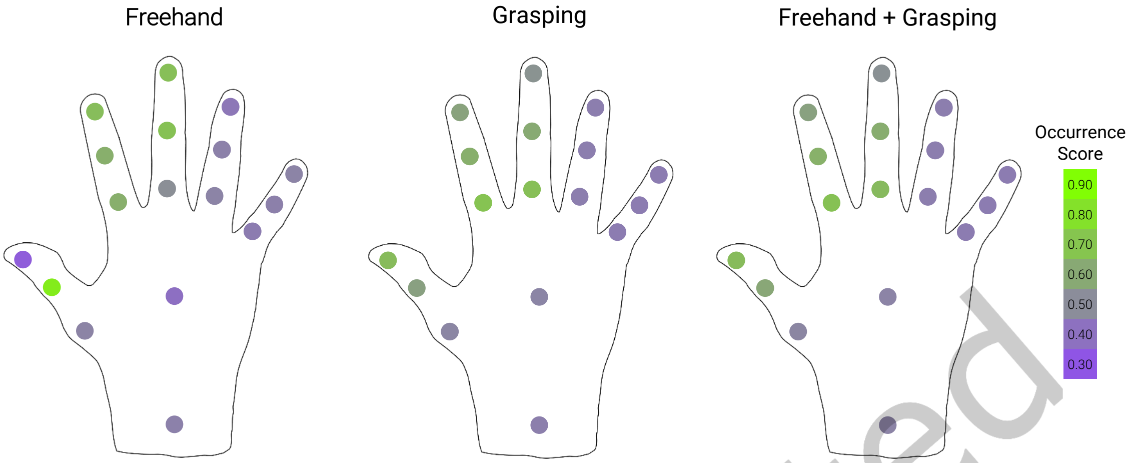


Fig. 7. Occurrence Score of each IMU in the top 5% layouts from count 1 to 17. Across all IMUs, we observed a minimum score was 0.33 and maximum of 0.84.

Results: We examined the Occurrence Score of each IMU as shown in Figure 7 and derived patterns that guide our further analysis. Since the gestures were performed by Thumb, Index, and Middle fingers, the IMUs from these three fingers appear more often in the top 5% layouts in all three conditions (Freehand, Grasping, and Both Combined). Interestingly, the Occurrence Score varies greatly across different segments of the same finger. The comparison between Freehand and Grasping conditions revealed three considerable differences: First, we observe that an IMU placed on the tip of the Thumb (T-dist) has a high Occurrence Score of 0.67 for Grasping microgestures, whereas it is only 0.33 for Freehand microgestures. We assume this is related to the nature of gestures performed on the palm in the Freehand condition, wherein the Thumb stretches out at a larger distance and bends lesser than during Grasping microgestures. In a typical grasp, the Thumb supports the object; hence the distance to reach the surface for performing a Grasping microgesture is relatively smaller. Second, for all fingers except the Thumb, Grasping microgestures tend to favor IMU placement on the proximal segment over the fingertip. In contrast, Freehand microgestures show a clear tendency to favor placement on the fingertip for Index and Middle fingers. Below, we investigate the effect of IMU position on classification performance in more detail.

Implications: For all three conditions, we noticed that a higher IMU count does not necessarily translate to higher recognition performance. F1 scores close to the optimal can be achieved already with a fairly small number of IMUs (3 to 6). We observed a large variation in performance depending on where a given number of IMUs is placed on the hand and fingers, which also depends on the microgesture condition as shown in Figure 7. These findings highlight the importance of creating a layout by choosing a right number of IMUs, a right combination of fingers, and finger segments for the desired set of grasp and microgestures to achieve optimal recognition accuracy.

4.3 Performance of IMU Placement at Segment Level

Having identified that the choice of finger segments for IMU placement can be crucial for obtaining high recognition performance, we now aim to investigate the influence of finger segments on recognition performance more systematically. This also informs the design of minimal form-factor devices that place IMUs only at the optimal segment.

Method. We used our initial 80:20 train-test split of the participants' data and evaluated using a single IMU under multiple settings. To reduce any effects caused by different grasp variations, we created grasp-dependent models. Moreover, for a clear understanding of individual fingers and their respective gestures, we performed finger-wise classification, i.e., atmost six gestures and one static hold class per finger. Overall, we trained 17 single-IMU layouts × [(1 Freehand × 3 gesturing fingers) + (9 Grasp variations × 3 gesturing fingers) + (3 Grasp variations × 1 gesturing finger)] = 561 models. For the analysis in this section, we focus on the IMU on gesturing fingers and on three representative grasp variations that have been identified in prior work to each represent a cluster of Grasping microgestures [83]. The detailed results, including IMUs on non-gesturing fingers and all 12 grasp variations will be released with our dataset.

Results. As illustrated by Figures 8 and 9, the F1 score varies greatly across different segments for Freehand as well as Grasping microgestures. In particular, it indicates that for some cases, the F1 score for a gesture may even rise from 0.0 to 1.0 depending on what segment the IMU is placed on the same finger. In the following, we highlight this effect for Freehand as well as Grasping microgestures.

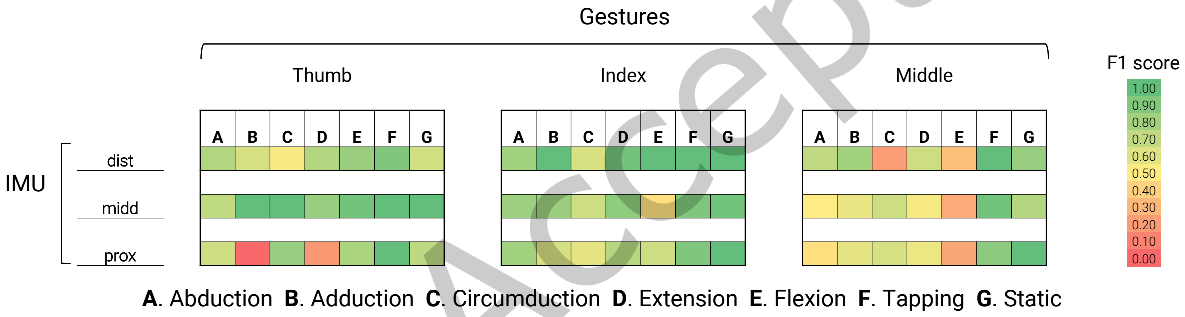


Fig. 8. F1 score of single IMU models trained for multi-class classification. The classes include six different gesture types possible with each finger (+1 static) for each model during Freehand microgestures. Note that different models were trained with IMU on each segment (distal, middle, proximal) and for different gesturing fingers.

(1) Freehand: The kinematics for each finger varies, and the motion required for each gesture is also different. As a result, the F1 score can have a large difference across segments (shown in Figure 8). We observed that the optimal segment is different for different fingers. In particular, for Thumb gestures, the middle segment (midd) achieved an average F1 score of 0.93, whereas the other two segments, i.e., distal (dist) and proximal (prox), have a relatively lower score of 0.72 and 0.60, respectively. The optimal segment for Index gestures is different: here, the prox-segment has an average F1 score of 0.91, while the performance on the other two segments is considerably lower with 0.78 (I-midd) and 0.76 (I-prox). For the Middle gestures, all segments achieved a similar F1 score of 0.60-0.65, the segment choice is still prominent for individual gestures wherein the performance may differ with 20-40% for Adduction, Abduction, and Circumduction. In contrast, the performance difference across segments is lower for the Tap gesture (10–13%). Surprisingly, due to the hand bio-mechanics, the IMU on the Handback can detect Thumb Flexion and Tap with an F1 score of 0.82 and 0.70, respectively. This finding can be beneficial to detect finger gestures in settings where a user might not want to wear any sensor on the finger (e.g., while working in a kitchen or car workshop).

We investigate this aspect of recognizing gestures from a non-gesturing finger in more detail in the next section.

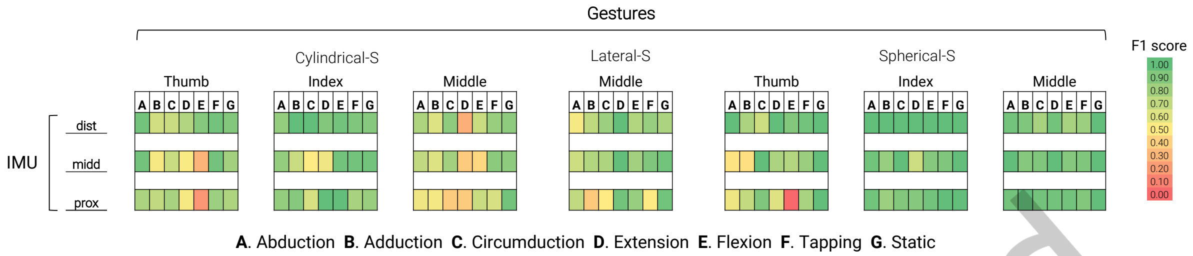


Fig. 9. F1 score of single IMU models trained for multi-class classification. The classes include 6 different gesture types possible with each finger (+1 static) during three exemplary grasp variations (Grasping microgestures).

(2) Grasping: Our results reveal a strong influence of segment choice for Grasping microgestures (see Figure 9). Similar to the Freehand condition, we observed a large difference in F1 score across different segments of the same finger. Furthermore, it is noteworthy that there are dissimilarities in the pattern of optimal segment across different grasp variations. This relates to the distinctive finger postures in different grasps, affecting how a finger moves while performing the gesture. In particular, for the Thumb and Index gestures on Cylindrical-S and Spherical-S, the dist segment appeared as the optimal segment in both grasp variations. However, for the Middle finger gestures, the optimal segment is different across all three grasp variations (Cylindrical-S has dist, Lateral-S has mid, and Spherical-S has prox). Moreover, the Index and Middle gestures on Spherical-S have a relatively lower variance across segments, which could be explained by the bigger real estate that affords comparatively larger movements than the other two grasp variations. In general, the substantial difference in the recognition performance at the segment level is due to the intricacies of the grasp variation, finger, and gesture.

Implications. Depending on the grasp, finger and type of movement during the gesture, the single-IMU performance across segments greatly varies. This formally validates our initial findings from the full combinatorial classification results: The choice of finger segment for the IMU sensor placement can have a very strong influence on classification performance. However, since these classification results differ based on the subset of grasps and chosen gesture classes, a one-fits-all design solution will likely not lead to best results. Hence, we propose a computational design tool in Section 6, which provides layout recommendations based on the user-defined parameters.

4.4 Placing IMU on a Non-gesturing Finger

Finger co-activation is a widely known phenomenon in bio-mechanics [78]. Our goal is to leverage finger co-activation and investigate if micro-movements caused in neighboring fingers are sufficient for gesture detection from a non-gesturing finger. This would be beneficial in situations where placement of an IMU on the gesturing finger would hinder the primary activity–e.g., having an IMU on the Index finger may hinder situations like using a knife. In such scenarios, placing the IMU on an alternative location capable of detecting gestures from a neighboring finger would be more desirable.

Method: To investigate the possibility of detecting gestures with any single finger, we used our initial 80:20 train-test split and trained five models for each of the three gesturing fingers; each model comprised a total of three

IMUs placed on every segment of the respective finger. For a detailed analysis, we performed grasp-dependent and finger-wise classification. This gives a total of 5 fingers w/ IMUs × 3 gesturing fingers = 15 models for Freehand. We trained another 150 models [(5 fingers w/ IMUs × 9 grasp variations × 3 gesturing fingers) + (5 fingers fingers w/ IMUs × 3 grasp variations × 1 gesturing finger]. In each multi-class model, we included all six gestures for an individual finger and the static class - totaling up to seven classes.

Results. Figure 10 and 11 show the F1 score on the test set for Freehand and Grasping when models are trained with IMUs on different fingers. These results indicate the feasibility of detecting gestures from IMUs on the non-gesturing finger:

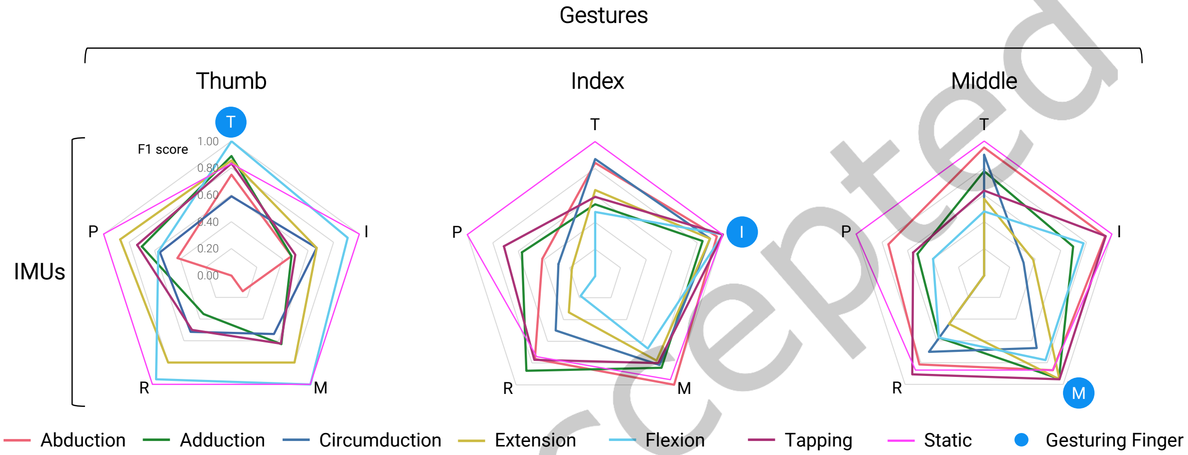


Fig. 10. F1 score of IMUs placed on gesturing as well as non-gesturing fingers for multi-class classification. The classes include six different gesture types possible with each finger (+1 static) for each model during Freehand microgestures. T, I, M, R, and P refer to the IMUs on Thumb, Index, Middle, Ring, and Pinky finger. The gesturing finger is denoted with a blue circle.

