SignQuery: A Natural User Interface and Search Engine for Sign Languages with Wearable Sensors

Hao Zhou[ℙ], Taiting Lu[ℙ], Kristina McKinnie[𝔾], Joseph Palagano[𝔾]
Kenneth DeHaan[𝔾], Mahanth Gowda[ℙ]
[ℙ]The Pennsylvania State University, [𝔾]Gallaudet University

ABSTRACT

Search Engines such as Google, Baidu, and Bing have revolutionized the way we interact with the cyber world with a number of applications in recommendations, learning, advertisements, healthcare, entertainment, etc. In this paper, we design search engines for sign languages such as American Sign Language (ASL). Sign languages use hand and body motion for communication with rich grammar, complexity, and vocabulary that is comparable to spoken languages. This is the primary language for the Deaf community with a global population of ≈ 500 million. However, search engines that support sign language queries in native form do not exist currently. While translating a sign language to a spoken language and using existing search engines might be one possibility, this can miss critical information because existing translation systems are either limited in vocabulary or constrained to a specific domain. In contrast, this paper presents a holistic approach where ASL queries in native form as well as ASL videos and textual information available online are converted into a common representation space. Such a joint representation space provides a common framework for precisely representing different sources of information and accurately matching a query with relevant information that is available online. Our system uses low-intrusive wearable sensors for capturing the sign query. To minimize the training overhead, we obtain synthetic training data from a large corpus of online ASL videos across diverse topics. Evaluated over a set of Deaf users with native ASL fluency, the accuracy is comparable with state-of-the-art recommendation systems for Amazon, Netflix, Yelp, etc., suggesting the usability of the system in the real world. For example, the recall@10 of our system is 64.3%, i.e., among the top ten search results, six of them are relevant to the search query. Moreover, the system is robust to variations in signing patterns, dialects, sensor positions, etc.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**; **Accessibility systems and tools**.

KEYWORDS

Accessibility, Sign Language, Recommendation System, Wearable Computing

ACM Reference Format:
Hao Zhou[ℙ], Taiting Lu[ℙ], Kristina McKinnie[𝔾], Joseph Palagano[𝔾], Kenneth DeHaan[𝔾], Mahanth Gowda[ℙ]. 2023. SignQuery: A Natural User Interface and Search Engine for Sign Languages with Wearable Sensors. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3570361.3613286

1 INTRODUCTION

The revolution in natural language processing and deep learning has led to advances in information retrieval, knowledge extraction, representation learning, and question answering. By exploiting these techniques, search engines such as Google, Baidu, and Bing enable several applications in learning, recommendations, resolving technical issues, home maintenance, health care, and advertisements [16, 82, 94, 99] and thus have become an integral part of human life.

The capabilities of search engines extend to a wide variety of spoken natural languages with seamless information sharing across different languages powered by advances in language translation techniques [25, 91]. In this paper, we extend the benefits of search engines to sign languages which are a form of natural languages

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10. . . $15.00
https://doi.org/10.1145/3570361.3613286

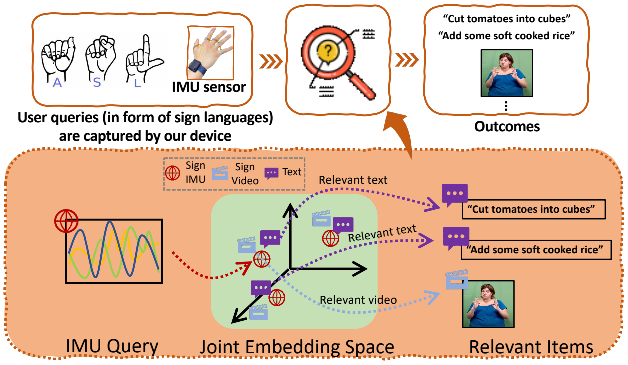

A
S
L
IMU sensor
User queries (in form of sign languages) are captured by our device
“Cut tomatoes into cubes”
“Add some soft cooked rice”
⋮
Outcomes
Sign IMU
Sign Video
Text
Relevant text
Relevant text
Relevant video
“Cut tomatoes into cubes”
“Add some soft cooked rice”
IMU Query
Joint Embedding Space
Relevant Items

Figure 1: The vision of *SignQuery* is to build a query system that serves Deaf users. In *SignQuery*, users could sign (captured by Inertial Measurement Unit, IMU sensors) instead of typing text to search for relevant documents (e.g. ASL videos) that are already in a native sign language, thus improving accessibility. To realize this, *SignQuery* takes advantage of the idea of cross-model embedding to embed various modalities into a joint embedding space. For a given query, *SignQuery* could return relevant documents by searching for the embeddings with high similarities.

that use the visual-manual modality for communication instead of audio. Primarily used by the Deaf community, sign languages mainly use hands and non-manual markers (e.g., face and body) to produce a natural language with its own grammar and lexicon with a large vocabulary and complexity similar to spoken languages [15, 33]. The population of the deaf and hard of hearing (DHH) individuals today, is upward of 10 million in the US and ≈500 million globally [1, 72]. Therefore, we believe enabling a natural system and interface for querying with sign languages can tremendously improve the accessibility to mainstream products that have long been inaccessible to signed languages.

Towards bridging this gap, this paper proposes a system called *SignQuery*, that supports queries in American Sign Language (ASL). Figure 1 depicts the high-level overview of the system. It consists of a system of wearable sensors such as smart rings, using which *SignQuery* tracks the hand motion of users to capture the query. The sensory information is converted into a representation that can be searched in a database of existing ASL videos or English text for recommending and providing relevant information in response to their query. To achieve such a search engine for better serving the Deaf community, one naive solution is to employ a two-step process in which sign language translation systems [20, 38] can be utilized to translate signs and videos to text. Search engines that are designed for hearing people can also be utilized. Such a naive solution heavily relies on accurate sign-to-text and video-to-text systems. Unfortunately, despite the recent efforts on sign language recognition and translation [20, 40, 67], the systems that perfectly interpret sign languages are still in concept development. Most systems have focused on constrained domains (e.g., weather forecast) [20, 21], leading to a generalization problem when comes to open-vocabulary searching in a collection of sign videos that covers various activities such as sports, entertainment, personal care, education, home and garden. In contrast, by encoding signs, text, and videos into the same representation space as shown in Figure 1, *SignQuery* performs direct matching without the low-accurate two-step process [31]. We experimentally justify such a choice in §5.3.5.

Prior work in this area mainly includes queries that are submitted in textual format, which is then matched with appropriate sign language videos online [31] to extract the videos that best match the queries. However, such a system overlooks an important fact which is *language deprivation*. Millions of deaf children are born to parents and educators that do not know sign language or are qualified to teach Deaf Education, resulting in severe language deprivation and literacy challenges [3, 77]. While text-based search is a solution for Deaf users who are also fluent in text, we believe providing a system that supports queries in a native language format is essential to ensure equitable access principles [19] for all Deaf users to search online. Additionally, the system can be used for instructional and educational purposes. Also, while cameras can be another alternative to wearable sensors, in contrast to camera-based approaches [52, 56] that can be privacy-sensitive and need good lighting and resolution, wearable sensors like *SignQuery* offer solutions that are privacy-agnostic and work anywhere including heavy occlusions or outdoors where the user is constantly moving.

Building the *SignQuery* system has a lot of challenges: (i) Sign languages involve complex and intricate motions of fingers from both hands. (ii) The modality of data captured from wearable sensors (i.e., Inertial Measurement Unit, IMU) is different from the target database online for searching, which is typically in the form of videos and text. IMU data from the query needs to be matched with online video and text to extract relevant documents corresponding to the query. (iii) The success of deep learning models in domains like speech and computer vision can be widely attributed to the availability of large-scale and diverse training datasets. However, there is no such training data currently available for wearable sensors with ASL. (iv) Similar to spoken languages, sign languages such as ASL also have accents and dialects (e.g., signing patterns). The *SignQuery* system must provide consistent accuracy across such natural variation.

SignQuery exploits a number of opportunities to address the above challenges. (i) Low intrusive wearable devices in the form of smart rings are designed to capture the hand motions of the querying sign language user. (ii) Deep Learning algorithms are designed to convert wearable sensor data, video, and text into a common representation space for measuring similarity and ranking the items in the database when the modality of querying (e.g., wearable sensor data) is different from the modality of targets (e.g., videos, text) in the database. (iii) Leveraging on the success of synthetic training data [55, 64], *SignQuery* derives training data from *How2Sign* [32], a comprehensive database of sign language videos in ASL with a large vocabulary size, and on diverse topics such as arts, sports, personal care, education, home and garden. Virtual IMU data is extracted from *How2Sign* without any overhead of training data generation from actual wearable devices. (iv) The *How2Sign* database is generated from a diverse set of users across different backgrounds, accents, and dialects. Therefore we believe building *SignQuery* on top of *How2Sign* provides inherent robustness to variation across users and accents.

An extensive real-user study with a group of Deaf native ASL signers is conducted to validate the performance of *SignQuery* with a vocabulary size of 15896. In a nutshell, *SignQuery* achieves a recall@10 of 64.3%, suggesting six out of the top ten search results is relevant to given queries, and qualitative results (see Fig. 10) depict *SignQuery* retrieves relevant documents given queries. These results are comparable with state-of-the-art recommendation systems for Amazon, Netflix, Yelp, etc [23, 58], indicating the usability of the system in the real world. Furthermore, the accuracy is also consistent across variations in dialects/accents, sensor wearing positions, and signing speeds, indicating the robustness.

