Robust and Lightweight Decoder for Unsupervised Multi-Finger Force Predictions Towards the Internet of Medical Things-based Applications

Long Meng and Xiaogang Hu, *Senior Member, IEEE*

***Abstract*— Finger force monitoring has become increasingly prevalent in the field of the Internet of Medical Things (IoMT) as a key indicator of muscle strength and health status, facilitating remote rehabilitation and personalized health monitoring. However, existing methods are limited by inaccurate decoding performance or complex procedures when derived in a supervised manner. To address these challenges, we developed a novel unsupervised approach featuring a robust and lightweight neural-drive decoder for multi-finger force predictions. High-density surface electromyogram (sEMG) signals were recorded from the finger extensor muscles during isometric finger extension tasks. Each MU was then assigned a probability indicating its association with the target finger, based on its mean firing rates during the activation periods of individual fingers. MUs with probabilities exceeding a predefined threshold were retained for the final force prediction. Our results demonstrate that the neural-drive decoder achieved a computation time of 68.83±13.63 ms, making it suitable for real-time applications. Furthermore, our decoder outperformed the sEMG-amplitude-based approach (R^2: 0.79±0.039 vs. 0.64±0.080, root mean square error: 4.89±0.73 vs. 7.31±1.88 % of maximum force, Pearson correlation coefficient: 0.87±0.028 vs. 0.76±0.06, mean absolute error: 3.86±0.62 vs. 6.08±1.51 % of maximum force). The developed neural decoder demonstrated advantages over the state-of-the-art neural decoders in terms of accuracy, training procedures, and practicality. Additionally, our approach exhibited robust performance across various probability thresholds, data sources, and background noise, highlighting its potential for finger force monitoring applications in diverse IoMT scenarios.**

***Index Terms*—Finger force monitoring, hand function, unsupervised neural decoding, Internet of Medical Things (IoMT), finger co-activations.**

This study was supported in part by the National Science Foundation (CBET-2246162, IIS-2330862, IIS-2319139) and the Department of Defense (W81XWH2110185). *(Corresponding author: Xiaogang Hu)*.

Long Meng is with the Department of Mechanical Engineering, Pennsylvania State University-University Park, PA, USA (e-mail: lmm7405@psu.edu).

Xiaogang Hu is with the Department of Mechanical Engineering, Pennsylvania State University-University Park, PA, USA, and also with the Departments of Kinesiology, and Physical Medicine & Rehabilitation, the Huck Institutes of the Life Sciences, and the Center for Neural Engineering, Pennsylvania State University-University Park, PA, USA (e-mail: xxh120@psu.edu).

I. INTRODUCTION

FINGER force control is a critical metric for assessing muscular strength and health. Finger force monitoring has become increasingly popular in the field of Internet of Medical Things (IoMT), which is the systematic integration of the Internet of Things (IoT) technologies and MedTech tools [1]. It has the potential to enhance remote rehabilitation and personalized health monitoring. Normal hand function guarantees manual dexterity for coordinated and skilled movements, enabling precise control for a variety of daily activities. However, individuals with musculoskeletal disorders, such as stroke [2], [3], spinal cord injury [4] or rheumatoid arthritis [5], often experience impaired hand function, including reduced force strength. The impairment or loss of hand function greatly impacts the quality of life, as it affects the ability to perform daily tasks independently. These individuals typically require assistance for normal daily living. Generally, isometric finger force control can indicate the level of independence in activities of daily living (ADL) [6], [7] and is associated with the upper-limb function and performance tests [8], [9]. Finger force monitoring plays a crucial role in the IoMT framework by enabling continuous assessment of the skeletal muscle conditions [10]. This is especially important for individuals undergoing rehabilitation due to neurological or musculoskeletal conditions. Continuous monitoring enables early detection of changes in hand function, which is vital for timely adjustment of intervention strategies [10]. This technology not only supports remote rehabilitation by providing detailed, real-time feedback to both patients and therapists but also enhances personalized health monitoring [11]. Accordingly, physicians and therapists can design, adjust, and optimize rehabilitation programs more effectively, leading to better patient outcomes. Additionally, the ability to monitor finger force remotely reduces the need for frequent hospital visits, making the rehabilitation process more convenient for patients [12].