(1) Freehand: We observed the effect of finger co-activation and the feasibility of detecting gestures from IMUs on a non-gesturing finger for all three gesturing fingers (see Figure 10). Unsurprisingly, placing an IMU on the gesturing finger results in a higher F1 score in most cases. However, it is important to note that depending on the finger and gesture, the IMUs on a non-gesturing finger can even yield a higher F1 score than when placed on the gesturing finger. This is particularly visible with gestures performed by the Middle finger. This observation is in line with findings from prior work that have reported the middle finger to induce higher involuntary movement in adjacent fingers [78, 86]. For Middle Circumduction, for instance, the F1 score on a non-gesturing finger (Thumb) increases by 34% (from 0.67 to 1.00) compared to placing an IMU on the gesturing finger (Middle). This can be explained by the involuntary Thumb movement caused while performing the Middle Circumduction on the palm. Also, Index Adduction achieved a 5% higher F1 score through placing IMUs on a non-gesturing finger (Middle) than gesturing finger. Even though Thumb has the least tendency amongst all the fingers to induce movements in the neighboring fingers, placing an IMU on the non-gesturing finger (Middle or Ring) produces a similar F1 score as that on the gesturing finger (Thumb) for Flexion, Extension and Circumduction. These promising results of placing an IMU on the non-gesturing fingers show the feasibility of detecting gestures beyond the conventional placement strategies.

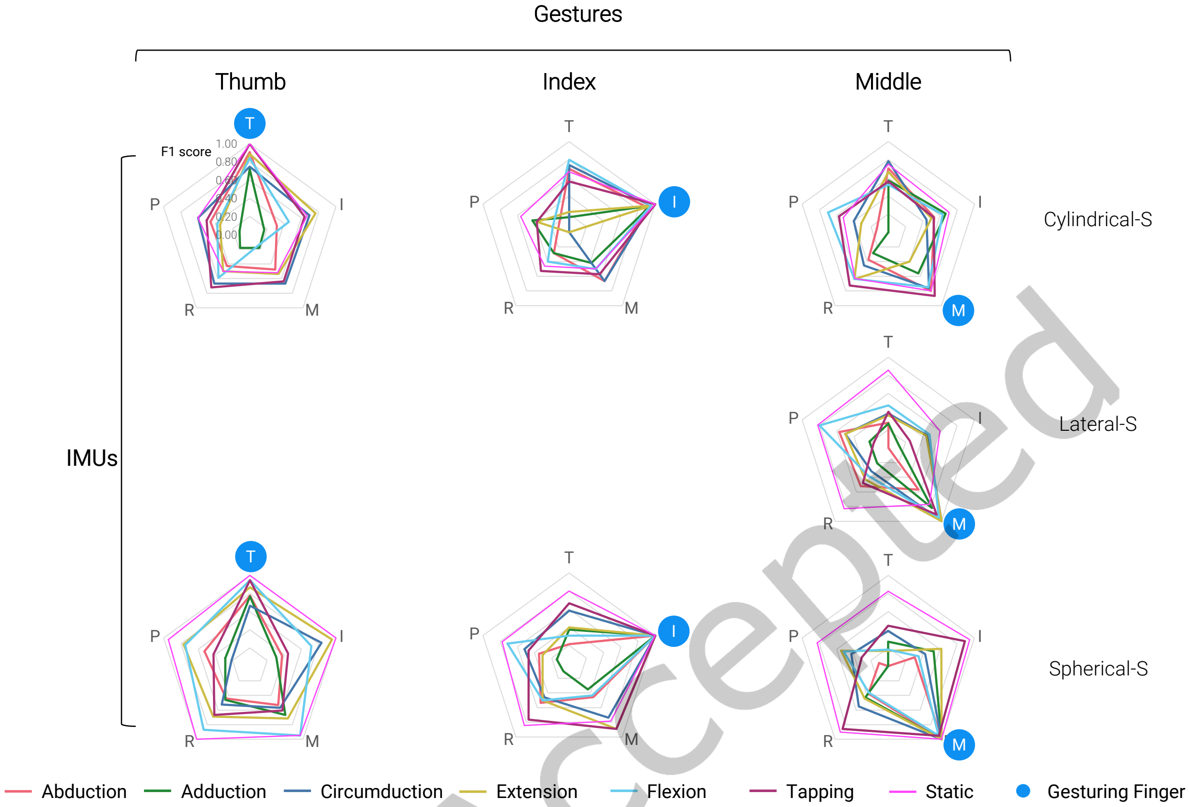


Fig. 11. F1 score performance of six different gesture types possible with each finger (+1 static) when the IMUs are placed on gesturing as well as non-gesuring fingers for three representative Grasp variations (Grasping microgestures).

(2) Grasping: As mentioned in prior work, fingers in contact with the object get support, thereby reducing the effect of co-activation [82]. Thus, all Thumb and Index gestures on Cylindrical-S (Knife) achieved the highest performance when the IMUs are placed on the gesturing finger. In spite of that, we observed that the non-gesturing finger can detect Thumb and Index gestures with a drop of only 15–20% from the F1 score obtained by an IMU on the gesturing finger. While this reduction is considerable, it may be acceptable for some gestures in settings that do not allow for augmenting the gesturing finger with IMUs. Based on the grasp type and gesture, the IMUs on the non-gesturing finger may even achieve a higher performance than the gesturing fingers, e.g., on Spherical-S (Pestle), Thumb Extension and Circumduction achieved a higher F1 score of 0.83 and 0.95, respectively, through IMUs on the non-gesturing finger (Index). In contrast, the IMUs placed on the gesturing finger (Thumb) achieved a comparatively lower score of 0.67 and 0.87. On Cylindrical and Spherical grasps, all fingers are in close contact with object but not all grasp types have the same contact fingers. For example, while holding Lateral-S (Spoon), the Ring and Pinky fingers are suspended in the air, which causes an involuntary movement in the other adjacent non-gesturing finger. As a result, the gesturing (Middle) and non-gesturing (Pinky) finger IMUs achieve a similar F1 score for Middle Abduction and can also detect Middle Flexion with an F1 score of 0.80 (0.15 lower from the IMUs on the gesturing finger). Additionally, we observed the possibility of detecting gestures with non-gesturing

fingers that are in contact with the object. With these many different factors affecting the performance, it is challenging for a designer to place the sensor at an alternative location intuitively.

Implications When the hands are busy, instrumenting gesturing fingers might not be possible in all cases. For example, while writing, instrumenting fingers involved in gripping the pen might hinder the primary activity. In such scenarios, placing an IMU on neighboring fingers can be efficient. Our findings show that placing IMUs on a non-gesturing finger may enable gesture detection at a comparable or even higher performance rate.

4.5 Generalizability of Layouts across Participants

Next, we aim to understand the extent of inter-personal differences in recognition performance. This is a crucial question because there can be inter-personal variations in the way the microgestures are performed. If there is a large difference in classification results across participants, the design tool that we describe in later Section 6 would need to account for it while suggesting a sparse layout.

Method. A comprehensive Leave-one-person-out (LOPO) evaluation with 12 participants × 393,213 layouts = 4,718,556 models will approximately take 25 days of computation time on our 40-core machine. To circumvent this problem, we first identified the best layout according to the F1 score for a given count of IMUs on our 80:20 participants split from the combinatorial results obtained with the combined condition (Freehand+Grasping). Subsequently, we used these best layouts and trained 204 models (12 participants × 17 best layouts for the IMU Counts) for a LOPO evaluation.

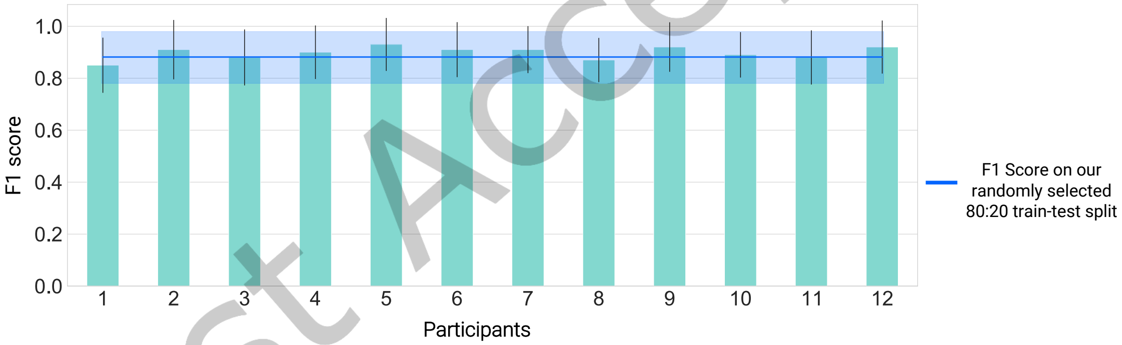


Fig. 12. Comparison of the F1 score achieved on our randomly selected two participants with leave-one-person-out. The blue horizontal line corresponds to the average F1 score across 17 IMUs for the previous 80:20 split, and the grey band shows the standard deviation in the F1 score across all IMU counts. The vertical columns represent the average F1 score for each participant, and the error bar represents the standard deviation for each participant from count 1 to 17 IMUs.

Results. Figure 12 depicts the results of the LOPO evaluation. We observe that the difference in F1 score from our randomly selected 80:20 train-test split and any LOPO model is about ±6%. It is worth noting that most participants achieved higher performance than our randomly chosen participants.

Implications. Despite the inter-personal variations in how the gestures are performed, our recognition pipeline still scales well and achieves high recognition performance with user-independent models. We observed only little variation in F1 scores across participants, which demonstrates that model predictions generalize to data from new users.

4.6 Grasp-Dependent v/s Grasp-Independent Models

In our combinatorial analysis, we trained grasp-independent classifiers by combining all grasp variations. Here, we aim to investigate if these initial results can be further improved if a subset of grasps is selected. This would be relevant for application cases that comprise selected activities with a known set of grasps, or for systems that can identify the current grasp, e.g., by using activity recognition.

Method. We classified all 12 grasp variations separately (grasp-dependent models) by using our initial 80:20 split of participants' data with 19 classes [(3 fingers × 6 gestures) + 1 static hold]. To save on the computation time, we performed the full combinatorial evaluation of grasp-dependent models until IMU count = 5. There were 12 grasp variations $\times \sum_{r=1}^{5} {}^{17}C_r$ layouts = 112, 812 models.

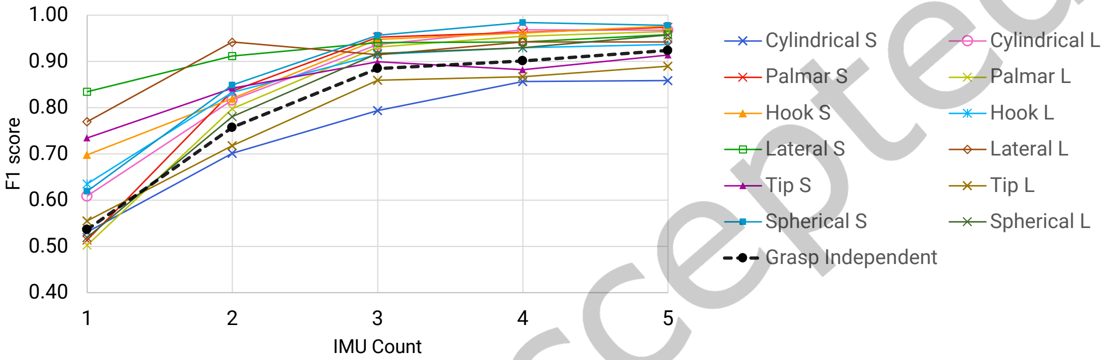


Fig. 13. Comparison of F1 score achieved by the best layouts until an IMU count = 5 for grasp-dependent and grasp-independent models.

Results. For 9 out of 12 grasp variations, the F1 score increased when the model is trained on a specific activity (see Fig. 13). Grasps like Lateral-S (Spoon), Tip-S (Needle), Lateral-L (Paper) showed an improvement in recognition of 20–30% compared to the grasp-independent model. In contrast, grasps like Cylindrical-S (Knife) and Tip-L (Pen) did not show any increment, which can be due to the object's geometry. Specifically, on such grasp variations, the fingers are tightly packed, hindering the finger movement while performing gestures.

Implications. The performance tends to improve if the model is trained for a specific grasp variation. Therefore, when a subset of grasp-variations are chosen that map to a specific context, our results from the combinatorial analysis can further improve. This feature of selecting grasps is also integrated in our later presented design tool for finding a sparse layout.