In summary, we enumerate our contributions below: ❶ To our best knowledge, *SignQuery* is the first system that supports searching in a database of existing ASL videos or English text for recommending and providing relevant information in response to queries in native sign languages (i.e., ASL) by Deaf users. ❷ We encode the query space (i.e., IMU data) and the search space (i.e., ASL videos and text) into a common representation space, allowing the direct matching of the query with online documents to retrieve the most relevant documents in response to the query. ❸ We extended the *How2sign* dataset with virtual IMU data pairs for multimodal learning. ❹ We conducted an extensive real-user study with Deaf users to validate the feasibility of *SignQuery*.

2 RELATED WORK

Visual Language Retrieval. Lee et al. [57] align words and image objects with similar semantics. Gabeur et al. [37] embed cues from audio, text, and videos into one retrieval space. Liu et al. [60] leverage video analysis tools to create a unified retrieval space. SPOT-ALIGN [31] links text with sign videos in a common space, emphasizing sign-video and text alignment. Differing from these text-based queries, *SignQuery* allows for sign language (ASL) searches. This approach offers better accessibility to Deaf users particularly when the search space includes ASL videos and a more natural interface for Deaf users, especially for those using ASL as their main communication mode due to language deprivation [3, 77].

Finger Motion Analytics using Wearables. Numerous wearables, such as IMU, WiFi, and acoustic signals, have been explored for finger motion recognition [63, 65, 74, 81, 86, 100]. Systems like uWave [59] utilize IMUs for user identity and device interaction. FingerIO [70] and FingerPing [95] employ acoustic signals for gesture classification. ZeroNet [64] uses IMU data from videos for hand gesture classification, while Capband [85] and ThumbTrak [83] adopt capacitive and proximity sensors respectively. ElectroRing [50] merges electrode and IMU signals for detailed finger gesture detection. However, unlike these systems which mostly target predefined gestures, to the best of our knowledge, *SignQuery* stands out as the first work centered on sign language native search engines, addressing challenges in representation learning, recommendation, and validation with actual sign language users.

Finger Motion Analytics using Cameras. Depth cameras like Kinect [52] and Leap motion [56] provide sophisticated finger motion tracking, and deep learning algorithms enable 3D tracking with just RGB cameras [18, 47, 68]. However, cameras have limitations, including privacy concerns, lighting and resolution needs, and the requirement for the user to be within the camera's sight. For more mobility, wrist-mounted cameras have been researched [44, 51, 92], but they present challenges in capturing all fingers and sensitivity to background temperatures. Unlike these methods, *SignQuery* offers a versatile solution free from environmental constraints and privacy issues.

Sign Language Recognition and Translation using Cameras. Sign language recognition tasks have been explored mainly in the vision world [40, 67, 104] where researchers want to align signs with extracted visual features. Camgoz et al. [21] propose a joint transformer of sign language recognition and translation where connectionist temporal classification loss is applied to achieve an end-to-end training manner. Although the above work demonstrated encouraging performance for translating sign languages into spoken languages such as English, and Germany in constrained domains (e.g., weather forecast), these systems still lack the ability of cross-domain generalization [54, 87], leading to inaccurate searching intents if such systems are employed in *SignQuery*. Thus, in contrast to these work, *SignQuery* captures the signs directly through IMU sensors and embeds them into a joint space with text and sign videos.

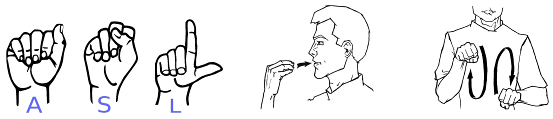
Sign Language Recognition and Translation using Wearables. Wearables have been explored for sign language recognition and translation. This includes electromyography (e.g., MyoSign [98], DeepSLR [90]), IMU (e.g., FinGTrAC [61], SignSpeaker [43], WearSign [97]), wearable cameras (e.g., DeepASL [36]), and acoustics (e.g., SonicASL [48]). These papers explore recognition or translation in constrained domains with a limited vocabulary (e.g., 20 - 150 words). Moreover, none of the above work was validated with Deaf users with native fluency. In contrast, *SignQuery* provides a natural user interface for querying directly in sign languages (e.g., ASL) with a search space of 15896 words that covers multiple domains such as sports, arts, personal care, education, home and garden. with high accuracy as validated by a user study of Deaf users with native ASL fluency.

3 PRELIMINARY

3.1 American Sign Language

Sign languages use gestures instead of sound for communication. They are considered a class of natural languages with their own grammar and lexicon. There are over 200 sign languages with millions of speakers [10]. ASL is primarily used in the USA and parts of Canada.

The majority of ASL signs involve the motion of one hand that is dominant including fingers and the other hand can also be a part of some ASL signs to complement the meaning. Fig. 2a shows the hand poses for fingerspellings (FS) of A, S, and L. Fig. 2b and Fig. 2c show hand motions involved in signing "eat" and "bike".



(a) ASL Finger-spelling [6]

(b) ASL sign: Eat [8]

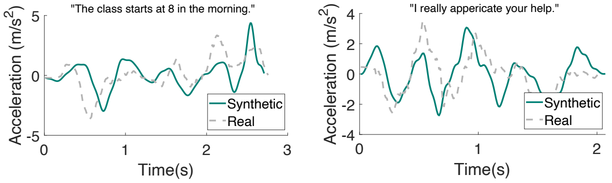
(c) ASL sign: Bike [7]

Figure 2: Examples of hand poses and motions in ASL

Moreover, facial grammar (by facial expressions) can be used to complement hand gestures. Eyebrows are raised to ask a yes/no question, show emphasis, etc. The entire signing motion including hands and facial grammar is denoted as *gloss* in linguistic terms. For simplicity, we use the *word* to represent the meaning (eat, bike, etc.) of the *gloss* and *sign* to represent the actual motion.

3.2 IMU Data and Virtual IMU Generation

IMU Data Introduction. IMU sensors are popularly used in motion tracking applications in AR/VR, sports analytics, smart healthcare, etc. An IMU primarily consists of an accelerometer, a gyroscope, and a magnetometer. An accelerometer measures the sum of acceleration and gravity vector. A magnetometer measures the direction of the magnetic field, whereas the gyroscope sensor measures the angular velocity. The accelerometer, magnetometer, and gyroscope sensors conduct the measurements in the *local frame* of reference of the sensor. To convert the measurements to the *global frame* (relative to earth), the orientation [103] of the sensors is first determined that computes the rotation of the sensor relative to the *global frame*. Briefly, *SignQuery* adopts opportunistic calibration techniques [103] and complementary filters to estimate orientations. *SignQuery* adopts A3 [103] to opportunistically select measurements from magnetometers and accelerometers when they are free of magnetic interference or motion artifacts, and fuses them with gyroscope measurements, thus periodically resetting drifts in gyroscope integration, as well as handling effects of magnetic interference and motion artifacts. *SignQuery*' experiments are conducted in environments with magnetic interference from objects such as metallic doors, furniture, etc. Given the opportunistic fusion of sensors, similar to A3, we do not observe the effects of environmental artifacts or long-term drift. More details are elaborated in A3 [103].



(a) IMU readings from right hand, middle finger, x axis

(b) IMU readings from right hand, middle finger, y axis

Figure 3: **Synthesized virtual IMU data looks similar to real IMU data, indicating the feasibility of training with synthetic data.**

Virtual IMU Generation. Designing a highly accurate *SignQuery* system over a large vocabulary requires high-quality and large-scale training data. However, generating such data on wearable devices could be time-consuming and laborious and there is no public dataset that is available. Therefore, *SignQuery* leverages techniques from IMUTube [55] and ZeroNet [64] to synthesize virtual IMU data from large-scale online sign videos for training. ■ Body Size Standardization: Keypoints extracted from videos are in units of pixels, and camera parameters are needed to convert to centimeters [68]. Yet, these parameters may not be available for public videos. Also, the extracted key points from videos depend

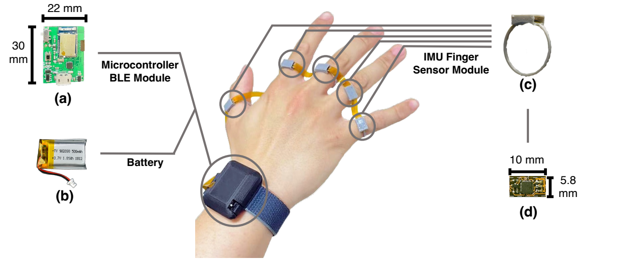


Figure 4: Design a portable wireless sensing device in *SignQuery*: (a) Microprocessor with BLE (b) Battery (c) IMU finger module (d) IMU sensor

on the user's body size. To compensate for this difference in body sizes of users, we use body size standardization adopted from [64] which allows working in the relative space without the need for camera parameters that may not be available for public videos. ■ Synthesizing Accelerometer Data: Locations of finger joints and the wrist are first extracted from videos. Then we double-differentiated the location data with finite differences to approximate the accelerometer data. Finally, we transform data to a common frame of reference (*local frame* of the wrist). ■ Synthesizing Orientation Data: In addition to the accelerometer data, deep learning models of *SignQuery* also use the orientation information extracted by the IMU. Toward this end, we estimate the orientation via the vector between the bottom finger joint and the fingertip as a direction vector to capture the orientation. Lastly, orientation data is converted to a common frame of reference (*local frame* of the wrist). Fig. 3 presents the comparisons of synthetic IMU data and real IMU. Evidently, the synthesized data looks similar to real ones. Note that IMU data for training is synthesized from video, while data for testing is collected by our sensor device (§3.3) from native ASL users (§5).