Most studies employed data gloves integrated with force sensors for finger force monitoring [5], [12], [13], [14], due to their ease of use and implementation. However, several limitations must be considered before use: 1) Data gloves can be cumbersome and interfere with natural hand movements, reducing comfort and affecting the accuracy of hand monitoring; 2) Achieving an optimal fit for a data glove, crucial for ensuring measurement accuracy, is challenging; 3) Data gloves may be unsuitable for specific clinical

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3520990

populations, such as individuals with an arm amputation. Considering these limitations, researchers are exploring alternative approaches for finger force monitoring. Extracting neural-drive signals (i.e., descending commands from the brain to the muscles) from multiple sources, including the brain [15] and muscles [16], is a promising method to predict finger forces. Specifically, many studies [17], [18], [19], [20] focused on developing and optimizing state-based decoding techniques to enhance motion intention performance recognition by distinguishing various brain states related to finger movements. However, the reliance of these studies on invasive techniques and animal models necessitates further validation, posing barriers to widespread use. Although non-invasive brain-machine interface (BMI) techniques [21], [22] using electroencephalogram (EEG) signals have been developed for finger force decoding, they face challenges such as low robustness, mobility limitations, and user discomfort issues. Additionally, scalp EEG generally has a low signal-to-noise ratio, complicating the discrimination of various brain activities [23].

Alternatively, surface electromyogram (sEMG), collected from the skin surface, serves as a promising physiological signal with a relatively high quality and can be easily employed to capture neural-drive signals. sEMG signals are formed by the temporal and spatial summation of hundreds of motor unit action potentials (MUAPs) that originate from motor unit (MU) activations [24]. The number of MUAPs in a certain timeframe correlates with the intensity of the descending neural drive signal to the muscles. This makes sEMG signals suitable for motion intention decoding [25], [26]. A prevailing strategy for finger force predictions involves performing regression analyses to continuously map representative features (e.g. sEMG amplitudes) to finger force [26], [27]. However, finger force prediction using macroscopic features is constrained by crosstalk between adjacent muscles or muscle compartments, motion artifacts, MUAP superimposition, and background noise. These sources of interference can lead to under- or over-estimation errors, thereby degrading the accuracy of finger force predictions.

In contrast, the advent of flexible high-density sEMG (HD-sEMG) electrodes facilitated the development of finger force prediction from a microscopic perspective at the MU level [28]. Generally, the spinal motoneurons receive excitatory neural input from the brain, and the populational MU firing rate is proportional to the neural drive. The populational firing rate, closely associated with finger forces, has been identified as a preferred feature for continuous and concurrent prediction of finger force. To obtain a finger-specific MU set, one prevailing strategy [29] was to decompose the sEMG data during the movement of a target finger. However, non-target fingers may be co-activated biomechanical and neural coupling across fingers, thereby leading to non-target MUs being unintentionally included in the target-finger MU set. This can lead to inaccurate predictions of target finger motor output. To enhance the MU set specificity, several studies [30], [31], [32] required subjects to intentionally avoid co-activations of non-target fingers and applied a MU refinement