4.7 Summary of Findings

The key takeaways from the above in-depth analyses are:

- *More is not always better*: Saturation in classification performance is achieved after a fixed count of IMUs as shown in Figure 6. In typical cases, a quite low number of 3–4 IMUs suffices for an F1 score of about 90%.
- *Possible to achieve gesture recognition via IMU on non-gesturing finger*: Our findings from placing IMUs on a non-gesturing finger in Section 4.4 opens up a new avenue for microgesture detection in HCI by leveraging movement patterns caused by complex hand bio-mechanics in non-instrumented fingers.
- *Effect of grasp type*: In our analysis of Grasping microgestures, we found the F1 score pattern dissimilar across different grasp variations – due to the influence of grasps on the finger pose and motions. This ultimately affects the spatial configuration of an optimal layout.
- *User-independent models*: We found that a performance of 90% and above with user-independent classification models. This demonstrates the viability of utilizing IMU-based input in future consumer-grade systems.

Given this multi-factorial design space that influences the classification performance, providing an automated system to a designer will enable rapid design iterations and decision making for optimal IMU placement. Inspired by these findings, we present a rapid technique to identify sparse layouts and a GUI-based computational design tool in the following sections.

5 SPARSEIMU: METHOD FOR RAPID SELECTION OF SPARSE IMU LAYOUTS

Training the models for all layouts of IMUs took about 50 hours (Freehand = 1:27:31, Grasping = 22:41:52 and Freehand + Grasping = 26:20:10). Modifying the set of gestures or objects requires re-training of the models, as a new setting can influence the importance of specific IMUs. Additionally, if one wants to explore design variations, like comparing different gesture sets or sets of objects, this results in a multiplicative increase of the number of models that need to be trained and evaluated. This large computation time makes an exploratory study of IMU layouts very slow if not impossible.

To overcome this issue, we propose a method referred to as *SparseIMU*. It uses a proxy metric describing the importance of individual IMUs. As a requirement, this method should be fast to compute and correlate well with the results obtained from training all model layouts. Specifically, the proxy metric is used to derive what IMUs contribute most to the classification. In this work, we study two such proxy metrics:

- *Feature Importance*, also called Mean Decrease in Impurity [59], calculates how well a feature splits the trials into their corresponding classes. This is a natural choice for Random Forests, as the same criterion is used to build the trees themselves. Instead of training and evaluating separate models for each combination of IMUs, this approach requires training only one Random Forest model that comprises all 17 IMUs. Then Feature Importance, calculated from this model, indicates how much an individual feature is contributing. For each IMU, we use multiple features (mean, variance, etc.). Therefore, we aggregate the features belonging to the same IMU using summation to infer an individual IMU's importance. Here, the IMU with the highest importance score is essential for the classification, and the one with the lowest score contributes the least in the classification.
- *Permutation Importance* is a posthoc interpretation metric to calculate the importance of a feature. Here, a model that comprises all IMUs is trained and evaluated on the original dataset. For a specific feature, all the values in the test data are then randomly permutated; the feature, therefore, no longer provides useful information. The model is evaluated again on this corrupted dataset and the difference in performance between the original and the corrupted dataset is computed. The larger the drop in performance, the more important is the feature [9]. This approach needs no further training and only one additional evaluation for each feature. The importance of an IMU is again calculated by summing the importances of its features.

Both proxy metrices provide an importance score for each IMU. Given a desired IMU count k, one could simply choose the layout created from the top k IMUs, based on their importance score. However, in practice, it is beneficial to expand the search space of possible "top" layouts. In particular, we search through all possible combinations of the top t IMUs (based on importance) chosen k at a time (${}^{t}C_k$). We choose a t such that the total number of layouts possible with the top t IMUs (${}^{t}C_k$) is at least 1% (or 10% if $k <= 3$) of the total number of possible layouts for the given count (${}^{17}C_k$) and train all those (${}^{t}C_k$) models. For instance, if the desired IMU count is $k = 5$, we would choose $t = 9$, since (${}^{9}C_5 > 0.01 \times$ (${}^{17}C_5$) and thus we would train 126 models. Additionally, modifying this threshold of 1% allows for a user-defined trade-off between evaluation time and sparse layout performance.

5.1 Validation of SparseIMU Method with the Combinatorial Maximum

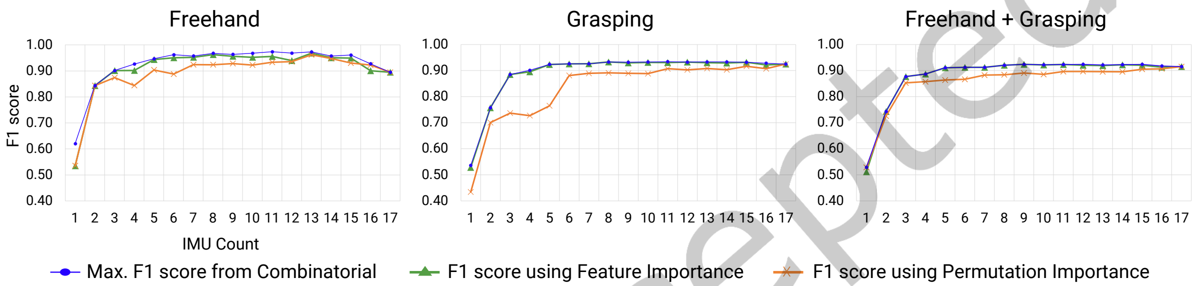


Fig. 14. Comparison between the F1 Score of layouts from the maximum combinatorial (see Fig. 6) and F1 score achieved by the layouts recommended from Feature and Permutation Importance.

To benchmark the selections generated from the two proxy metrics (Feature and Permutation Importance), we use the IMU layouts from our combinatorial results that achieved the maximum F1 score in Section 4.2. To quantify the differences, we obtain a Spearman's correlation (ρ) between the F1 score from the max. combinatorial layout and the layouts from the two metrics. Permutation Importance received $\rho = 0.7785$ for the Freehand, 0.6617 for the Grasping, and 0.8864 for the combined condition (all p<0.005). In contrast, Feature Importance received considerably higher correlations, with $\rho = 0.8630$, 0.9380, and 0.9419 for the respective conditions (p<0.005). The high correlation using Feature Importance is also visible in Figure 14, where the layouts consistently obtained an F1 score closer to the best performance in the combinatorial results. Therefore, we use this metric further to calculate the computation time.

Runtime. We now quantify the significant reduction of computation time required to select sparse layouts with the proposed SparseIMU method using Feature Importance. Given the 323K models needed to evaluate the entire combinatorial space, we used our institution's cluster system with a 40-core setup. Of note, this high-end configuration machine used in our combinatorial results is not widely accessible. In contrast, we evaluate our rapid method's performance on a commodity laptop (8-core MacBook Air). As shown in Figure 15, the time required to find the sparse layout by our method is significantly shorter, despite the use of a commodity laptop. This reduction is possible due to the considerably smaller number of model training required across each IMU count. For instance, if we were looking for a layout with $k = 5$ IMUs out of $n = 17$ possible IMUs in the Freehand condition, the time reduces from 3 mins on the compute cluster to 1 minute on a consumer-grade laptop. Moreover, for Grasping Microgestures and Both Combined conditions, it reduces from about 50 mins to 5 mins and from

1 hour to about 6 mins, respectively. While it takes longer to find solutions for IMU counts 7-11, we note that the method still performs significantly faster than the baseline. Moreover, we expect that layouts with this large number of IMUs need to be rarely considered, since going beyond 3–4 IMUs will only lead to a maximum increase of 4% in the F1 score, as we have shown above (see Figure 6). Overall, the reduction in time achieved by our method on a commodity laptop offers strong benefits for rapid iteration. In the next section, we use our method in a computational design tool.

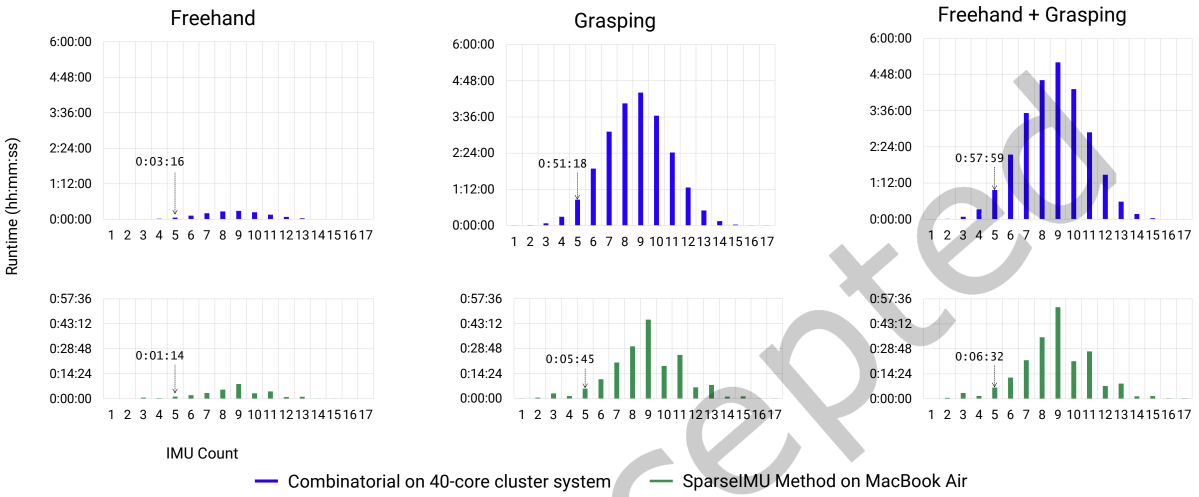
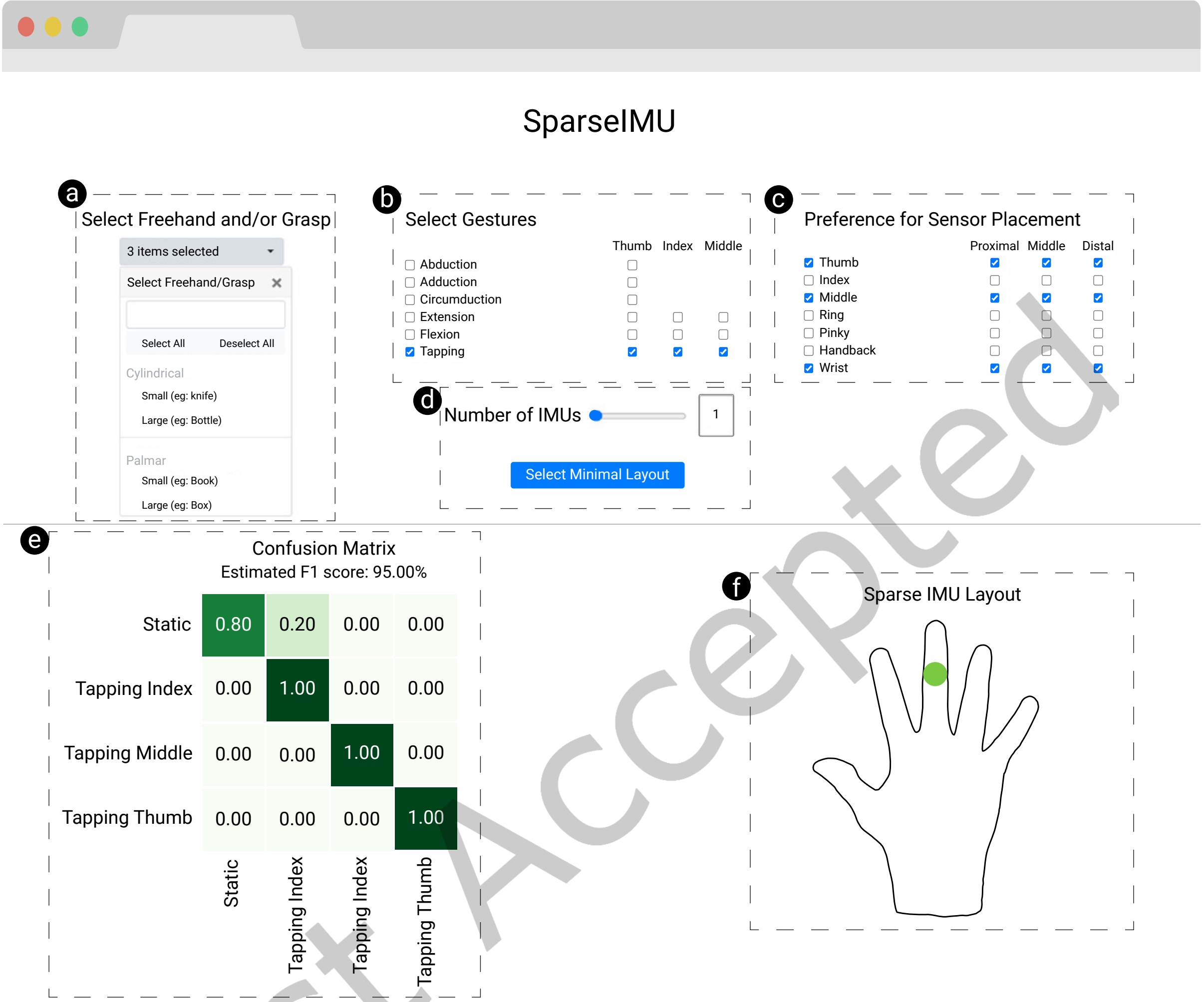


Fig. 15. Runtime comparison between SparseIMU method and the Combinatorial Search for all three conditions: Freehand, Grasping, and Both Combined.