3.3 Sensor Device in *SignQuery*

Justification on Sensor Device Choice. Depicted in Fig. 4, we aim to design an unobtrusive, portable, and low-power sensor device for comfortable wearing. Alternatives like sensor gloves [11, 26, 101] are known to impede natural and dexterous finger motion [79]. Platforms like OuraRing [73] are proprietary without access to raw sensor data. Armbands like EMG [62] have high overhead in calibration and training data. In contrast, our sensor device can work anywhere (validated in §5) while allowing users to perform normal daily life activities including working on their laptops (typing, browsing, etc.), eating, drinking, etc. Our sensor device is inspired by [101] and upgraded in the following ways: (i) flexible printed circuit board (FPCB) is exploited such that a ring-shaped and lightweight sensor device is realized to enhance the flexibility when users are signing. (ii) A low-power wireless microcontroller is integrated to achieve power consumption of 32 mA (i.e., supporting an ≈16 hours battery life), which is 6.19× energy efficient than [101]. The details of our platform design are elaborated on next.

Sensor: Depicted in Fig. 4c, we embed ICM20948 [46] (Fig. 4d) which provides 9-axis IMU data (accelerometer, magnetometer, and gyroscope) on fingers and wrist.

Microcontroller: The IMU sensors are controlled by a microcontroller. To fit into a compact form factor on the wrist, we self-develop the microcontroller (Fig. 4a) based on MDBT42Q-512KV2 [78], which uses nRF52832 *System-on-Chip* (SoC) [71] with BLE, 64 MHz ARM Cortex M4F CPU, 512kB flash memory, and 64kB RAM.

PCB Design: We design finger IMU sensors in the shape of a ring using FPCB with electronics that can bend. A 2-layer PCB design is used for the wrist sensor to embed electronics on both sides of the PCB and minimize sizes. Autodesk EAGLE [4] was used for PCB design.

Communication: The microcontroller communicates with the finger sensors using *Serial Peripheral Interface* (SPI) [80] to assemble IMU data, which is streamed over BLE to a smartphone at a sampling rate of 100 Hz.

Battery: We use a 3.7V, 500mAh LiPo battery (Fig. 4b) housed within the wrist module. Overall power consumption is about 32mA for continuous streaming of sensor data, thus offering a battery life of about 16 hours.

Packaging and Weight: For the finger ring module as shown in Fig. 4c, we used *Fused Deposition Modeling* (FDM) technology [5] to build 3D printed housing with the TPU (Thermoplastic Polyurethanes) material for elasticity, flexibility, and comfort. We create several different sizes of rings to accommodate different users. The wrist module which integrates the microcontroller is enclosed by a PLA (Polylactic Acid) plastic casing for sturdiness and rigidity. Overall, the size of breakout for finger rings module and wrist module are 10 mm × 5.8 mm × 0.05 mm (W × L × H) and 22 mm × 30 mm × 1.6 mm respectively. The total weight of the device is 21.2g. The average weight of each finger sensor ring is 0.3g. The total weight of the wrist sensor module is 19.7g, including the PLA housing (6.7g), microcontroller PCB (3.3g) and the LiPo battery (9.7g). The form factor of the finger modules is comparable to a ring whereas the form factor of the wrist module is comparable to a smartwatch.

Software Framework: The software is implemented on the NRF52832 microcontroller using C++ and appropriate libraries for IMU, BLE streaming, and sensor addressing via SPI protocol [45, 84]. The microcontroller reads data from six IMUs over SPI and then streams the

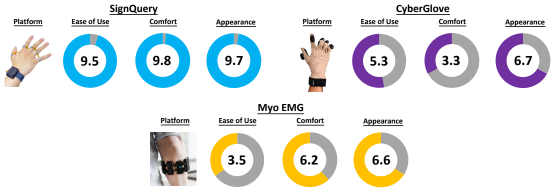


Figure 5: User experience survey on SignQuery compared with two alternative sensing devices.

data to a mobile device over BLE with a sampling rate of 100 Hz and a low-latency of 11.1 ms.

Usability Study for the Sensor Device: We conducted a usability survey to evaluate the user experience of the *SignQuery* sensor device in comparison to two other finger motion tracking devices: CyberGlove [26] and EMG [9]. Participants wore each device for three hours during regular daily activities, such as working on laptops or eating. Afterwards, they rated each device in terms of *ease of use*, *comfort*, and *appearance* on a scale from 0 to 10. The results, shown in Fig.5, indicate that CyberGlove was less favorable due to its weight, which restricted dexterity. The Myo offered better comfort but required tedious calibration and skin warming during each wear. Though single-ring sensors, like[2, 102], are another possible comparison, they don't track all fingers, hence were excluded from our study. The current *SignQuery* version stands out for its usability during daily activities. Looking ahead, as discussed in §6, we aim to harness large language models to grasp the ASL's finger-based semantic nuances, hoping to refine the *SignQuery* design into a more compact, ring-like form for greater accessibility.

4 SIGN LANGUAGE QUERY

In this section, we define retrieval tasks formally, and describe the key signal processing and deep learning modules in the underlying *SignQuery*.

4.1 Cross Model Retrieval Formulation

Given a query q, the objective of *SignQuery* is to find the best match p from a target set $\mathcal{P}$ (e.g., ASL videos, or English text). To realize this, we aim to encode a query q and its corresponding target p into a joint embedding space Ω such that the similarity of $f_Q(q) \in \Omega$ and $f_P(p) \in \Omega$ is maximized *if and only if* the target p is the best match for the query q. Specifically, we define two main tasks based on the types of queries and targets.

■ *IMU-to-Video* (**I2V**): *SignQuery* encodes IMU query signing (captured by smart rings which are worn on users' fingers, see §3.3) to find the best matching video from the joint embedding space. Formally, we have

$$v = \underset{v \in \mathcal{V}}{\operatorname{argmax}} \ \operatorname{sim}(f_I(i), f_V(v)), \tag{1}$$

where $\mathcal{V}$ denotes a set of sign videos, $\operatorname{sim}(i, v) = i \cdot v$ denotes the dot product of two embeddings as both are unit-normalized, i and v denote IMU and video respectively, and $f_I(\cdot)$ and $f_V(\cdot)$ denote the encoding functions for IMUs and videos respectively.

■ *Text-to-Video* (**T2V**) : *SignQuery* also supports *Text-to-Video* queries for Deaf users who are bilingual with ASL and English. Similarly, we define the task as follows.

$$v = \underset{v \in \mathcal{V}}{\operatorname{argmax}} \ \operatorname{sim}(f_T(t), f_V(v)), \tag{2}$$

where t and v denote text and video respectively, and $f_T(\cdot)$ and $f_V(\cdot)$ denote the encoding functions for text and videos respectively.

Although not our main goal, as a byproduct of the above formulation with the joint embedding space, *SignQuery* can also perform secondary tasks such as *Video-to-IMU* (**V2I**) and *Video-to-Text* (**V2T**) where we retrieve IMU and text matchings in response to a video query. Moreover, *IMU-to-Text* (**I2T**) and *Text-to-IMU* (**T2I**) where we retrieve text (or IMU) matchings in response to an IMU (or text) query, are supported for completeness.

4.2 Learning Network of *SignQuery*

An ASL user submits queries to search engines using wearable IMU data. In response to the query, ASL videos (or text) from a large online database are searched to obtain relevant videos (or text) corresponding to the query. To facilitate matching between IMU query, video, and text data which are inherently in different modalities, we encode them into a joint representation space to promote accurate searching and matching. As illustrated in Fig. 6 and described in §3.2, *SignQuery* extracts virtual IMU data from sign videos. Later, sign videos, the corresponding IMU, and text are encoded into a common representation space that allows matching IMU queries with relevant videos or text. Next, we elaborate on how we encode different modalities.