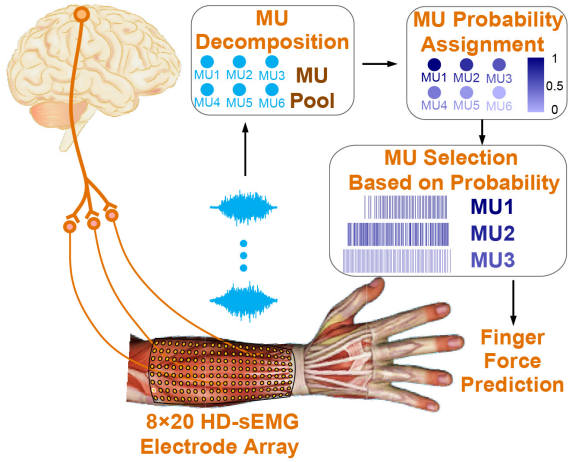

Fig. 1. General framework of the unsupervised neural-drive approach for multi-finger force predictions. An 8×20 HD-sEMG electrode array was attached to the forearm muscle for sEMG data collection. We first decomposed the sEMG data to form a MU pool for the target finger. Given that the MU pool contains plenty of MUs associated with non-target fingers, we assigned each MU a probability to indicate its degree of association with the target finger. Then, we performed the MU selection procedure to retain MUs that exceeded the predefined probability threshold. The retained MUs were used for finger force predictions.

procedure to exclude non-target finger MUs. However, as it was inherently difficult to completely avoid co-activations of non-target fingers, this approach was challenging to implement in practice. Even with careful instruction and training, it still required significant time and effort to successfully obtain such single-finger data. In addition, the MU refinement procedure in those studies retained MUs closely associated with the target-finger force without considering the degree of correlation, potentially retaining interfering MUs. Furthermore, the MU refinement procedure was conducted in a supervised manner, relying on finger force data to calculate MU correlation. This limited the algorithm applicability to individuals with impaired motor functions, such as amputees and stroke survivors. Recently, a promising unsupervised neural decoder was developed via a two-stage MU refinement procedure (1. MU clustering and labeling, and 2. MU weighting). However, single-finger data were still required for the MU extraction. Moreover, the two-stage MU refinement procedure could benefit from further simplification to enhance its efficiency and applicability. An alternative unsupervised neural decoder [34] attempted to share MUs obtained from non-target fingers. However, this sharing procedure significantly increased the number of MUs used for finger force predictions, leading to high computation time, which are infeasible for real-time applications. Additionally, MU initial sources were still extracted from single-finger trials. Consequently, our study was motivated by the need for an accurate and efficient unsupervised neural-drive decoding approach to monitor finger forces in IoMT, with the aim of simplifying the data requirements for decoder training, enhancing the accuracy of force prediction, and ensuring robustness against non-target MUs and variations in background noise.

Accordingly, we introduced a new probability-based strategy, as illustrated in Fig. 1. Specifically, we performed initial MU extraction directly using HD-sEMG data when multiple fingers were active. Unlike previous study [30], the raw MU pool obtained from the multi-finger data included a large number of MUs associated with multiple fingers. Correspondingly, we developed an effective probability-based strategy to refine the MU pool. The probability of a MU being related to the target finger was calculated based on the average firing rate distribution during the desired activation periods of each finger. Then, MUs were retained if the probability was above a predefined threshold. Our developed approach can achieve an accurate prediction performance of multiple finger forces with strong generalization across various probability thresholds, background noise and data sources for initial MU extraction and probability calculation, which has the potential to be implemented in real-time IoMT applications.

The main contributions of the current study include:

1) We developed a novel unsupervised neural-drive decoding approach for multi-finger predictions that achieved consistently high prediction accuracy, suitable for scenarios where finger forces were either unmeasurable or inconvenient to obtain, such as amputees controlling their prostheses.

2) Our decoder training eliminated the need for isolated single-finger tasks by allowing co-activations of non-target fingers. This greatly simplified the data collection process, making the training setup more efficient and less restrictive.

3) The neural decoder was characterized by strong robustness against interference from non-target MUs, task variations, and background noise. This robustness was consistent across a wide range of probability thresholds.

4) The computation time and required computational resources were suitable for real-time applications in IoMT.

II. MATERIALS

A. Subject Information

For this experiment, we enrolled seven participants, including six males and one female, aged between 21 and 35 years, with no reported history of muscular or neurological conditions. They were either faculty members or graduate students at a university. Before participating, subjects signed an informed consent form to acknowledge their comprehensive understanding of and agreement to the research objectives, their rights and potential risks. This study protocol has been approved by the Institutional Review Board of the Pennsylvania State University (Approval Number: STUDY00021035).