6 COMPUTATIONAL DESIGN TOOL FOR RAPID SELECTION OF CUSTOM SPARSE LAYOUTS

Based on the SparseIMU method for selecting IMU layouts, we contribute a computational design tool. It assists designers in the following tasks:

- *Finding a sparse IMU layout that achieves high gesture recognition accuracy:* Using the designer's specifications, the tool selects optimal designs in near real time and indicates the expected recognition accuracy. This also allows the designer to quickly obtain an initial understanding of how well a desired set of microgestures can be recognized while the user is holding certain objects. The design tool assists designers in locating fingers and precisely locating the segment of the finger where the IMU should be placed.
- *Exploring location alternatives:* Considerations of ergonomic wearability or aspects inherent to certain application cases may restrict the space where IMUs can be deployed on the user's hand. For instance, a smart ring with an in-built IMU can be more suitably placed on the ring finger than the thumb. And an application case involving dexterous manipulation of objects may benefit from IMUs placed on the proximal phalanges, rather than close to the fingertips. The tool allows the designer to restrict what locations can be augmented with IMUs, and to quickly explore alternatives.
- *Finding gestures that perform well:* While it is understood that not all gestures are compatible and will have a high performance for a specific set of objects and constraints, one key functionality of the design tool is to provide a visual representation that depicts the performance of the individual gestures. This enables the designer to quickly inspect which gestures perform well and which do not, and choose the most compatible gestures that offer high recognition accuracy.



Select Freehand and/or Grasp

3 items selected

Select Freehand/Grasp ✕

Select All Deselect All

Cylindrical
Small (eg: knife)
Large (eg: Bottle)

Palmar
Small (eg: Book)
Large (eg: Box)

Select Gestures

	Thumb	Index	Middle
□ Abduction	□		
□ Adduction	□		
□ Circumduction	□		
□ Extension	□	□	□
□ Flexion	□	□	□
☑ Tapping	☑	☑	☑

Preference for Sensor Placement

	Proximal	Middle	Distal
☑ Thumb	☑	☑	☑
□ Index	□	□	□
☑ Middle	☑	☑	☑
□ Ring	□	□	□
□ Pinky	□	□	□
□ Handback	□	□	□
☑ Wrist	☑	☑	☑

Number of IMUs: 1

Confusion Matrix
Estimated F1 score: 95.00%

	Static	Tapping Index	Tapping Index	Tapping Thumb
Static	0.80	0.20	0.00	0.00
Tapping Index	0.00	1.00	0.00	0.00
Tapping Middle	0.00	0.00	1.00	0.00
Tapping Thumb	0.00	0.00	0.00	1.00

Fig. 16. Screenshot of the computational design tool for designing sparse IMU layouts. (a) User can select Freehand and/or multiple Grasp variations. (b) The tool automatically recommends possible gesture combinations with three fingers. (c) Additional constraints with respect to the placement of the IMUs can be specified. (d) The number of required IMUs can be selected and button click generates the results in form of (e), a confusion matrix showing the gesture-wise performance and an overall estimated F1 score, and (f), the location of the IMUs present in the sparse IMU layout.

A screenshot of the design tool is shown in Figure 16. The designer first selects Freehand and/or a set of Grasp variations(s) that the microgestures should be compatible with. Next, she selects the set of microgestures that shall be recognized and indicates which fingers are used for gesturing. Then, the designer can place additional constraints for IMU placement. Entire fingers or individual finger segments, as well as the back of the hand or wrist can be added or removed from the set of possible locations. As the last step, the designer selects the desired number of IMUs, to trade-off between a minimal or more complete instrumentation of the hand. With the click of a button, the IMU layout is then selected.

To visually present the recognition accuracy of chosen gestures, the tool displays a confusion matrix, along with location of the individual IMUs on the hand. If the designer is not satisfied with the Tool's recommendation, she can quickly explore options in an iterative manner. For instance, she may fine-tune the set of gestures or explore alternative locations for placing IMUs.

Implementation. It is noteworthy that our tool is different from a conventional lookup table which would require 17.5 trillions of entries to cover the various combinations of IMUs, subsets of gestures and grasp variations. Instead, by training only a few models using the SparseIMU method, our tool supports every possible custom user input while minimizing the computational complexity and storage. Furthermore, it allows the designer to rapidly iterate on multiple custom input options. Specifically, the tool uses the microgestures dataset and the SparseIMU method to identify the optimal IMU layout for a given set of requirements and constraints. The tool creates new classification models with our initial 80:20 split of train and test data. In addition to the required gestures, a Static hold is automatically added as a negative class. For generating the confusion matrix and an estimated accuracy, we use our test set. The Flask web framework for Python was used to create the tool's back-end. The front-end was styled using the Bootstrap toolkit, and JavaScript was used for client-side scripting. The Snap.svg JavaScript library was used to render the selected IMU layout.

#	Condition	Grasp Variations	Gestures	IMU Constraints	Required IMU Count	Max. F1 score from Combinatorial	F1 score from Tool
1	**Freehand**	Freehand	Thumb Adduction, Middle Abduction, Middle Circumduction, Thumb Tapping, Index Flexion, Middle Tapping, Middle Extension, Thumb Extension, Index Circumduction	anywhere but Thumb	3	0.90	0.88
2	**Freehand**	Freehand	Middle Abduction, Thumb Tapping, Middle Extension, Middle Circumduction	anywhere but Index	5	1.00	1.00
3	**Grasping**	Cylindrical-S, Cylindrical-L, Hook-L, Palmar-S, Spherical-S, Tip-L	Middle Extension, Middle Abduction, Index Tapping, Index Adduction, Index Abduction, Middle Circumduction, Middle Flexion, Middle Tapping, Thumb Adduction	anywhere but Middle	3	0.81	0.79
4	**Grasping**	Palmar-S, Tip-S	Middle Adduction, Middle Flexion, Middle Circumduction, Middle Tapping	anywhere but Ring	5	1.00	0.92
5	**Freehand + Grasping**	Freehand, Palmar-L, Hook-S, Cylindrical-S, Tip-L, Spherical-S, Spherical-L	Index Tapping, Thumb Adduction, Thumb Circumduction, Index Flexion, Middle Tapping, Thumb Abduction, Thumb Extension, Middle Extension, Thumb Tapping	anywhere but Pinky	3	0.95	0.95
6	**Freehand + Grasping**	Freehand, Cylindrical-S, Palmar-L	Thumb Adduction, Index Circumduction, Middle Tapping, Thumb Tapping	anywhere	5	1.00	1.00

Table 1. Comparison of maximum F1 score from Combinatorial Search and Tool Output for six example cases. It includes the randomly selected grasp variations, gestures, user-defined constraints, and required IMU count. For the classification, we also had a negative class (Static hold) in each case.

6.1 Tool Evaluation

In addition to the validation of the SparseIMU method in section 5.1, we performed another benchmarking to compare the tool's output with the combinatorial results when the designer applies constraints and opts for choosing a subset of grasp variation and gestures. Therefore we created six example cases covering all three conditions. We randomly selected grasp variations, gestures and added finger-wise placement constraints.

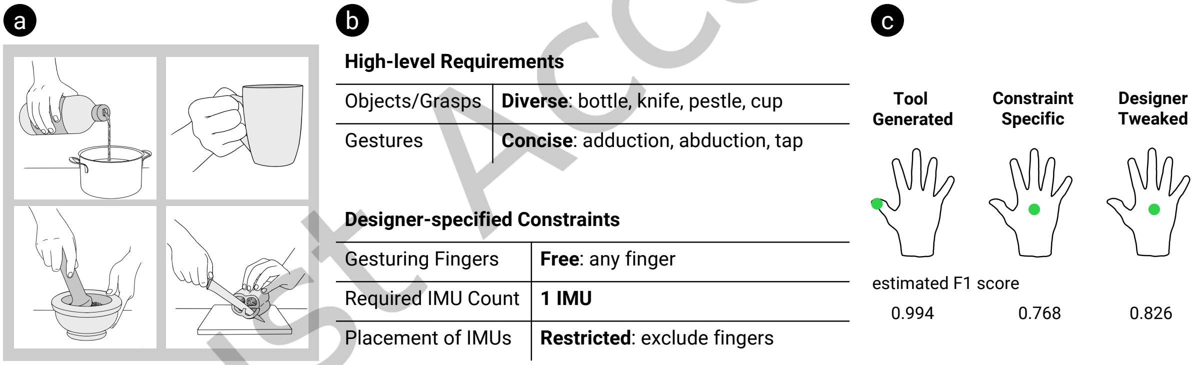
Informed by results from the first validation study, we chose two variations of IMU counts that we consider particularly promising for applications: 3 IMUs for a good recognition performance with very good wearability due to the low number of IMUs; and 5 IMUs for further increased recognition performance with a level of wearability that is still acceptable in many applications. We compared our tool's estimation by creating new combinatorial results for each case.

Results. Table 1 lists the example cases along with the results. In five out of six cases, the tool selected layouts that achieved an F1 score that was as high as the best performing combinatorial result or a maximum of 2% lower. The largest difference of 8% occurred in case 4, wherein the tool selected a layout with an F1 score of 0.92, while the best performing combinatorial layout achieved a full 1.00. Noteworthy, the tool also performed well in case 3, in which most of the randomly selected gestures involve the Middle finger whereas the constraint was to exclude the Middle finger from placing IMUs. Despite this demanding constraint, the tool successfully selected a layout that achieves performance close to the layout found by exploring the entire combinatorial space.

7 APPLICATION SCENARIOS

In this section, we present a set of four scenarios, each illustrating a realistic application of freehand and grasping microgestures with different design requirements and constraints. We demonstrate how our computational design tool can assist designers in deciding between various layouts, which is a non-trivial problem potentially requiring a trade-off, and can help in refining IMU-based sensing solutions.

7.1 Kitchen: Supporting Diverse Objects with Minimal Instrumentation



High-level Requirements

Objects/Grasps	**Diverse**: bottle, knife, pestle, cup
Gestures	**Concise**: adduction, abduction, tap

Designer-specified Constraints

Gesturing Fingers	**Free**: any finger
Required IMU Count	**1 IMU**
Placement of IMUs	**Restricted**: exclude fingers

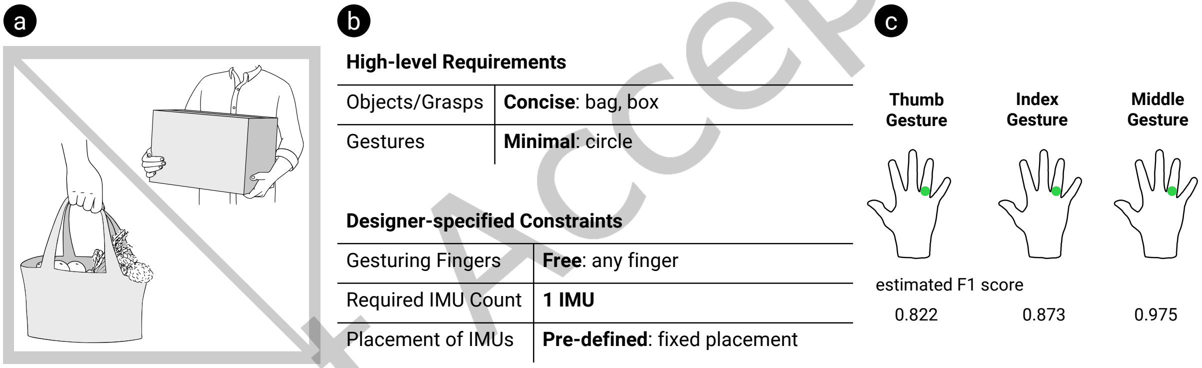
Fig. 17. Supporting diverse objects a) with minimal instrumentation b) in a smart kitchen scenario requires a trade-off between F1 score and IMU postion c).

Smart kitchens, providing in-situ instructions while cooking, have been a popular research area over the last decade [50]. We envision our computational design tool to support a designer, Alice, in the development of an in-situ recipe manager that supports information access using microgestures while cooking. For her first prototype, Alice wants to enable microgestures on *four objects* commonly found in the kitchen: knife, bottle, cup and pestle (cf., Fig. 17-a). For browsing a recipe, her application requires a *small, concise set of gestures*: back (abduction), forward (adduction) and select (tap). Due to frequent hand washing, the layout should be minimal (1 IMU) and restricted to the back of the hand or wrist (cf., Fig. 17-b).

Tool Output: With the selection of objects and gestures (and no further constraints imposed), the computational design tool suggests the thumb as common finger capable of performing all desired gestures, and the thumb's middle segment for IMU placement. Being 'most ideal', this sensor location achieves an F1 score of 99.4% (cf., Figure 17-c). However, Alice, excluded the fingers as sensor locations for sanitary reasons. This restrains sensor placement to the back of the hand and wrist, which achieve an F1 score of 76.8% and 56.6% respectively. For both, the confusion matrices reveal that the adduction gesture has a lower score, likely due to the large distance between the IMU and the gesturing finger. As a result, Alice settles on a trade-off between IMU location and available gestures. To keep the IMU position on the back-of-the-hand, she updates her design to include only tap and abduction gestures, increasing F1 score to 82.6%.

7.2 On-the-Go Interaction

7.2.1 Sensor Placement on Non-Gesturing Finger. As voice user interfaces are oftentimes prone to false activation [80], wake-gestures are an attractive remedy [73, 105, 106]. Bob aims to explore wake-gestures that work in on-the-go scenarios where both hands are occupied, e.g., while carrying two bags or a box (cf., Fig. 18-a). Furthermore, he intends to leverage an existing smart ring that he intends to 'hack' to access its IMU data. It does not matter which finger performs the gesture. However, ideally, the ring would keep its current position: worn on the ring finger's proximal segment.