4.2.1 Video Encoding Branch f_V.

Video Simplification: To encode a video, one naive idea is to directly apply a video encoder (e.g., I3D [22], Video Transformer [14]). However, these models are tailored for tasks such as video understanding, and video classification. Directly using those weights may result in discrepancy because a sign video mainly contains a user signing in the air, which is semantically simple, while videos (e.g., a man riding a bike in a park, a dog chasing a cat on the grass) are semantically richer. Also,

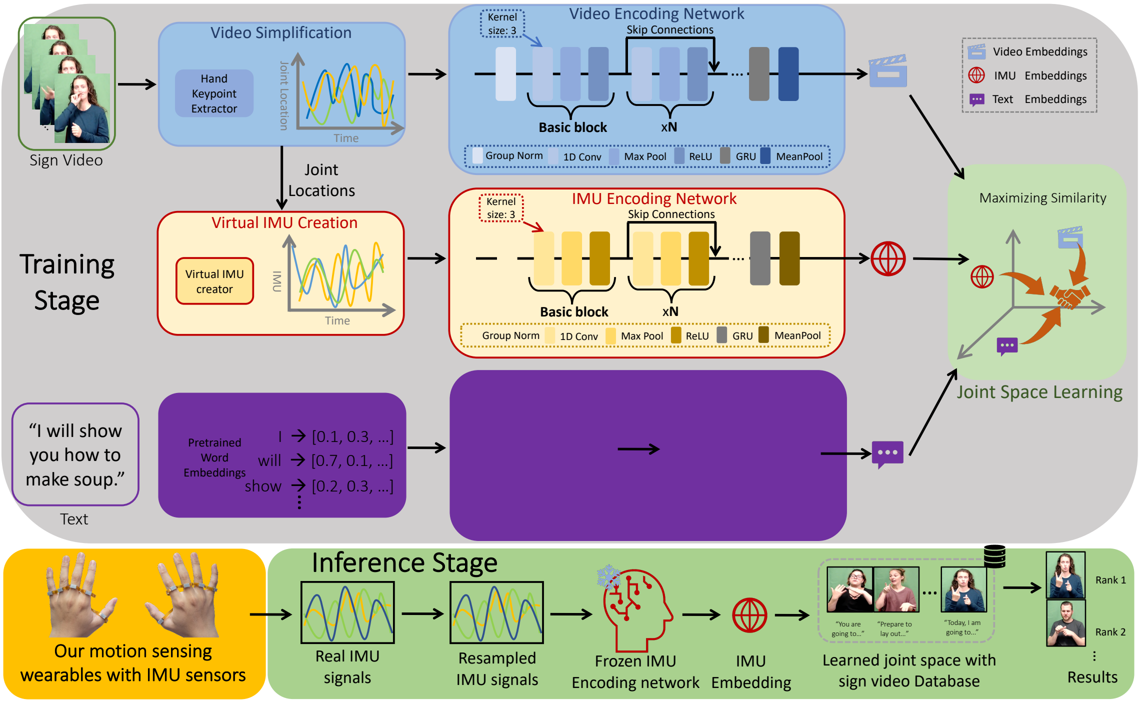


Figure 6: *SignQuery* **Overview. In training, Video, IMU, and Text are embedded into a joint embedding space such that the similarity of the same contents from different modalities is maximized. In inference, we freeze the IMU encoding network and rank all available sign videos in the database given the input IMU query. We then recommend the highest-ranking videos. Note that we show the I2V task due to space limitation, we provide other retrieval results in §5.**

the salient information (e.g., hand motions) can be easily ignored since sign videos contain much redundancy and insignificant information (e.g., background and appearance of signers) [24], leaving training from scratch challenging. Moreover, while the IMU signing data in *SignQuery* is captured by smart rings worn on fingers (§3.2), sign videos contain information not only from hands, but also from the signer's head movement, body movement, and facial expressions. As shown in Fig. 7, when two different users sign for the same sentence, their poses and facial expressions are different, while hand motions are quite similar. Therefore, we propose the *Video Simplification* technique that reduces the information discrepancy between video and IMU. Specifically, each video will be preprocessed by an off-the-shelf hand keypoint extractor (*Google MediaPipe* [96]), then we transform the locations of fingers with respect to wrists according to §3.2. To match the information from smart rings (which are worn on the bottom of fingers), we further keep locations from the bottom joints of all fingers.

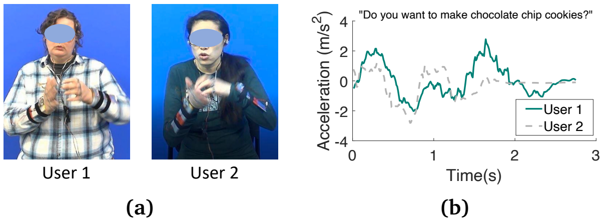


Figure 7: (a) Users are signing for the same sentence, and their upper body poses and facial expressions are different. (b) Hand motions captured by IMU sensors are similar.

Video Encoding Network Ψ_v: As shown in Fig. 6, the proposed video encoding network Ψ_v builds upon the success of *ResNet* [41], and we replace all 2D operations with 1D operations (e.g., convolutions, normalization). In detail, the location data $v \in \mathbb{R}^{t \times n}$ (t denotes time and $n = 2 \times 5 \times 3$ denoting the location data from two hands, each with five joints, and each joint with three axes) is normalized by a *GroupNorm* operation [93] independently along with feature dimension, then goes through a series of blocks of 1D-Conv with skip connections, pooling, and activation functions in between, and a *GRU* [29] is connected at the end to learn temporal information. Finally, a mean pooling operation is used to generate an embedding $\Psi_v(v) \in \mathbb{R}^d$.

4.2.2 IMU Encoding Branch f_I.

Virtual IMU Generation: Note that there is no large-scale public training data currently available for wearable sensors with ASL, resulting in the impossibility of training a joint embedding space for cross-model retrieval. Toward this end, *SignQuery* synthesizes IMU training data from a sign video dataset (*How2Sign* [32], more details in §5.2) and that is sufficient to train a joint embedding space. As shown in Fig. 6, after the location is obtained from the same hand keypoint extractor, the data is then double-differentiated with a finite time window. The virtual IMU is thus generated. More details can be found in §3.2, ZeroNet [64], and IMUTube [55] as we adopt techniques from these papers to generate virtual IMU data as well as handling domain shift between real and virtual IMU data. Finally, as also depicted in the figure, during the inference stage, the real IMU data can be used for making inferences with the model trained and domain adapted with virtual IMU data.

IMU Encoding Network Ψ_i: Similar to Video Encoding Network Ψ_v, we propose to use a similar architecture for IMU encoder Ψ_i. The input vector dimension is $n = 2 \times 10 \times 3$ denoting the IMU data from two hands, each hand with five sensor rings, and each ring with three axes of acceleration and orientation data. The output of the IMU Encoding Network is $\Psi_i(i) \in \mathbb{R}^d$.

4.2.3 Joint Space Learning and Loss Function. Inspired by the marginal ranking loss, the Max of Hinge (MH) loss penalizes the model according to the negatives closest to each training query [35]. Formally, given a pair of (i, v), the hardest negatives for IMU and video are given by $i^H = \text{argmax}_{j \neq i} \text{sim}(j, v)$ and $v^H = \text{argmax}_{j \neq v} \text{sim}(i, j)$ respectively. Thus, the MH loss is defined as

$$\begin{aligned} \mathcal{L}_{MH} = \max_{v^H} [\alpha + \text{sim}(i, v^H) - \text{sim}(i, v)]_+ \\ + \max_{i^H} [\alpha + \text{sim}(i^H, v) - \text{sim}(i, v)]_+ \end{aligned} \quad (3)$$

where $[x]_+ \equiv \max(0, x)$ and α denotes the margin. From Eq. 3, one can see that the margin is fixed along the training process. However, we empirically found that a varying margin leads to better performance. Thus, we define a Restricted Max of Hinge (RMH) loss as follows.

$$\begin{aligned} \mathcal{L}_{RMH} = \max_{v^H} [\alpha(t) + \text{sim}(i, v^H) - \text{sim}(i, v)]_+ \\ + \max_{i^H} [\alpha(t) + \text{sim}(i^H, v) - \text{sim}(i, v)]_+ \end{aligned}$$

$$\alpha(t) = \begin{cases} 0 & t < 3 \\ 0.05 & t \geq 3 \end{cases} \quad (4)$$

where t denotes training epochs. We found this trick helpful because, during the first few epochs, the learned embeddings are not stable, thus leading to inaccurate v^H or i^H. Therefore, we restrict the hardest negative selection, then release the restriction when models are more stable. We empirically validated the design in §5.

4.2.4 Text Encoding Branch f_T. Although Deaf users primarily use sign language, *SignQuery* still provides text queries for users who are bilingual with both sign language (ASL) and spoken language (English). To encode a text query into the joint embedding space of IMU and video, we adopt similar processing steps in [31] for the text encoder f_T which consists of three components: (i) pre-trained word embeddings to encode individual words; (ii) the *NetVlAD* [13] to learn relation among word embeddings; (iii) and a simple fully-connected layer to get a fixed-length vector $f_T(t) \in \mathbb{R}^d$. Note that since minimizing the loss function (Eq. 4) for all three modalities at the same time is not applicable, we take advantage of CLIP [76] training scheme: a joint embedding space for IMU and video is first trained using the proposed loss, then we embed text into the same joint space with a frozen IMU encoding network, supervised by the loss in Eq. 4. In the end, *SignQuery* supports query inputs in form of signs (IMU) or text per users' preference.

Table 1: Dataset Property Comparisons between *How2Sign* (train/evaluation set) and Our User Study (testing set). Note that the signers from these two datasets have no overlapping. Note that our study is across months and the training and testing data were collected years apart.

Properties	*How2Sign* [32] (Train/Evaluation)	*SignQuery* (Ours) (Test)
Language	ASL	ASL
Duration (h)	79	24
No. of Signers	11	12
No. of Sentences	24109 / 3178	2571
Modality	Video, Text	Video, Text, IMU

5 EVALUATION

5.1 Implementation

SignQuery is implemented on a combination of desktop and smartphone devices. Our deep learning models are implemented with Pytorch [75] library and the training is performed on a desktop with Intel i7-8700K CPU, 16GB RAM memory, and an NVIDIA Quadro RTX 8000 GPU. We set the dimension of embeddings, d, to 768, and the number of convolution layers, N, to 4. We use the Adam optimizer [53] with a batch size of 128 and a learning rate of 3e-4. To avoid overfitting issues that may happen in the training process, we apply the L2 regularization [17] with a parameter of 0.02 and also add dropouts [89] with a parameter of 0.4. Note that we opt for the parameters based on simple grid search methods. Once a model is generated from training, the inference is done on a smartphone device using Pytorch Mobile [34] on Samsung S20 and OnePlus 9 Pro smartphones.

5.2 Datasets, and Evaluation Protocols

Dataset for Training. In *SignQuery*, we derive from a recently published ASL dataset *How2Sign* [32], a multimodal dataset with videos and corresponding translations in English. *How2Sign* consists of 79 hours of continuous instructional videos in ASL, which cover a wide variety of topics such as sports, entertainment, personal care, education, home and garden. We use videos and translations for training and evaluating our deep learning model. The training and evaluation sets consist of 24109 and 3178 data points respectively, and overall, the vocabulary size is 15,896. To extract hand keypoints, we use *Google MediaPipe* [96] that estimates keypoints based on hand anatomical constraints. We believe it is sufficient to represent signs, thus making training a reliable joint embedding space possible.

Dataset for Testing. To validate *SignQuery*, we conduct a study with 12 Deaf users with native ASL fluency (6 females, 6 males). The study has been approved by the IRB committee. The users are aged between 20-50 and weigh between 45-96kgs. During the study, sign videos are played and users are required to understand and re-sign the same content while wearing the sensor device on both hands with the sensor snugly fit on the fingers as shown in Fig. 4. Each user takes 3 breaks with removing and remounting the sensor device in between and the total study time per user is two hours. We summarize training (and evaluation), and testing datasets in Table 1.

Metrics of Evaluation. To evaluate retrieval performance, we follow the protocols used in the existing retrieval literature [31, 60, 66] and report standard metrics recall@K (recall at rank K, R@K, higher is better) and Median Rank (MedR, lower is better). For example, recall@10 being 30 indicates that 30% of the top 10 recommendations by *SignQuery* in response to the user's query are relevant.

Tasks of Evaluation. Although we have six tasks with IMU, video, and text modalities, since the objective is to verify the viability of using signs as queries directly, we mainly evaluate *SignQuery* using IMU and video, namely the retrieval task of IMU-to-Video (I2V) by default. We show other tasks briefly for completeness.

5.3 Performance Evaluation

To assess the performance of *SignQuery*, we conduct the following analysis: ■ We first evaluate the retrieval performance on the *How2Sign* database, which validates our idea of providing a natural way for Deaf users to search online. ■ We then conduct robustness studies to characterize the performance of *SignQuery* under various situations such as dialect, race, gender, signing speed, ages. ■ We also perform an ablation study to validate the design choice in *SignQuery*. ■ We compare *SignQuery* with the state-of-the-art method on the same retrieval tasks. ■ We also provide qualitative results from different tasks to show the retrieval performance of *SignQuery*. ■ Finally, we evaluate power consumption and latency of *SignQuery* on smartphones.

Table 2: ***SignQuery*** **performance of all tasks on** ***How2Sign*****. Note that** ***eval*** **denotes the evaluation set from** ***How2Sign*****, while** ***test*** **denotes all data in the test set from our user study (§5.2). I, V, and T denote IMU, Video, and Text respectively. For example, I2V denotes the IMU-to-Video task where we retrieve relevant video matchings in response to an IMU query.**

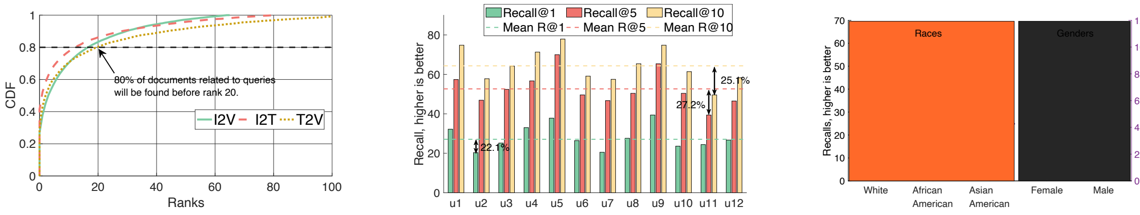
Tasks	R@1↑		R@5↑		R@10↑		MedR↓	
	eval	test	eval	test	eval	test	eval	test
I2V	32.1	28.1	60.4	52.7	70.5	64.3	3.5	5.5
I2T	33.4	31.6	63.6	55.8	78.6	67.5	3.0	4.3
T2V	31.5	28.0	71.5	53.4	78.7	60.8	3.0	5.0
V2I	33.8	29.7	65.9	57.3	76.7	69.6	3.0	4.0
T2I	33.6	29.2	74.2	57.2	84.6	67.7	2.0	4.0
V2T	34.6	28.9	75.1	52.5	78.1	64.1	3.0	5.5

5.3.1 *Retrieval Results on How2Sign.*

Table 2 describes the overall performance of *SignQuery* on all six tasks that we defined in §4 with the *How2Sign* dataset. Evidently, *SignQuery* returns good retrieval results on both evaluation and test sets, which suggests that the quality of virtual IMU generation is sufficient to compare with real IMU data, despite the minute difference between them. Furthermore, the performances of I2V and T2V are close (e.g., 28.1 vs 28.0 for recall@1, and 64.3 vs 60.8 for recall@10), indicating directly using signs as queries is a viable and better solution for Deaf users considering that the sign language is the native language and therefore, they may express their ideas naturally. Moreover, Fig. 8a depicts the overall rank distributions on three tasks. Clearly, the retrieval performance of these tasks is comparable. Interestingly, one can see that 80% of documents related to the queries are found before rank 20, suggesting that *SignQuery* can retrieve the most relevant items within the first 20 results for all queries. We believe this is sufficient because normally people would at most check the first 15-20 results before they update their queries. These results are comparable with state-of-the-art recommendation systems for Amazon, Netflix, and Yelp dataset [23, 58], suggesting the usability of *SignQuery* in the real world.

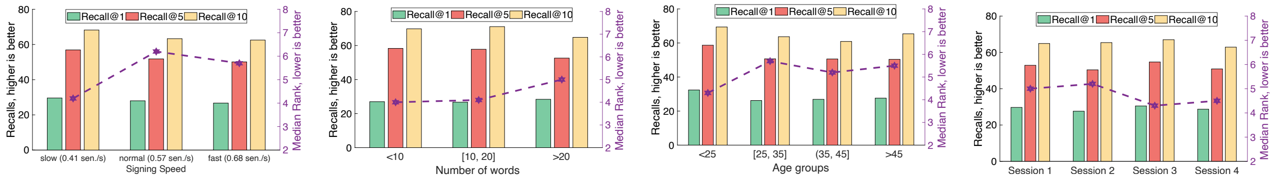
5.3.2 *Robustness Study.*

Generalization to New Users. Fig. 8b depicts the variation in retrieval performance of I2V across users. Our inspection of the variations indicates that some users



(a) Rank Distributions (b) Users (c) Races and Genders

Figure 8: (a) Retrieval rank distribution; More than 80% of documents that are related to queries can be found before rank 20. (b) *SignQuery* performs well with new users. (c) *SignQuery* can be generalized to different races and genders.



(a) (b) (c) (d)

Figure 9: *SignQuery* provides a stable retrieval performance over (a) signing speeds (b) sentence lengths (c) age groups (d) sensor positions and orientations.

have variations or dialects. We observed different sign patterns for some users, which we believe is the main source of the variations. Nevertheless, for recall@1, user 2 has the worst performance, which is 22.1% off from the average performance and the accuracy is pretty stable across users. Note that we only use synthetic training data and yet the accuracy is consistent across completely new and real test users. Therefore, we believe *SignQuery* provides a stable retrieval performance on various users with diversity in gender, body masses, sizes, etc.

Robustness to Races and Genders. Fig. 8c describes the retrieval performance of *SignQuery* on different races and genders. Overall, we believe *SignQuery* provides stable performance over different races and genders. Asian Americans perform slightly worse. We believe this is probably because the *How2Sign* dataset has very few Asian Americans which just represents the generic sample of the population in the USA. Nevertheless, the absolute numbers are reasonable for a recommendation and query system, indicating the ability of *SignQuery* to adapt to unseen races. *SignQuery* can also perform stable across genders. In a nutshell, we believe *SignQuery* has the ability to generalize to both races and genders.

Robustness to Speed. To validate the impact of signing speeds, Based on text lengths and the corresponding video lengths of the collected data, we roughly categorize all users into three groups, i.e., slow (0.41 sentences/second or 86.1 words/minute), normal (0.57 sentences/second or 119.7 words/minute), and fast (0.68 sentences/second or 142.8 words/minute). . Evidently, as shown in Fig. 9a, *SignQuery* can adapt to different signing speeds. We believe this is due to the model architecture where multiple 1D convolution blocks are used and connected by skip connections similar to ResNet [41] as features at different levels are then integrated together to account for the information calculated from different speeds.