High-level Requirements

Objects/Grasps	**Concise**: bag, box
Gestures	**Minimal**: circle

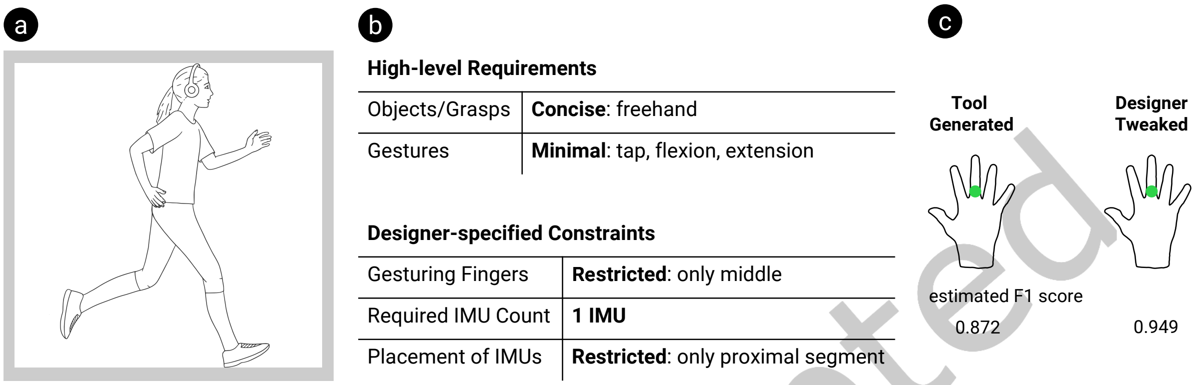
Designer-specified Constraints

Gesturing Fingers	**Free**: any finger
Required IMU Count	**1 IMU**
Placement of IMUs	**Pre-defined**: fixed placement

Fig. 18. An on-the-go scenario a) with pre-defined sensor placement on a non-gesturing finger b) leverages co-activation c).

Tool Output: Bob starts by evaluating the circle gesture performed with the thumb and the IMU present on the ring finger. The tool outputs an F1 score estimate of 82.2%. As wake-gestures should be resilient to false activation, Bob is not satisfied yet and explores further possibilities. As the position of the IMU is non-negotiable, he includes index and middle as gesturing fingers which achieve an F1 score of 87.3% and 97.5% respectively. The middle finger's promising performance (97.5%) is explained with the higher co-activation sensed on the ring finger (where the sensor is worn). Here, the computational design Tool allowed Bob to iteratively explore the gesture space and finally arrive at a tailored solution.

7.2.2 Finding Unambiguous Combination of Gestures. Listening to music while running is a typical combination, but controlling the music app on a smartphone or smartwatch's touchscreen requires Taylor, a frequent runner, to take unplanned breaks as shown in Figure 19-a. Conventionally, she needs to pause her run for performing the desired command (switch tracks or play/pause). These frequent and unnecessary halts for simple inputs affect her lap timings. She would prefer to use her middle finger for gesturing since she keeps switching the index and thumb poses in different fist forms while running. Her requirements are only for three gestures, including Tap, Flexion, and Extension. Also, due to vigorous hand movements and to keep the IMU firmly attached to her finger, she chooses to place the IMU ring in the proximal segment, which can be on any finger (see Figure 19-b).



High-level Requirements

Objects/Grasps	**Concise**: freehand
Gestures	**Minimal**: tap, flexion, extension

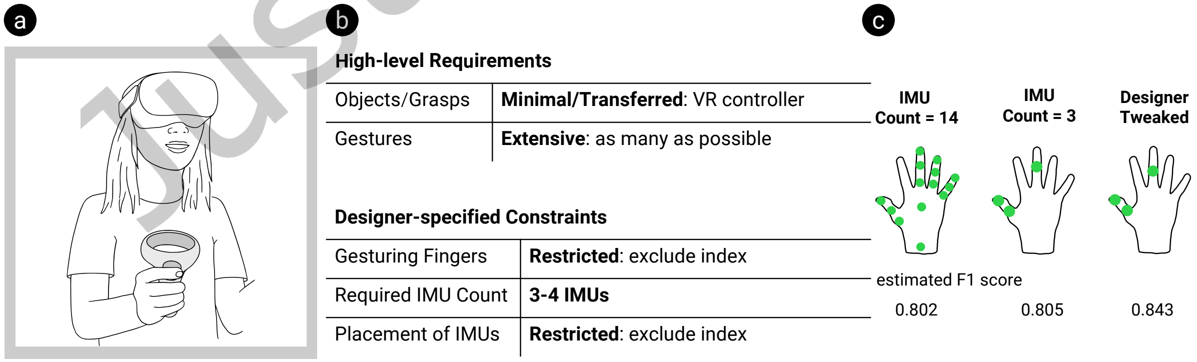
Designer-specified Constraints

Gesturing Fingers	**Restricted**: only middle
Required IMU Count	**1 IMU**
Placement of IMUs	**Restricted**: only proximal segment

Fig. 19. Supporting Freehand a) with minimal but clearly distinguishable gesture set b) in a running scenario with a restricted placement choice c).

Tool Output: Taylor started by opting for Freehand gesture and then made her gesture choices, and selected all fingers' proximal segment. As one's intuition, the tool suggested placing the IMU on the Middle Finger's proximal segment. It predicts an estimated score of 87.2%. By analyzing the confusion matrix, Taylor found out Flexion and Tap gestures get confused and subsequently decided to find the performance of other gestures. Using the rapid evaluation provided by the tool, she found out that replacing Flexion with Abduction solves this issue, and an estimated F1 score of 95% is possible (see Figure 19-c). Here, the tool was beneficial in finding an alternative gesture that can be detected at a higher performance while preserving all the other requirements.

7.3 VR Controller: Diverse Gestures with Minimal IMUs



High-level Requirements

Objects/Grasps	**Minimal/Transferred**: VR controller
Gestures	**Extensive**: as many as possible

Designer-specified Constraints

Gesturing Fingers	**Restricted**: exclude index
Required IMU Count	**3-4 IMUs**
Placement of IMUs	**Restricted**: exclude index

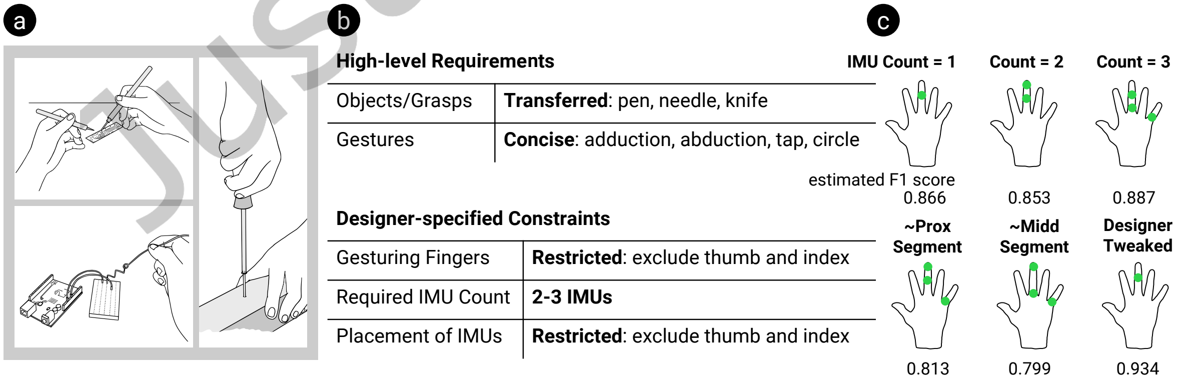
Fig. 20. Minimal setup with 3–4 IMUs a) with maximum diverse set of gestures b) finding the balance between gestures and accuracy.

Exploring diverse gestural inputs for VR [48] has been a popular area for experimentation in HCI and media arts. Dan plans a VR media arts installation which uses microgestures on a hand-held VR controller to contrast private and public interactions by subtly expanding the controller's range of functions. Thus, as demonstrated by [38], he aims for a miniaturized device equipped with 3–4 IMUs in combination of a commodity VR controller. He wants to avoid placing IMUs on the index finger which operates the VR contoller's push button and also not use it as a gesturing finger. To facilitate playful public or private interactions, he hopes to support as many different gestures as possible.

Tool Output: Dan explores the solution space for all possible IMU locations excluding the index finger (14 IMUs total). The tool yields an F1 score of 80.2% if 12 gestures are supported. Dan iteratively decreases the IMU count (while keeping the amount of gestures to 12) inspecting performance after each decrement. He identifies a saturation in F1 score at 3 IMUs (80.5%), which illustrates that a higher number of IMUs does not necessarily imply better performance (cf., Fig. 20-c). After further tweaking their configuration, Dan settles on a 3-IMU configuration and a set of 10 gestures. This choice is a trade-off allowing for a relatively high amount of gestures while still achieving an F1 score of 84.3%. As Dan aims for a rather playful, explorative VR installation, he considers this level of score acceptable. This highlights how the choice of a final layout depends on the weight the designer assigns to the different parameters (e.g., amount of gestures vs. performance) which in turn strongly relate to the specific application (e.g., playful vs. safety-critical purposes).

7.4 Electronics Workshop: Microgestures while Performing High-Precision Tasks

Carla seeks to explore how users can make use of microgestures to access additional instructions during high-precision tasks such as soldering. She envisions tools such as a soldering iron, soldering lead, or a screwdriver (cf., Fig. 21-a). As these tools are not available in our dataset, she uses our computational design tool to make an informed best guess by determining a set of initial layouts to elaborate on. Here, our Tool draws strength from the similarity in grasp types: the soldering iron (not present in the dataset) is typically held in a fashion similar to the pen (present in the dataset); holding fine soldering lead or wire in place resembles holding a needle, and holding a screwdriver demonstrates a similar (cylindrical) grasp like holding a knife. Carla envisions four gestures: forward, backward, select, and circle which she intends to use to browse an instruction manual. She furthermore excludes thumb and index finger–both as gesturing fingers and for IMU placement–to not interfere with the high-precision soldering task, and constrains the number of IMUs to 2 or 3 (cf., Fig. 21-b).



High-level Requirements

Objects/Grasps	**Transferred**: pen, needle, knife
Gestures	**Concise**: adduction, abduction, tap, circle

Designer-specified Constraints

Gesturing Fingers	**Restricted**: exclude thumb and index
Required IMU Count	**2-3 IMUs**
Placement of IMUs	**Restricted**: exclude thumb and index

Fig. 21. Transfer of grasps a) with restrictions on Thumb and Index b) finding the optimal finger segment c).

Tool Output: The computational design tool suggests placing the IMUs on the middle finger which achieves a competitive F1 score of 88.7% when 3 IMUs are used. Yet, at closer inspection, the tool also reveals that accuracy varies depending on the finger segment on which the IMU is placed, ranging from 80% to 88%. Hence, the choice of finger segment is crucial. Moreover, the tool shows that there is only 2% gain in score from placing 3 IMUs on the middle and pinky finger (88.7%), compared to only one IMU on its middle segment. Thus, a single IMU is sufficient to cover all gestures Carla had planned for her scenario. Further exploration shows that an increase in accuracy can be obtained for the 1-IMU layout to 93.4% by removing the adduction gesture (cf. Fig. 21-c). As follow up, Carla conducts a small-scale data collection using the 1-IMU layout recommended by the tool. Here, the tool provided a best guess in terms of IMU placement and gesture choice which served as a strong foundation for further iterations.

8 COMPARING THE TOOL'S OUTPUT WITH LIVE GESTURE RECOGNITION

To further demonstrate the tool's practical usefulness and generalizability to real-world applications, we collected another dataset with different hardware configurations and participants. This section compares the predicated F1 score from the computational tool with another system deployed for live gesture recognition.

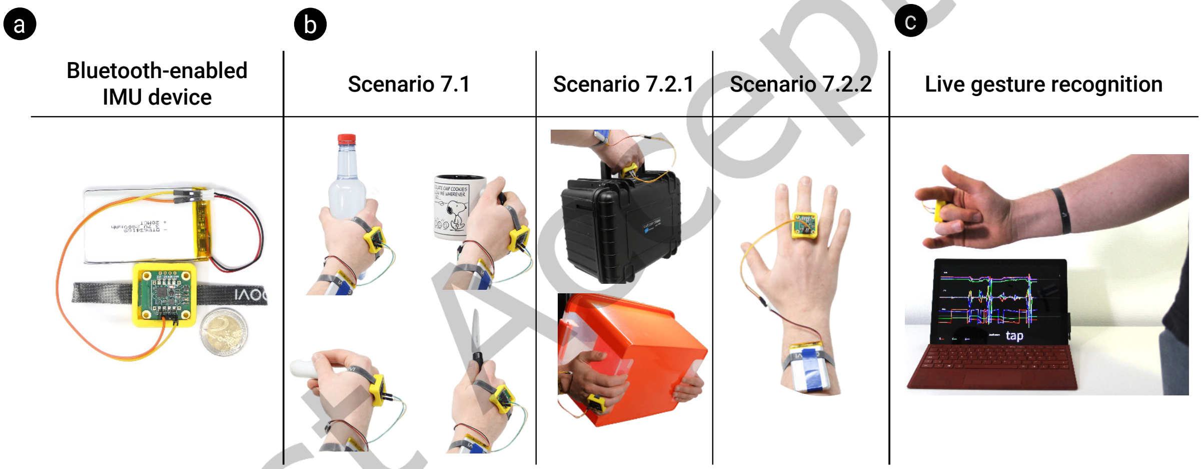


Fig. 22. Minimal wireless hardware with battery a); scenarios involving multiple objects and freehand b); live classification of gestures c).

Apparatus. With a focus on mobility and wearability, we developed a working wireless system that consists of a 9-Axis IMU (MPU9250, InvenSense Inc., CA, USA) and a Bluetooth module. As with previous work for gesture detection with a low-power wearable device [17], we sampled the accelerometer at 35 Hz (lower than in our microgestures dataset). Similarly, gyroscope and magnetometer were sampled at 35 Hz. For powering the device, we used a 2000mAh (DTP634169) lithium polymer battery. We also created a 3D printed casing with hooks to attach velcro straps so that the device can be easily worn on different fingers and varied hand sizes. An additional velcro strap and adhesive tape were used to affix the battery to the arm such that it would not interfere with hand actions. We created two such devices (as shown in Figure 22-a) and synchronized them to enable data collection from multiple hand segments simultaneously. Raw data from the devices is wirelessly streamed over Bluetooth to a PC for live classification.

Scenarios. To keep the data collection feasible, we selected three scenarios from Section 7.1, 7.2.1 and 7.2.2. These represent multiple settings with gestures on diverse objects, on-the-go interaction with sensor placement on the non-gesturing finger, and finding an unambiguous combination of gestures for freehand input, as shown in Figure 22-b.

Participants. We recruited 6 right-handed participants (3M, 3F, mean age: 22.2; SD: 2.5) with an average hand sizes from Wrist to the tip of Thumb = 132mm (SD:9mm), Index = 168mm (SD:10mm), Middle = 175mm (SD:12mm), Ring = 163mm (SD:10mm), Pinky = 144mm (SD:10mm). It is noteworthy that all 6 participants were different from those who participated in creating the microgestures dataset (section 3.4).

Task and Procedure. We used the same procedure as described in Section 3.5 i.e. we counterbalanced the two conditions (Freehand and Grasping) and further counterbalanced the order of objects in each scenario. Once the object or freehand condition was selected, we presented the gesture/non-gesture states in a randomized order. We developed a custom software tool using Flask framework in Python to label the trials that the experimenter controlled during data collection. Overall, we recorded 5 trials for each gesture and Static hold for a negative class, totaling 870 trials (145 trials per participant), comprising 10 unique gestures and static hold classes on 7 different object/grasp types.

To evaluate a potential bias resulting from orientation, the data collection for this experiment was performed in a room that was different from the microgestures dataset. Additionally, the orientation of the participants was rotated by 90 degrees left from their original orientation in the microgestures dataset. The sitting/standing posture and the start and stop for labeling were similar for all scenarios as in the microgestures dataset, except for the scenario with freehand gestures (Figure 22-c). Here, we kept the posture to standing as defined in the scenario and marked the start and stop of gestures when the arm started swaying upwards from the standstill posture and returned to the initial state. Hence, the assumption is that even though coarse hand movement is involved, IMU placement is still crucial for detecting fine finger movements (gestures). The complete data collection for each participant took about 45 minutes.

Feature Extraction and Classification Model. In order to perform a systematic comparison, we extracted the same six features as used in the analyses above and in the computational design tool. These features are mean, median, minimum, standard deviation, and variance calculated from each of the 9-axis of the IMU. It is important to note that live classification requires a time window of streamed data as opposed to our tool in which we classified the entire trial. Therefore, the features were extracted on a window size of 90 and an overlap of 70 frames - only for the data collected in this study. The tool configurations remain untouched, which extracts features over the entire trial. We also used the same classifier with default parameters as used in our computational tool, i.e., Random Forest (RF) with max_depth = 30. We trained a separate grasp-independent multiclass model (not encoding grasp/object information in the class labels but only gestures) for each scenario and IMU placement. Since our participant count is lower than in the microgestures dataset, in addition to the *user-independent* models with leave-one-person-out cross validation training and testing, we also created *user-dependent* models and evaluated with leave-one-trial-out cross validation technique.

Results. Table 2 shows the comparison between the estimated F1 score from the computational tool and the performance achieved in the live classification. To understand the relative performance across configurations within a scenario, we calculate their normalized F1 score. The normalized F1 score is calculated by normalizing the F1 score of a given configuration with respect to the highest-performing configuration within this scenario. The table shows the normalized F1 scores (represented as percentages) along with absolute values for completeness. We observed that even with different hardware and participants, the results for live recognition are in congruence with the tool's prediction. Specifically, the tool correctly predicts the performance ranking of configurations, and the normalized F1 scores across configurations matches reasonably closely. Of course, this does not hold

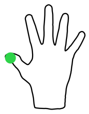
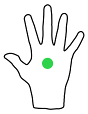
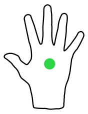
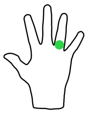
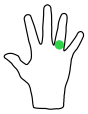
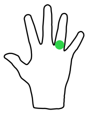
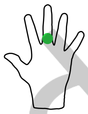
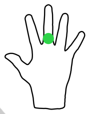
		Scenario 7.1 Supporting Diverse Objects with Minimal Instrumentation			Scenario 7.2.1 Sensor Placement on Non-Gesturing Finger			Scenario 7.2.2 Finding Unambiguous Combination of Gestures	
Objects/Grasps		bottle, knife, pestle, cup			bag, box			freehand	
Gestures		Thumb Adduction, Thumb Abduction, Thumb Tap	Thumb Adduction, Thumb Abduction, Thumb Tap	Thumb Abduction, Thumb Tap	Thumb Circle	Index Circle	Middle Circle	Middle Tap, Middle Flexion, Middle Extension	Middle Tap, Middle Abduction, Middle Extension
IMU placement									
Computational Design Tool		100% I 0.994	77% III 0.768	83% II 0.826	84% III 0.822	90% II 0.873	100% I 0.975	92% II 0.872	100% I 0.949
Live Classification	User-independent	100% I 0.860	67% III 0.577	79% II 0.682	93% III 0.903	98% II 0.953	100% I 0.973	95% II 0.615	100% I 0.644
	User-dependent	100% I 0.902	95% III 0.857	98% II 0.886	98% III 0.949	99% II 0.959	100% I 0.969	96% II 0.820	100% I 0.857

Table 2. Comparison of F1 scores from the computational design tool output and live classification. For each of the three scenarios, the object/grasp information, gesture, and location of IMU placement are described. We also included a negative class (Static hold) wrt. objects/grasps. For each scenario, the normalized F1 score of a configuration is calculated by normalizing it to the highest achieved F1 score. For completeness, we also report the absolute F1 score obtained for each configuration below the ranking. The performance ranking is denoted in roman characters.

true for the absolute values, which strongly depend on the (largely differing) settings of a configuration (live classifier, different hardware, model, train trials). However, the normalized F1 score gives an indication of what changes (improvement or deterioration) to expect when switching from one configuration to a different one. It is noteworthy that our results are consistent for all three scenarios with user-dependent as well as independent models, demonstrating the generalizability of our method.

9 DISCUSSION, LIMITATIONS, AND FUTURE WORK

While this work takes a significant first step towards rapid dense-to-sparse exploration of IMU layouts for finger microgestures, there are several aspects that need to be considered for extending this line of research:

9.1 Grasps, Objects, and Gestures in and beyond the Microgestures dataset

When constructing our dataset, we leveraged prior work on grasp types [79] to build six categories and selected representative small and large objects as well as corresponding realistic actions (cf., Figure 3). While exhaustively covering all conceivable objects for each grasp type is impossible, we anticipate generalizability for objects not present in the dataset. A few characteristics of finger movements directly depend on the grasp type and hence generalize for objects beyond the ones present in the dataset, such as the feasibility of gestures with a specific finger, and the co-activation of the non-gesturing finger. There are few other characteristics of object manipulation which might not generalize and which future work needs to address. For example, two objects may afford the same grasp type but fulfill different purposes (e.g., pen vs. soldering iron) and require different movements (fluent writing vs. a steady hold for soldering). Our computational design tool incorporates this limitation by assuming the user would briefly pause the primary activity while keeping the object in hand. The additionally collected activity data allows future work to use it for Transfer Learning [109] as both gesture and non-gesture conditions are present.

Moreover, future work may choose to augment our dataset with additional objects and activities or gestures. A promising area to expand to are rhythmic gestures incorporating a larger temporal duration, or repetitive gestures (e.g., double taps), which indicate benefits as robust wake-gestures or hot words [54]. It will also be relevant to study objects with advanced material properties, such as pronounced surface texture, friction, or deformability. For reasons of feasibility, our dataset contains gestures performed by Thumb, Index, and Middle fingers. Future work should investigate gestures performed by other fingers. Our dataset is collected using right-handed and young participants. Future work may study how this data generalizes to other populations such as the elderly (potentially limited range of motion, tremor) or children (smaller hands). We have carefully selected different object geometries that afford different orientations of hand and fingers to reduce potential dataset bias. For instance, the thumb faces upwards while holding the book, but it is sideways while holding the bottle. As a next step, future work may use data augmentation techniques to arbitrary facing (or even orientation) of the head by adding randomized orientation offsets to the raw data [97].

Data collection and labeling is a well-known problem in HCI and Machine Learning; the manually-labeled frames in our dataset can provide a quality source for auto-labeling of new data, reducing the tedious manual efforts of data labeling. Finally, it is worthwhile mentioning that our dataset offers a starting point to enable always-available input using IMUs. However, it would be fruitful if future works investigate effortless methods for data collection and labeling in the wild.

9.2 Computing, Refining, and Transferring Layout Suggestions

In this work, we contributed a tool that assists in rapidly iterating through layout suggestions for IMU placement. We understand our computational design tool's output not as a final choice, but as a 'best guess' for further refinement. For instance, if a layout with multiple IMUs is selected, an inverse kinematics (IK) model could be applied post-hoc to the set of suggested layouts to further leverage the inherent co-activation between the fingers and refine the final layout. Analogously, the current version of our tool comprises F1 score as evaluation criteria, but does not cover other metrics. In cases where robustness against false activation is a key design concern, individually showing precision/recall scores might be beneficial. Likewise, while our tool's design is relatively easy to use, visually depicting the gestures to instruct new users and strategies for an alternative representation of the confusion matrix and the F1 score can help understand the classification results.

We anticipate that the tool's layout suggestions can serve as a valuable starting point to quickly reduce the design space and for further improvement of performance in an end-to-end working system. Additional techniques such as collecting more training data to include additional variations, adding more features, performing hyperparameter tuning to tailor the classifier's behavior to the specific dataset, creating an ensemble of classifiers, and optimizing the hardware's sample rate to improve the recognition rate can be applied, if desired. Our findings show that grasp-dependent models may further improve the classification performance. This also suggests that the combination of target Freehand and/or Grasp variations affects the model's performance, where our computational design tool can be useful in rapid testing and iterations to find the balance between users' choice and classification performance. Currently, our tool suggests sensor placement based on gestures and finger choices. However, future work would include multiple alternatives at the first go or even further, it may work conversely as well, i.e., given the placement choice of sensors and the count, the tool will recommend the best gestures that can be detected. Inspired by Kohlsdorf et al. [49], future versions of the tool may also incorporate techniques to estimate the chances of false positives for each gesture by comparing the selected gestures to a large corpus of everyday activity data. This would facilitate the end-to-end framework for gesture recognition and the practical implications of real-world deployments.

While we performed user-independent evaluations in our analysis, in our initial tests, we found the performance of user-dependent models is higher with the same model architecture. With the advances in deep learning models and their interpretability methods, we believe a more sophisticated model pipeline can be constructed based on our analysis results. This would also help researchers in benchmarking different techniques to select sparse layouts.

Our current layout selections are measured by classification performance, but other factors like the required amount of training data, battery performance, hardware cost or dimensions of the sensing device could be integrated into future versions of the tool. We also see some possibility that suggestions prove useful beyond their application with IMU data. While there is some uncertainty, other approaches making use of high-dimensional data from different sensors (e.g., EMG/FSR [16, 65, 77]) can potentially expand upon the suggested layouts.

10 CONCLUSION

In this work, we presented the first computational design approach for realizing sparse IMU layouts to recognize microgestures effectively – with hands-free and while holding everyday object conditions. Our *SparseIMU* method that uses a customized version of a well-known ML metric (Feature Importance) to select sparse IMU layouts rapidly. We also contributed a computational design tool that selects sparse IMU layouts based on higher-level inputs (objects, gestures) and constraints (e.g., choice of placement) specified by the designer. We empirically validated the accuracy of the IMU layouts selected by our design tool with the combinatorial results obtained by training 393, 213 models. Selecting a sparse layout with our SparseIMU method is significantly faster than exploring the complete combinatorial space and shows a high quantitative agreement. We also contribute the first microgestures dataset, consisting of 18 gestures and 3 non-gesture states performed with freehand and 12 objects covering all the six grasp types. Using a dense network of 17 synchronized IMUs placed all over the dominant hand, we collected the data from 12 participants. Our dataset comprises fully annotated dense IMU data consisting of 13,860 trials (3 million frames). Through our dataset, we believe new insights can be derived not only for HCI research but might also be helpful for an array of other fields, including machine learning, optimization and bio-mechanics.