Robustness to Various Sentences Lengths. Fig. 9b shows the variation as a function of the length of the sentence. *SignQuery* is also adapted to different sentence lengths and the retrieval performance does not degrade with the increasing sentence lengths. This is because our training data consists of different lengths of sentences. Therefore, *SignQuery* can easily adapt to various lengths.

Robustness to Age Groups. Due to the language evolving, users of different ages may use signs differently (e.g., some users may do a sign twice to emphasize, while others may not). To validate *SignQuery* is robust to such variations, we split users into four age groups as shown in Fig. 9c. Evidently, *SignQuery* is able to deal with a wide span of ages. We believe this is because in addition to diverse training data, the sequence learning module employed in *SignQuery*, even if some signs are emphasized/missing, based on the context, *SignQuery* can infer the meaning of signs.

Robustness to Sensor Position and Orientation. Fig. 9d depicts the variation in accuracy across sessions. Note that during user study, we remove and remount the sensor devices when users have breaks. Hence, we can validate any effects of changes in sensor position or orientation with respect to the human body. Evidently, the retrieval performance is stable across four sessions. This is because the sensor devices fit snugly to both hands, and any minor variation in positions/orientation across breaks is typically much smaller than the hardware noise floor, thus having a negligible impact on the accuracy.

Table 3: ***SignQuery*** **can adapt to different surroundings and potential magnetic interference.**

Locations	Surroundings	R@1↑	R@5↑	R@10↑
Conf. Room	projectors	25.8	46.3	52.7
Studio	lights,cameras	27.5	49.3	58.7
Hall	empty	28.0	53.4	60.8

Robustness to Magnetic Inferences. Since our user study was conducted at three different places and each place has different surroundings (e.g., tables, chairs, lights, projectors, and cameras), potentially resulting in magnetic inferences, we summarize the retrieval performance when data was collected at these places in Table 3. Evidently, *SignQuery* is not affected by the surroundings thanks to the opportunistic error compensation strategies from A3 [103], making *SignQuery* ubiquitous.

5.3.3 *Ablation Study.*

In this ablation study, we focus on the main designs of *SignQuery*, i.e., video simplification, $\mathcal{L}_{RMH}$, and sequence modeling. For other design choices (e.g., temporal aggregation methods), we only provide the design consideration as follows. While more sophisticated pooling (compared to mean pooling) methods can be applied, *SignQuery* is opt for simple and efficient methods which have been validated in other retrieval work [30, 66].

Effectiveness of Video Simplification. We conduct an ablation study for video simplification with the tasks of **I2V** and **T2V** as shown in Table 4. For the model without video simplification, we extract features directly from videos using pretrained I3D [22] weights, then the rest part is the same as *SignQuery* (i.e., sequence learning with GRU, and $\mathcal{L}_{RMH}$). Unsurprisingly, we observe 21.6%, 21.4%, and 22.0% improvement for recall@1, recall@5, and recall@10 respectively, when compared with the full version of *SignQuery*. This is because of the following reason: information from sign videos is redundant and even irrelevant (e.g., backgrounds, appearances of signers). Therefore, the embeddings with such noisy information are inaccurate in common representation space. In contrast, *SignQuery* employs the video simplification technique that makes the model learn from the most direct information sources. Despite sign language being complex, hand motions are always the first cues to check to fully understand the meaning.

Effectiveness of $\mathcal{L}_{RMH}$. Compared to the Max of Hinge loss that penalizes the model according to the most negative samples for each query), we conducted an ablation study for the Restricted Max of Hinge loss we proposed to mitigate the effect of unstable embeddings at the beginning of training time. As shown in Table 4, $\mathcal{L}_{RMH}$ brings improvements on all three tasks. As stated in §4, the learning process typically is not stable, leading to inaccurate embeddings for finding the most negative samples in the Max of Hinge Loss. Therefore, instead of finding the hardest negatives with some margins, we set the margin to zero to restrict the selection and after a few epochs, we release the restriction by setting margins back. We found this trick helpful to stabilize the training process, thus improving the retrieval performance (e.g., 10.6%, 4.6%, and 2.9% improvement for recall@1, recall@5, and recall@10 respectively).

Table 4: **Study the effectiveness of *Video Simplification* and $\mathcal{L}_{RMH}$ (§4).**

	I2V			
Settings	R@1↑	R@5↑	R@10↑	MedR↓
w/o Video Simpli.	23.1	43.4	52.7	10.7
w/o $\mathcal{L}_{RMH}$	25.4	50.4	62.5	6.0
SignQuery	28.1	52.7	64.3	5.5
	I2T			
Settings	R@1↑	R@5↑	R@10↑	MedR↓
w/o $\mathcal{L}_{RMH}$	31.1	52.9	62.0	5.0
SignQuery	31.6	55.8	67.5	4.3
	T2V			
Settings	R@1↑	R@5↑	R@10↑	MedR↓
w/o Video Simpli.	25.8	46.3	52.7	6.5
w/o $\mathcal{L}_{RMH}$	27.5	49.3	58.7	6.0
SignQuery	28.0	53.4	60.8	5.0

Table 5: **The impact of sequence modeling on performance.**

Sequence Modeling	R@1↑	R@5↑	R@10↑	MedR↓
No Modeling	12.4	31.1	44.7	13.5
LSTM [42]	27.4	53.5	63.1	6.0
bi-GRU	28.2	53.3	65.2	5.0
GRU [29]	28.1	52.7	64.3	5.5

Impact of Sequence Modeling Methods. To assess the quality of the sequence learning module in *SignQuery*, we compare the design choice GRU [29] with other alternatives such as *No Modeling* (i.e., the sequence of features is aggregated by mean pooling directly), LSTM [42], and bi-directional GRU. We report the performance of **I2V** in Table 5. Evidently, sequence modeling brings a great improvement compared to the *No Modeling*. However, we only observe marginal differences in the other methods, thus we chose GRU by considering the tradeoff between performance and sizes of the parameters. We plan to use sophisticated sequence modeling (e.g., Transformers [88]) in future research.

5.3.4 *Comparing with the State-Of-The-Art.*

To show the effectiveness of *SignQuery*, we compare it with a vision-based method *SPOT-ALIGN* [31] which is the first work to propose the **T2V** retrieval task in a

Table 6: Comparison with *SPOT-ALIGN* [31]. Note that since there is no public code for *SPOT-ALIGN*, we re-implement the idea based on the description of the paper. We present the improvement in red and the decrement in blue. We also present the performances reported in *SPOT-ALIGN* in "()" with gray color. Evidently, *SignQuery* outperforms *SPOT-ALIGN* for most of the metrics.

	Metrics	*SPOT-ALIGN* [31]	*SignQuery*	diff (%)
T2V	R@1↑	22.4 (32.8)	28.0	+25.0(-14.6)
	R@5↑	46.3 (47.7)	53.4	+15.3(+11.9)
	R@10↑	63.2 (52.9)	60.8	-3.8(+14.9)
	MedR↓	6.0 (7.0)	5.0	+16.7(+28.5)
V2T	R@1↑	23.1 (23.3)	28.9	+25.1(+24.0)
	R@5↑	47.4 (48.5)	52.5	+10.8(+8.2)
	R@10↑	53.7 (53.7)	64.1	+19.4(+19.4)
	MedR↓	7.0 (7.0)	5.5	+21.4(+21.4)

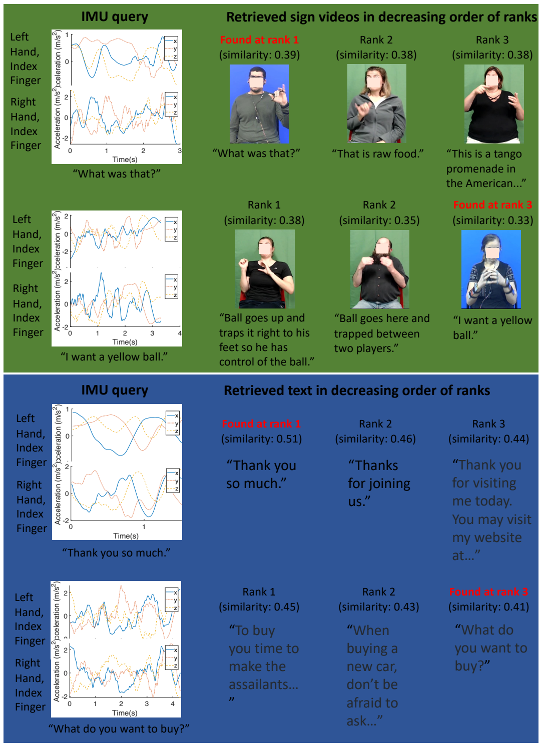


Figure 10: Qualitative results on IMU-to-Video and IMU-to-Text. For each query, we show the query in form of IMU signals from the left and right hand (only index finger due to space limitation) and text (on the bottom of the IMU signal, and the text is not seen during training, only for visualization). For IMU-to-Video, we show frames from the top 3 ranked video as well as their corresponding text (text and videos that are found at ground truth ranks are not used during retrieval, only for visualization purposes). For IMU-to-Text, we show the top 3 ranked text. Evidently, retrieved items are relevant to the queries.

sign video database. As it only has video and text, we, therefore, compare the retrieval performance of *SignQuery* on **T2V** and **V2T** tasks. As depicted in Table 6, *SignQuery* outperforms *SPOT-ALIGN* on most of the metrics. For **T2V** and **V2T**, we observe ≈25% improvement on recall@1 because *SignQuery* simplifies sign videos by focusing on important information. We believe the performance of *SignQuery* is promising because it provides a natural, ubiquitous, and privacy-preserving solution while being agnostic to lighting, background, and other ambient conditions.