Our analysis revealed three major findings: i) With only 3–4 IMUs, an F1 score of about 90% can be achieved in a challenging classification task with 18 classes of Freehand and Grasping microgestures, ii) placing an IMU on a different segment on the same finger may significantly affect the classification performance, and iii) we demonstrated the feasibility of detecting gestures with an IMU placed on a non-gesturing finger. Finally, through a set of systematically designed application cases and a user study, we demonstrate how our computational design tool enables designers to employ a rapid and iterative design process for realizing microgestures for diverse scenarios across multiple objects. Our contributions in this article take advantage of fingers' dexterity and uncover the sensing potential of IMUs towards bringing computing at user's fingertips – practically everywhere and always.

ACKNOWLEDGMENTS

We thank Dustin Hoffmann and Markus Valtin from the Control Systems Group at Technische Universität Berlin, Germany, for contributing the major parts in hard- and software of the hand sensor system; without their help, this project would not have been possible. We are grateful to Frank Beruscha and Thorsten Sohnke from Bosch Research for their invaluable input on applications and our work's practicality. We also thank Marie Mühlhaus for illustrations and Luca Bläsius for his help with comparing the tool's output with live recognition evaluation. This work received funding from Bosch Research and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 714797).

REFERENCES

[1] Sheldon Andrews, Ivan Huerta, Taku Komura, Leonid Sigal, and Kenny Mitchell. 2016. Real-Time Physics-Based Motion Capture with Sparse Sensors. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP '16)*. ACM, New York, NY, USA, Article 5, 10 pages. https://doi.org/10.1145/2998559.2998564

[2] Leonardo Angelini, Francesco Carrino, Stefano Carrino, Maurizio Caon, Omar Abou Khaled, Jürgen Baumgartner, Andreas Sonderegger, Denis Lalanne, and Elena Mugellini. 2014. Gesturing on the Steering Wheel: A User-Elicited Taxonomy. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '14)*. ACM, New York, NY, USA, 1–8. https://doi.org/10.1145/2667317.2667414

[3] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: Subtle and Eyes-Free Mobile Input with a Magnetically-Tracked Finger Ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2043–2046. https://doi.org/10.1145/1978942.1979238

[4] Daniel Ashbrook and Thad Starner. 2010. MAGIC: A Motion Gesture Design Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2159–2168. https://doi.org/10.1145/1753326.1753653

[5] Daniel Lee Ashbrook. 2010. *Enabling mobile microinteractions*. Ph.D. Dissertation. Georgia Institute of Technology.

[6] Sandra Bardot, Surya Rawat, Duy Thai Nguyen, Sawyer Rempel, Huizhe Zheng, Bradley Rey, Jun Li, Kevin Fan, Da-Yuan Huang, Wei Li, and Pourang Irani. 2021. ARO: Exploring the Design of Smart-Ring Interactions for Encumbered Hands. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction (MobileHCI '21)*. ACM, New York, NY, USA, Article 12, 11 pages. https://doi.org/10.1145/3447526.3472037

[7] Roger Boldu, Alexandru Dancu, Denys J.C. Matthies, Pablo Gallego Cascón, Shanaka Ransir, and Suranga Nanayakkara. 2018. Thumb-In-Motion: Evaluating Thumb-to-Ring Microgestures for Athletic Activity. In *Proceedings of the Symposium on Spatial User Interaction (SUI '18)*. ACM, New York, NY, USA, 150–157. https://doi.org/10.1145/3267782.3267796

[8] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. 2020. ContactPose: A Dataset of Grasps with Object Contact and Hand Pose. In *The European Conference on Computer Vision (ECCV '20)*. https://doi.org/10.48550/arxiv.2007.09545

[9] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[10] Bingni W Brunton, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. 2013. Optimal sensor placement and enhanced sparsity for classification. *arXiv e-prints* (2013). https://doi.org/10.48550/arxiv.1310.4217

[11] Ian M. Bullock, Joshua Z. Zheng, Sara De La Rosa, Charlotte Guertler, and Aaron M. Dollar. 2013. Grasp Frequency and Usage in Daily Household and Machine Shop Tasks. *IEEE Transactions on Haptics* 6, 3 (2013), 296–308. https://doi.org/10.1109/TOH.2013.6

[12] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User Elicitation on Single-Hand Microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3403–3414. https://doi.org/10.1145/2858036.2858589

[13] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 549–556. https://doi.org/10.1145/2807442.2807450

[14] Weiya Chen, Chenchen Yu, Chenyu Tu, Zehua Lyu, Jing Tang, Shiqi Ou, Yan Fu, and Zhidong Xue. 2020. A Survey on Hand Pose Estimation with Wearable Sensors and Computer-Vision-Based Methods. *Sensors* 20, 4 (2020). https://doi.org/10.3390/s20041074

[15] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. 2018. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing* 307 (2018), 72–77. https://doi.org/10.1016/j.neucom.2018.03.067

[16] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-Power Gesture Input with Wrist-Worn Pressure Sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 161–166. https://doi.org/10.1145/2642918.2647396

[17] Artem Dementyev, Tomás Vega Gálvez, and Alex Olwal. 2019. SensorSnaps: Integrating Wireless Sensor Nodes into Fabric Snap Fasteners for Textile Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New York, NY, USA, 17–28. https://doi.org/10.1145/3332165.3347913

[18] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 278–288. https://doi.org/10.1145/3025453.3025739

[19] Sijiang Du and M. Vuskovic. 2004. Temporal vs. spectral approach to feature extraction from prehensile EMG signals. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. (IRI '04)*. 344–350. https://doi.org/10.1109/IRI.2004.1431485

[20] Karsten Eckhoff, Manon Kok, Sergio Lucia, and Thomas Seel. 2020. Sparse Magnetometer-free Inertial Motion Tracking – A Condition for Observability in Double Hinge Joint Systems. *21st IFAC World Congress* 53, 2 (2020), 16023–16030. https://doi.org/10.1016/j.ifacol.2020.12.403

[21] Daniele Esposito, Emilio Andreozzi, Gaetano D. Gargiulo, Antonio Fratini, Giovanni D'Addio, Ganesh R. Naik, and Paolo Bifulco. 2020. A Piezoresistive Array Armband With Reduced Number of Sensors for Hand Gesture Recognition. *Frontiers in Neurorobotics* 13 (2020). https://doi.org/10.3389/fnbot.2019.00114

[22] Junjun Fan, Xiangmin Fan, Feng Tian, Yang Li, Zitao Liu, Wei Sun, and Hongan Wang. 2018. What is That in Your Hand? Recognizing Grasped Objects via Forearm Electromyography Sensing. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '18, Vol. 2)*. ACM, New York, NY, USA, Article 161, 24 pages. https://doi.org/10.1145/3287039

[23] Manus | Prime II for Mocap. 2021. Retrieved 2021-03-13 from https://www.manus-vr.com/mocap-gloves

[24] David F. Fouhey, Wei-cheng Kuo, Alexei A. Efros, and Jitendra Malik. 2018. From Lifestyle Vlogs to Everyday Interactions. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '18)*. 4991–5000. https://doi.org/10.1109/CVPR.2018.00524

[25] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. 2018. First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '18)*. https://doi.org/10.48550/ARXIV.1704.02463

[26] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. 2019. Interactive Hand Pose Estimation Using a Stretch-Sensing Soft Glove. *ACM Transaction on Graphics* 38, 4, Article 41, 15 pages. https://doi.org/10.1145/3306346.3322957

[27] Jun Gong, Yang Zhang, Xia Zhou, and Xing-Dong Yang. 2017. Pyro: Thumb-Tip Gesture Recognition Using Pyroelectric Infrared Sensing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 553–563. https://doi.org/10.1145/3126594.3126615

[28] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New York, NY, USA, 1059–1070. https://doi.org/10.1145/3332165.3347947

[29] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. RotoSwype: Word-Gesture Typing Using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300244

[30] Aakar Gupta, Jiushan Yang, and Ravin Balakrishnan. 2018. Asterisk and Obelisk: Motion Codes for Passive Tagging. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 725–736. https://doi.org/10.1145/3242587.3242637

[31] Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. 2010. Imaginary Interfaces: Spatial Interaction with Empty Hands and without Visual Feedback. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 3–12. https://doi.org/10.1145/1866029.1866033

[32] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O. Riedl. 2019. Friend, Collaborator, Student, Manager: How Design of an AI-Driven Game Level Editor Affects Creators. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300854

[33] Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D. Twigg, and Kenrick Kin. 2018. Online Optical Marker-Based Hand Tracking with Deep Labels. *ACM Transactions on Graphics* 37, 4, Article 166, 10 pages. https://doi.org/10.1145/3197517.3201399

[34] Seongkook Heo, Michelle Annett, Benjamin J. Lafreniere, Tovi Grossman, and George W. Fitzmaurice. 2017. No Need to Stop What You're Doing: Exploring No-Handed Smartwatch Interaction. In *Proceedings of the 43rd Graphics Interface Conference 2017 (GI '17)*. Canadian Human-Computer Communications Society / ACM, 107–114. https://doi.org/10.20380/GI2017.14

[35] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + Touch = New Tools. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 27–36. https://doi.org/10.1145/1866029.1866036

[36] G. Holmes, A. Donkin, and I.H. Witten. 1994. WEKA: a machine learning workbench. In *Proceedings of Australian New Zealnd Intelligent Information Systems Conference (ANZIIS '94)*. 357–361. https://doi.org/10.1109/ANZIIS.1994.396988

[37] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1526–1537. https://doi.org/10.1145/2858036.2858483

[38] Kunpeng Huang, Ruojia Sun, Ximeng Zhang, Md. Tahmidul Islam Molla, Margaret Dunne, Francois Guimbretiere, and Cindy Hsin-Liu Kao. 2021. WovenProbe: Probing Possibilities for Weaving Fully-Integrated On-Skin Systems Deployable in the Field. In *Designing Interactive Systems Conference 2021 (DIS '21)*. ACM, New York, NY, USA, 1143–1158. https://doi.org/10.1145/3461778.3462105

[39] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time. *ACM Transactions on Graphics* 37, 6, Article 185 (2018), 15 pages. https://doi.org/10.1145/3272127.3275108

[40] James N Ingram, Konrad P Körding, Ian S Howard, and Daniel M Wolpert. 2008. The statistics of natural hand movements. *Experimental brain research* 188, 2 (2008), 223–236. https://doi.org/10.1007/s00221-008-1355-3

[41] Yasha Iravantchi, Mayank Goel, and Chris Harrison. 2019. BeamBand: Hand Gesture Sensing with Ultrasonic Beamforming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/3290605.3300245

[42] L Jäncke, H Steinmetz, S Benilow, and U Ziemann. 2004. Slowing fastest finger movements of the dominant hand with low-frequency rTMS of the hand area of the primary motor cortex. *Experimental Brain Research* 155, 2 (2004), 196–203. https://doi.org/10.1007/s00221-003-1719-7

[43] Lynette A Jones and Susan J Lederman. 2006. *Human hand function.* Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195173154.001.0001

[44] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. *Digits: Freehand 3D Interactions Anywhere Using a Wrist-Worn Gloveless Sensor.* ACM, New York, NY, USA, 167–176. https://doi.org/10.1145/2380116.2380139

[45] Hyejoo Kim, Hyeon-Joo Kim, Jinyoon Park, Jeh-Kwang Ryu, and Seung-Chan Kim. 2021. Recognition of Fine-Grained Walking Patterns Using a Smartwatch with Deep Attentive Neural Networks. *Sensors* 21, 19 (2021). https://doi.org/10.3390/s21196393

[46] Ju-Whan Kim, Han-Jong Kim, and Tek-Jin Nam. 2016. M.Gesture: An Acceleration-Based Gesture Authoring System on Multiple Handheld and Wearable Devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2307–2318. https://doi.org/10.1145/2858036.2858358

[47] Ju-Whan Kim and Tek-Jin Nam. 2013. EventHurdle: Supporting Designers' Exploratory Interaction Prototyping with Gesture-Based Sensors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 267–276. https://doi.org/10.1145/2470654.2470691

[48] Francis Kinder. 2021. Wearable VR Glove Trailer. Retrieved March 21, 2021 from https://vimeo.com/527250099

[49] Daniel Kohlsdorf, Thad Starner, and Daniel Ashbrook. 2011. MAGIC 2.0: A web tool for false positive prediction and prevention for gesture recognition systems. In *2011 IEEE International Conference on Automatic Face Gesture Recognition (FG '11)*. 1–6. https://doi.org/10.1109/FG.2011.5771412

[50] Thomas Kosch, Kevin Wennrich, Daniel Topp, Marcel Muntzinger, and Albrecht Schmidt. 2019. The Digital Cooking Coach: Using Visual and Auditory in-Situ Instructions to Assist Cognitively Impaired during Cooking. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '19)*. ACM, New York, NY, USA, 156–163. https://doi.org/10.1145/3316782.3321524

[51] Yuki Kubo, Yuto Koguchi, Buntarou Shizuki, Shin Takahashi, and Otmar Hilliges. 2019. AudioTouch: Minimally Invasive Sensing of Micro-Gestures via Active Bio-Acoustic Sensing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. ACM, New York, NY, USA, Article 36, 13 pages. https://doi.org/10.1145/3338286.3340147

[52] Gierad Laput and Chris Harrison. 2019. Sensing Fine-Grained Hand Activity with Smartwatches. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300568

[53] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 321–333. https://doi.org/10.1145/2984511.2984582

[54] Juyoung Lee, Shaurye Aggarwal, Jason Wu, Thad Starner, and Woontack Woo. 2019. SelfSync: Exploring Self-Synchronous Body-Based Hotword Gestures for Initiating Interaction. In *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC '19)*. ACM, New York, NY, USA, 123–128. https://doi.org/10.1145/3341163.3347745

[55] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. 2021. DualRing: Enabling Subtle and Expressive Hand Interaction with Dual IMU Rings. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '21, Vol. 5)*. ACM, New York, NY, USA, Article 115, 27 pages. https://doi.org/10.1145/3478114

[56] Jhe-Wei Lin, Chiuan Wang, Yi Yao Huang, Kuan-Ting Chou, Hsuan-Yu Chen, Wei-Luan Tseng, and Mike Y. Chen. 2015. BackHand: Sensing Hand Gestures via Back of the Hand. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 557–564. https://doi.org/10.1145/2807442.2807462

[57] Stephen Shiao-ru Lin, Nisal Menuka Gamage, Kithmini Herath, and Anusha Withana. 2022. MyoSpring: 3D Printing Mechanomyographic Sensors for Subtle Finger Gesture Recognition. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '22)*. ACM, New York, NY, USA, Article 15, 13 pages. https://doi.org/10.1145/3490149.3501321

[58] Allan Christian Long, James A Landay, and Lawrence A Rowe. 2001. *Quill: a gesture design tool for pen-based user interfaces.* University of California, Berkeley.

[59] Gilles Louppe. 2014. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502* (2014). https://doi.org/10.48550/ARXIV.1407.7502

[60] Hao Lü and Yang Li. 2012. Gesture Coder: A Tool for Programming Multi-Touch Gestures by Demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2875–2884. https://doi.org/10.1145/2207676.2208693

[61] Teachable Machine. 2021. Retrieved 2021-04-07 from https://teachablemachine.withgoogle.com/

[62] Tomás Mantecón, Ana Mantecón, Carlos R. del Blanco, Fernando Jaureguizar, and Narciso García. 2015. Enhanced gesture-based human-computer interaction through a Compressive Sensing reduction scheme of very large and efficient depth feature descriptors. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '15)*. 1–6. https://doi.org/10.1109/AVSS.2015.7301804

[63] Jess McIntosh, Asier Marzo, and Mike Fraser. 2017. SensIR: Detecting Hand Gestures with a Wearable Bracelet Using Infrared Transmission and Reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 593–597. https://doi.org/10.1145/3126594.3126604

[64] Jess McIntosh, Asier Marzo, Mike Fraser, and Carol Phillips. 2017. EchoFlex: Hand Gesture Recognition Using Ultrasound Imaging. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1923–1934. https://doi.org/10.1145/3025453.3025807

[65] Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. 2016. EMPress: practical hand gesture classification with wrist-mounted EMG and pressure sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2332–2342.

[66] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. 2019. Real-Time Pose and Shape Reconstruction of Two Interacting Hands with a Single Depth Camera. *ACM Transactions on Graphics* 38, 4, Article 49 (2019), 13 pages. https://doi.org/10.1145/3306346.3322958

[67] Ju Young Oh, Ji-Hyung Park, and Jung-Min Park. 2020. FingerTouch: Touch Interaction Using a Fingernail-Mounted Sensor on a Head-Mounted Display for Augmented Reality. *IEEE Access* 8 (2020), 101192–101208. https://doi.org/10.1109/ACCESS.2020.2997972

[68] Ahmed A A Osman, Timo Bolkart, and Michael J. Black. 2020. STAR: A Sparse Trained Articulated Human Body Regressor. In *European Conference on Computer Vision (ECCV '20)*. 598–613. https://star.is.tue.mpg.de

[69] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. 2019. AuraRing: Precise Electromagnetic Finger Tracking. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '19, Vol. 3)*. Association for Computing Machinery, New York, NY, USA, Article 150, 28 pages. https://doi.org/10.1145/3369831

[70] Aman Parnami, Apurva Gupta, Gabriel Reyes, Ramik Sadana, Yang Li, and Gregory D. Abowd. 2016. Mogeste: A Mobile Tool for In-Situ Motion Gesture Design. In *Proceedings of the 8th Indian Conference on Human Computer Interaction (IHCI '16)*. ACM, New York, NY, USA, 35–43. https://doi.org/10.1145/3014362.3014365

[71] Kayur Patel, Naomi Bancroft, Steven M. Drucker, James Fogarty, Amy J. Ko, and James Landay. 2010. Gestalt: Integrated Support for Implementation and Analysis in Machine Learning. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 37–46. https://doi.org/10.1145/1866029.1866038

[72] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. http://arxiv.org/abs/1201.0490

[73] Patryk Pomykalski, Mikołaj P. Woźniak, Paweł W. Woźniak, Krzysztof Grudzień, Shengdong Zhao, and Andrzej Romanowski. 2020. Considering Wake Gestures for Smart Assistant Use. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. ACM, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3383089

[74] Alan Poston. 2000. Human engineering design data digest. *Washington, DC: Department of Defense Human Factors Engineering Technical Advisory Group* (2000), 61–75.

[75] S. Poularakis, G. Tsagkatakis, P. Tsakalides, and I. Katsavounidis. 2013. Sparse representations for hand gesture recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '13)*. 3746–3750. https://doi.org/10.1109/ICASSP.2013.6638358

[76] Christina Salchow-Hömmen, Leonie Callies, Daniel Laidig, Markus Valtin, Thomas Schauer, and Thomas Seel. 2019. A Tangible Solution for Hand Motion Tracking in Clinical Applications. *Sensors* 19, 1 (2019). https://doi.org/10.3390/s19010208

[77] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling Always-Available Input with Muscle-Computer Interfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 167–176. https://doi.org/10.1145/1622176.1622208

[78] Marc H Schieber. 1991. Individuated finger movements of rhesus monkeys: a means of quantifying the independence of the digits. *Journal of neurophysiology* 65, 6 (1991), 1381–1391. https://doi.org/10.1152/jn.1991.65.6.1381

[79] G. Schlesinger. 1919. *Der mechanische Aufbau der künstlichen Glieder*. Springer, Berlin, Heidelberg, 321–661. https://doi.org/10.1007/978-3-662-33009-8_13

[80] Lea Schönherr, Maximilian Golla, Thorsten Eisenhofer, Jan Wiele, Dorothea Kolossa, and Thorsten Holz. 2020. Unacceptable, where is my privacy? Exploring Accidental Triggers of Smart Speakers. (2020). arXiv:2008.00508 https://arxiv.org/abs/2008.00508

[81] Matthias Schröder, Jonathan Maycock, and Mario Botsch. 2015. Reduced Marker Layouts for Optical Motion Capture of Hands. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games (MIG '15)*. ACM, New York, NY, USA, 7–16. https://doi.org/10.1145/2822013.2822026

[82] Adwait Sharma, Michael A. Hedderich, Divyanshu Bhardwaj, Bruno Fruchard, Jess McIntosh, Aditya Shekhar Nittala, Dietrich Klakow, Daniel Ashbrook, and Jürgen Steimle. 2021. SoloFinger: Robust Microgestures While Grasping Everyday Objects. In *Proceedings*

of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21). ACM, New York, NY, USA, Article 744, 15 pages. https://doi.org/10.1145/3411764.3445197

[83] Adwait Sharma, Joan Sol Roo, and Jürgen Steimle. 2019. Grasping Microgestures: Eliciting Single-Hand Microgestures for Handheld Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19).* ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300632

[84] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. 2020. Ready, Steady, Touch! Sensing Physical Contact with a Finger-Mounted IMU. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '20, Vol. 4).* ACM, New York, NY, USA, Article 59, 25 pages. https://doi.org/10.1145/3397309

[85] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces (ISS '18).* ACM, New York, NY, USA, 177–187. https://doi.org/10.1145/3279778.3279799

[86] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. 2015. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15).* ACM, New York, NY, USA, 3643–3652. https://doi.org/10.1145/2702123.2702136

[87] Yuta Sugiura, Fumihiko Nakamura, Wataru Kawai, Takashi Kikuchi, and Maki Sugimoto. 2017. Behind the palm: Hand gesture recognition through measuring skin deformation on back of hand by using optical sensors. In *56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE '17).* 1082–1087. https://doi.org/10.23919/SICE.2017.8105457

[88] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. 2019. Learning the signatures of the human grasp using a scalable tactile glove. *Nature* 569, 7758 (2019), 698–702. https://doi.org/10.1038/s41586-019-1234-z

[89] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. 2020. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In *European Conference on Computer Vision (ECCV '20).* https://grab.is.tue.mpg.de

[90] Brandon T. Taylor and V. Michael Bove. 2009. Graspables: Grasp-Recognition as a User Interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09).* ACM, New York, NY, USA, 917–926. https://doi.org/10.1145/1518701.1518842

[91] Hsin-Ruey Tsai, Min-Chieh Hsiu, Jui-Chun Hsiao, Lee-Ting Huang, Mike Chen, and Yi-Ping Hung. 2016. TouchRing: Subtle and Always-Available Input Using a Multi-Touch Ring. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16).* ACM, New York, NY, USA, 891–898. https://doi.org/10.1145/2957265.2961860

[92] Hsin-Ruey Tsai, Cheng-Yuan Wu, Lee-Ting Huang, and Yi-Ping Hung. 2016. ThumbRing: Private Interactions Using One-Handed Thumb Motion Input on Finger Segments. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16).* ACM, New York, NY, USA, 791–798. https://doi.org/10.1145/2957265.2961859

[93] Markus Valtin, Christina Salchow-Hömmen, Thomas Seel, Daniel Laidig, and Thomas Schauer. 2017. Modular finger and hand motion capturing system based on inertial and magnetic sensors. *Current Directions in Biomedical Engineering* 3 (03 2017), 19–23. https://doi.org/10.1515/cdbme-2017-0005

[94] Tijana Vuletic, Alex Duffy, Laura Hay, Chris McTeague, Gerard Campbell, and Madeleine Grealy. 2019. Systematic literature review of hand gestures used in human computer interaction interfaces. *International Journal of Human-Computer Studies* 129 (2019), 74–94. https://doi.org/10.1016/j.ijhcs.2019.03.011

[95] Lefan Wang, Turgut Meydan, and Paul Ieuan Williams. 2017. A Two-Axis Goniometric Sensor for Tracking Finger Motion. *Sensors* 17, 4 (2017). https://doi.org/10.3390/s17040770

[96] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. 2016. Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16).* ACM, New York, NY, USA, 851–860. https://doi.org/10.1145/2984511.2984565

[97] Daniel Weber, Clemens Gühmann, and Thomas Seel. 2021. RIANN—A Robust Neural Network Outperforms Attitude Estimation Filters. *AI* 2, 3 (2021), 444–463. https://doi.org/10.3390/ai2030028

[98] Martin Weigel and Jürgen Steimle. 2017. DeformWear: Deformation Input on Tiny Wearable Devices. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT '17, Vol. 1).* ACM, New York, NY, USA, Article 28, 23 pages. https://doi.org/10.1145/3090093

[99] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-Defined Gestures for Surface Computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09).* ACM, New York, NY, USA, 1083–1092. https://doi.org/10.1145/1518701.1518866

[100] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07).* ACM, New York, NY, USA, 159–168. https://doi.org/10.1145/1294211.1294238

[101] Katrin Wolf, Sven Mayer, and Stephan Meyer. 2016. Microgesture Detection for Remote Interaction with Mobile Devices. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16).* ACM, New York, NY, USA, 783–790. https://doi.org/10.1145/2957265.2961865

[102] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. 2011. A Taxonomy of Microinteractions: Defining Microgestures Based on Ergonomic and Scenario-Dependent Requirements. In *Human-Computer Interaction – INTERACT 2011*. Springer, Berlin, Heidelberg, 559–575. https://doi.org/10.1007/978-3-642-23774-4_45

[103] Katrin Wolf, Robert Schleicher, Sven Kratz, and Michael Rohs. 2013. Tickle: A Surface-Independent Interaction Technique for Grasp Interfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. ACM, New York, NY, USA, 185–192. https://doi.org/10.1145/2460625.2460654

[104] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, et al. 2022. Enabling hand gesture customization on wrist-worn devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. ACM. https://doi.org/10.48550/arXiv.2203.15239

[105] Xuhai Xu, Haitian Shi, Xin Yi, WenJia Liu, Yukang Yan, Yuanchun Shi, Alex Mariakakis, Jennifer Mankoff, and Anind K. Dey. 2020. EarBuddy: Enabling On-Face Interaction via Wireless Earbuds. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376836

[106] Yukang Yan, Chun Yu, Yingtian Shi, and Minxing Xie. 2019. PrivateTalk: Activating Voice Input with Hand-On-Mouth Gesture Detected by Bluetooth Earphones. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New York, NY, USA, 1013–1020. https://doi.org/10.1145/3332165.3347950

[107] Hui-Shyong Yeo, Juyoung Lee, Hyung-il Kim, Aakar Gupta, Andrea Bianchi, Daniel Vogel, Hideki Koike, Woontack Woo, and Aaron Quigley. 2019. WRIST: Watch-Ring Interaction and Sensing Technique for Wrist Gestures and Macro-Micro Pointing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. ACM, New York, NY, USA, Article 19, 15 pages. https://doi.org/10.1145/3338286.3340130

[108] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. 2019. Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New York, NY, USA, 963–971. https://doi.org/10.1145/3332165.3347867

[109] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76. https://doi.org/10.1109/JPROC.2020.3004555