Table 7: Justification of joint embedding space with the task of I2V. *translation-based* denotes both queries and targets need to be converted to text before matching, while *SignQuery* performs direct match in a joint embedding space.

Method	R@1↑	R@5↑	R@10↑	MedR↓
translation-based	16.2	32.5	47.3	17
SignQuery	28.1	52.7	64.3	5.5

5.3.5 *Justification for the proposal of joint embedding space.*

It's crucial for *SignQuery* to facilitate native sign language queries, upholding equitable access principles [19]. While some Deaf users might opt for translation-based methods [20, 21], these demand converting signs and sign videos into text, often leading to data loss. Unlike these systems, *SignQuery* utilizes a joint embedding space where signs and sign videos converge in a shared space. This ensures direct and accurate searches based on similarity (i.e., relevant documents with the same or similar contextual meanings will be closer in the shared representation space). To justify *SignQuery*, we compare with translation-based search as shown in Table 7. Evidently, *SignQuery* with direct search returns documents that are more relevant than that of the method of translation, justifying our choice of joint embedding space.

5.3.6 *Qualitative Results.*

Fig. 10 depicts retrieval results qualitatively. Since our main objectives are using IMU queries directly, we present the query results for **I2V** and **I2T**. Evidently, *SignQuery* accurately retrieves the corresponding target given queries. Even when the queries are vague, *SignQuery* can return the items that look relevant to the query due to the joint embedding space for IMU, video, and text. Overall, we believe these results are encouraging in the context of applications in query systems for Deaf users.

5.3.7 *Power Consumption and Latency.*

The power consumption of the sensor device itself was discussed in §3. Here, we analyze the power consumption of the deep learning model of *SignQuery* when implemented on smartphones using Batterystats and Battery Historian [12] tools. Since users' signing can be short or long, we analyze the power consumption based on sentence lengths. As depicted in Fig. 11, continuous processing discharges at the rate of ≈12% on both phone models. And the average latency of execution of the deep learning model for different lengths on Samsung S20 and Oneplus 9 Pro are 30.0 ms and 41.1 ms

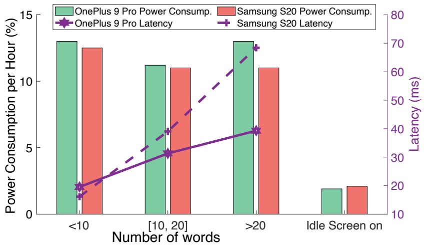


Figure 11: **Power consumption and latency analysis on different models with different sentence lengths.**

respectively, suggesting the real-time processing ability of *SignQuery*. Furthermore, when only the screen is on, the discharge rate is only 2.1% and 1.9% for Samsung S20 and Oneplus 9 Pro respectively.

6 DISCUSSION

Applications Leveraging Large Language Models. Large Language Models (LLMs) like ChatGPT have revolutionized natural language processing. While *SignQuery* currently retrieves documents using signed queries, its integration with LLMs can open up more applications. Current sign-to-text translation systems have limited domains, leading to accuracy issues. However, using *SignQuery* with diverse training can serve as a starting point for improved translations, especially when combined with the analytical prowess of LLMs. Additionally, the data from *SignQuery* can aid in creating sign language animations, beneficial for both the Deaf community and learners of sign language. In essence, combining *SignQuery* with LLMs offers vast potential.

Decreasing Form Factor. As stated in §3.3, the sensor device of *SignQuery* was selected out of many candidates (e.g., Armbands, gloves) based on the requirements of being unobtrusive, portable, and low-power for comfortable wearing. To further improve the accessibility of *SignQuery*, we plan to learn the structure of semantic meaning of ASL between fingers, inspired by how BERT [28] explored semantic meaning between words. We also plan to leverage LLMs such as *ChatGPT* to refine the accuracy of premature translation of sign languages. We believe this opens up opportunities to further decrease the size of the sensing device by only using a few sensors or even a pair of rings and watches, thus increasing the accessibility of *SignQuery*.

Employing Advanced Algorithms to Make *SignQuery* Efficient. *SignQuery* incorporated simple but effective deep learning algorithms, such as ResNet [41] and Gated Recurrent Unit [29]. Future research aims to implement more advanced algorithms that possess powerful feature learning and sequence modeling capabilities, to further enhance the system’s performance. While advanced algorithms like Transformers [88] typically offer good performance, their deployment cost on mobile devices could be high due to the large number of parameters (in the millions, or more) required. To address this challenge, *SignQuery* intends to utilize model compression techniques [27, 39] to achieve a balance between performance and energy consumption. Additionally, *SignQuery* plans to optimize its current search strategy, which involves a matrix multiplication of the query with all existing target embeddings, to identify relevant items based on calculated similarities. This strategy can be enhanced by incorporating advanced approximation algorithms (e.g., approximated nearest neighbor [49, 69]) to improve search speed, thus making *SignQuery* more efficient.

7 CONCLUSION

SignQuery realized a search engine that supports sign language queries in native form, which is essential to provide the Deaf community equitable access to highly relevant information for applications like recommendations, learning, advertisements, healthcare, and entertainment. *SignQuery* proposed a deep learning algorithm to transform wearable sensor data, video, and text into a joint embedding space for measuring similarity and ranking the items in the database when the modality of querying (wearable sensor data) is different from the modality of targets (videos, text). To reduce the training overhead, *SignQuery* leveraged the idea of synthetic training and derived virtual training data from sign language videos in ASL on diverse topics. Evaluated by a comprehensive user study with native Deaf users, *SignQuery* achieved the recall@10 of 64.3%, namely, among the top ten search results, six of them are relevant to the search query. Moreover, *SignQuery* was robust to users with diversity in dialects, ages, weight, signing speed, etc. While the performance is encouraging, we believe this opens ample opportunities for future research in several areas, especially in the area of ASL query using wearable sensors.

ACKNOWLEDGEMENTS

Our gratitude goes to the anonymous reviewers and shepherds for their invaluable feedback. We are thankful to the Penn State Affirmative Action Office and Jessica Dehaut for providing us with ASL interpretation services during the course of the project. We are also grateful to Timothy Miller for his assistance during our user studies. This research has been partly funded by NSF grants: CAREER-2046972, CNS-1909479, and SHF-2211018.

REFERENCES

[1] [n. d.]. How many deaf people are there in United States. https://research.gallaudet.edu/Demographics/deaf-US.php.

[2] [n. d.]. Oura Ring: The most accurate sleep and activity tracker. https://ouraring.com/.

[3] 2017. Studying Language Acquisition in Deaf Children. https://www.bu.edu/articles/2017/asl-language-acquisition/.

[4] 2021. EAGLE (program) - Wikipedia. https://en.wikipedia.org/wiki/EAGLE_(program).

[5] 2021. Fused filament fabrication - Wikipedia. https://en.wikipedia.org/wiki/Fused_filament_fabrication#Fused_deposition_modeling.

[6] 2023. ASL Finger Spelling. https://en.wikipedia.org/wiki/American_Sign_Language.

[7] 2023. ASL Sign for BIKE. https://www.lifeprint.com/asl101/pages-signs/b/bicycle.htm.

[8] 2023. ASL Sign for EAT. http://www.lifeprint.com/asl101/pages-signs/e/eat.htm.

[9] 2023. Myo Armband. https://learn.adafruit.com/myo-armband-teardown/inside-myo.

[10] 2023. World Federation of the Deaf. https://wfdeaf.org/our-work/.

[11] 5DT Data Glove Ultra - 5DT [n. d.]. https://5dt.com/5dt-data-glove-ultra/.

[12] Android Developer 2021. Profile battery usage with Batterystats and Battery Historian. https://developer.android.com/topic/performance/power/setup-battery-historian.

[13] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5297–5307.

[14] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6836–6846.

[15] Benjamin J Bahan. 1997. Non-manual realization of agreement in American Sign Language. (1997).

[16] Andrew L Beam and Isaac S Kohane. 2018. Big data and machine learning in health care. *Jama* 319, 13 (2018), 1317–1318.

[17] Mario Bertero, Christine De Mol, and Giovanni Alberto Viano. 1980. The stability of inverse problems. In *Inverse scattering problems in optics*. Springer, 161–214.

[18] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. 2018. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*.

[19] Rebecca M Callahan and Dara Shifrer. 2016. Equitable access for secondary English learner students: Course taking as evidence of EL program effectiveness. *Educational Administration Quarterly* 52, 3 (2016), 463–496.

[20] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7784–7793.

[21] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10023–10033.

[22] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[23] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 world wide web conference*. 1583–1592.

[24] Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. 2022. Two-Stream Network for Sign Language Recognition and Translation. *arXiv preprint arXiv:2211.01367* (2022).

[25] KR1442 Chowdhary and KR Chowdhary. 2020. Natural language processing. *Fundamentals of artificial intelligence* (2020), 603–649.

[26] CyberGlove Systems LLC [n. d.]. http://www.cyberglovesystems.com/.

[27] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. 2020. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proc. IEEE* 108, 4 (2020), 485–532.

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[29] Rahul Dey and Fathi M Salem. 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, 1597–1600.

[30] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9346–9355.

[31] Amanda Duarte, Samuel Albanie, Xavier Giró-i Nieto, and Gül Varol. 2022. Sign language video retrieval with free-form textual queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14094–14104.

[32] Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. 2021. How2sign: a large-scale multimodal dataset for continuous american sign language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2735–2744.

[33] Karen Emmorey. 2001. *Language, cognition, and the brain: Insights from sign language research*. Psychology Press.

[34] Facebook. 2021. Introduce lite interpreter workflow in Android and iOS. "https://pytorch.org/mobile/android/".

[35] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612* (2017).

[36] Biyi Fang, Jillian Co, and Mi Zhang. 2017. Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In *Proceedings of the 15th ACM conference on embedded network sensor systems*. 1–13.

[37] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 214–229.

[38] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. 2018. Hierarchical LSTM for sign language translation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[39] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).

[40] Aiming Hao, Yuecong Min, and Xilin Chen. 2021. Self-mutual distillation learning for continuous sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11303–11312.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[42] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[43] Jiahui Hou, Xiang-Yang Li, Peide Zhu, Zefan Wang, Yu Wang, Jianwei Qian, and Panlong Yang. 2019. Signspeaker: A real-time, high-precision smartwatch-based sign language translator. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–15.

[44] Fang Hu et al. 2020. FingerTrak: Continuous 3D Hand Pose Tracking by Deep Learning Hand Silhouettes Captured by Miniature Thermal Cameras on Wrist. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2020).

[45] Adafruit Industries. 2021. Adafruit ICM20X. https://github.com/adafruit/Adafruit_ICM20X.

[46] Invense. 2021. "ICM20948" datasheet [online]. https://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2021/10/DS-000189-ICM-20948-v1.5.pdf.

[47] Umar Iqbal et al. 2018. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*.

[48] Yincheng Jin, Yang Gao, Yanjun Zhu, Wei Wang, Jiyang Li, Seokmin Choi, Zhangyu Li, Jagmohan Chauhan, Anind K Dey, and Zhanpeng Jin. 2021. SonicASL: An acoustic-based sign language gesture recognizer using earphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–30.

[49] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[50] Wolf Kienzle, Eric Whitmire, Chris Rittaler, and Hrvoje Benko. 2021. ElectroRing: Subtle Pinch and Touch Detection with a Ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[51] David Kim et al. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *ACM UIST*.

[52] Kinect 2021. Microsoft Kinect2.0. https://developer.microsoft.com/en-us/windows/kinect.

[53] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[54] Oscar Koller. 2020. Quantitative survey of the state of the art in sign language recognition. *arXiv preprint arXiv:2008.09918* (2020).

[55] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. 2020. IMUTube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–29.

[56] Leap Motion Developer 2012. Leap Motion. https://developer.leapmotion.com/.

[57] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European conference on computer vision (ECCV)*. 201–216.

[58] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[59] Jiayang Liu et al. 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing* (2009).

[60] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487* (2019).

[61] Yilin Liu, Fengyang Jiang, and Mahanth Gowda. 2020. Finger gesture tracking for interactive applications: A pilot study with sign languages. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–21.

[62] Yang Liu, Chengdong Lin, and Zhenjiang Li. 2021. WR-Hand: Wearable Armband Can Track User's Hand. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–27.

[63] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. NeuroPose: 3D hand pose tracking using EMG wearables. In *Proceedings of the Web Conference 2021*. 1471–1482.

[64] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. When video meets inertial sensors: Zero-shot domain adaptation for finger motion analytics with inertial sensors. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 182–194.

[65] Yilin Liu, Shijia Zhang, Mahanth Gowda, and Srihari Nelakuditi. 2022. Leveraging the properties of mmwave signals for 3d finger motion tracking for interactive iot applications. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 3 (2022), 1–28.

[66] Antoine Miech, Ivan Laptev, and Josef Sivic. 2018. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516* (2018).

[67] Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. 2021. Visual alignment constraint for continuous sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11542–11551.

[68] Franziska Mueller et al. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *IEEE CVPR*.

[69] Marius Muja and David G Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence* 36, 11 (2014), 2227–2240.

[70] Rajalakshmi Nandakumar et al. 2016. Fingerio: Using active sonar for fine-grained finger tracking. In *ACM CHI*.

[71] Nordic NRF52832 [n. d.]. https://www.nordicsemi.com/products/nrf52832.

[72] World Health Organization. [n. d.]. Deafness and Hearing Loss. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[73] Oura Ring 2021. Oura Ring: The most accurate sleep and activity tracker. https://ouraring.com/.

[74] Abhinav Parate et al. 2014. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *ACM MobiSys*.

[75] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia

Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019), 8026–8037.

[76] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[77] Natalia Flores Ramírez, Amy M Lieberman, and Rachel I Mayberry. 2013. The initial stages of first-language acquisition begun in adolescence: when late looks early. *Journal of Child Language* 40, 2 (2013), 391–414. https://doi.org/10.1017/S0305000911000535

[78] Raytac MDBT42Q-512KV2 Nordic nRF52832 Bluetooth Module [n. d.]. https://www.raytac.com/product/ins.php?index_id=31.

[79] Alba Roda-Sales, Joaquín L Sancho-Bru, Margarita Vergara, Verónica Gracia-Ibáñez, and Néstor J Jarque-Bou. 2020. Effect on manual skills of wearing instrumented gloves during manipulation. *Journal of biomechanics* 98 (2020), 109512.

[80] Serial Peripheral Interface - Wikipedia [n. d.]. https://en.wikipedia.org/wiki/Serial_Peripheral_Interface.

[81] Michael Sherman et al. 2014. User-generated free-form gestures for authentication: Security and memorability. In *ACM MobiSys*.

[82] Brent Smith and Greg Linden. 2017. Two decades of recommender systems at Amazon. com. *Ieee internet computing* 21, 3 (2017), 12–18.

[83] Wei Sun, Franklin Mingzhe Li, Congshu Huang, Zhenyu Lei, Benjamin Steeper, Songyun Tao, Feng Tian, and Cheng Zhang. 2021. ThumbTrak: Recognizing Micro-finger Poses Using a Ring with Proximity Sensing. *arXiv preprint arXiv:2105.14680* (2021).

[84] Ha Thach. 2022. NRF52 BLE for Arduino. https://github.com/adafruit/Adafruit_nRF52_Arduino.

[85] Hoang Truong et al. 2018. CapBand: Battery-free Successive Capacitance Sensing Wristband for Hand Gesture Recognition. In *ACM SenSys*.

[86] Yu-Chih Tung and Kang G Shin. 2015. Echotag: Accurate infrastructure-free indoor location tagging with smartphones. In *ACM MobiCom*.

[87] Gul Varol, Liliane Momeni, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman. 2021. Read and attend: Temporal localisation in sign language videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16857–16866.

[88] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[89] Stefan Wager, Sida Wang, and Percy S Liang. 2013. Dropout training as adaptive regularization. In *Advances in neural information processing systems*. 351–359.

[90] Zhibo Wang, Tengda Zhao, Jinxin Ma, Hongkai Chen, Kaixin Liu, Huajie Shao, Qian Wang, and Ju Ren. 2020. Hear sign language: A real-time end-to-end sign language recognition system. *IEEE Transactions on Mobile Computing* 21, 7 (2020), 2398–2410.

[91] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 38–45.

[92] Erwin Wu, Ye Yuan, Hui-Shyong Yeo, Aaron Quigley, Hideki Koike, and Kris M Kitani. 2020. Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-worn Camera via Dorsum Deformation Network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 1147–1160.

[93] Yuxin Wu and Kaiming He. 2018. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.

[94] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. 2018. Artificial intelligence in healthcare. *Nature biomedical engineering* 2, 10 (2018), 719–731.

[95] Cheng Zhang et al. 2018. FingerPing: Recognizing Fine-grained Hand Poses using Active Acoustic On-body Sensing. In *ACM CHI*.

[96] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214* (2020).

[97] Qian Zhang, JiaZhen Jing, Dong Wang, and Run Zhao. 2022. Wearsign: Pushing the limit of sign language translation using inertial and EMG wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–27.

[98] Qian Zhang, Dong Wang, Run Zhao, and Yinggang Yu. 2019. MyoSign: enabling end-to-end sign language recognition with wearables. In *Proceedings of the 24th international conference on intelligent user interfaces*. 650–660.

[99] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.

[100] Hao Zhou, Taiting Lu, et al. 2023. One Ring to Rule Them All: An Open Source Smartring Platform for Finger Motion Analytics and Healthcare Applications. In *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*.

[101] Hao Zhou, Taiting Lu, Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2022. Learning on the Rings: Self-Supervised 3D Finger Motion Tracking Using Wearable Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–31.

[102] Hao Zhou, Taiting Lu, Yilin Liu, Shijia Zhang, Runze Liu, and Mahanth Gowda. 2023. One Ring to Rule Them All: An Open Source Smartring Platform for Finger Motion Analytics and Healthcare Applications. In *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*. 27–38.

[103] Pengfei Zhou et al. 2014. Use it free: Instantly knowing your phone attitude. In *ACM MobiCom*.

[104] Ronglai Zuo and Brian Mak. 2022. C2SLR: Consistency-enhanced continuous sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5131–5140.