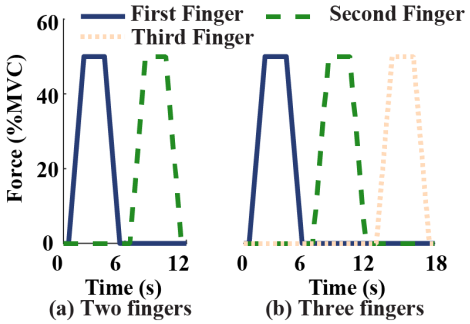

Fig. 2. Multi-finger force trajectories. (a) The two-finger trajectory. The plateau periods are 2 s-4 s and 8 s-10 s for the first finger and the second finger, respectively. (b) The three-finger trajectory. The plateau periods are 2 s-4 s, 8 s-10 s, and 14 s-15 s for the first, second, and third fingers, respectively.

B. Data Acquisition

In the experimental setup, subjects were free to adjust the chair height for optimal comfort. Their forearms were positioned neutrally on the table, and their wrists were kept in a neutral position to minimize contamination of finger force measurements.

1) **Force Measurement**

The experiment involved four fingers (i.e., index, middle, ring, and pinky fingers). Each finger was fastened to a miniature load cell (SM-200N, Interface) for force measurement, which recorded at a sampling frequency of 1000 Hz. At the beginning of the experiment, subjects were instructed to perform their maximal voluntary muscle contraction (MVC) in the extension direction using each finger sequentially, which was used to normalize the forces of the corresponding fingers. Note that the MVC normalization procedure was employed to ensure forces from different fingers of different subjects fell within a unified range, allowing for intuitive comparison and presentation. In practice, we do not need to measure the MVC for the normalization procedure. Previous studies have reported a strong enslaving effect between the ring and pinky fingers [35], [36], [37]. Based on the sEMG recordings, we also

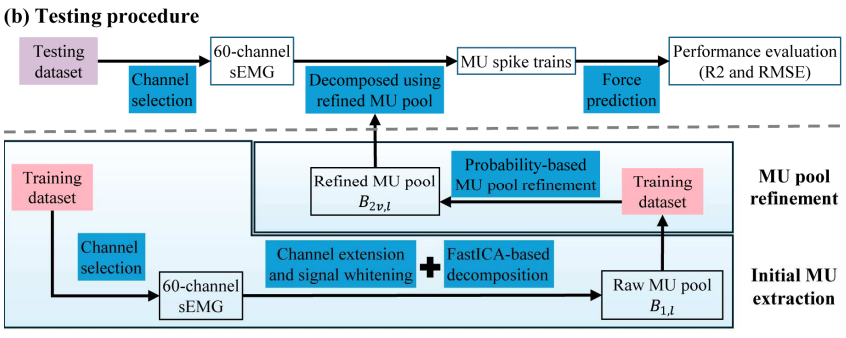

Fig. 3. Flowchart of the unsupervised algorithm for neural decoding. For finger l, the channels and corresponding separation matrix for (b) the testing procedure were obtained from (a) the training and refinement procedures, where $l \in \{index, middle, ring-pinky\}$.

observed similar activation patterns in the extensor digitorum communis (EDC) muscle compartments of the ring finger and the pinky finger [38]. Therefore, we required subjects to extend the ring and pinky fingers simultaneously, with their forces aggregated as the ring-pinky finger [30]. As shown in Fig. 2, the multi-finger forces included two-finger trials and three-finger trials. A total of 24 (3 combinations × 8 trials/combination) trials were performed for the two-finger task with a duration of 12 s. The sequence of the two involved fingers was randomly assigned (Fig. 2(a)). Each finger followed a trapezoid trajectory in succession with a 1 s rest provided between trajectories, similar to that of the three-finger task. A total of 8 trials were performed for the three-finger tasks with a duration of 18 s. For a natural interaction, subjects performed the multi-finger tasks based on their habits, without specific requirements such as avoiding the co-activation of non-target fingers.

2) **sEMG Data Acquisition and Preprocessing**

To reduce skin-electrode impedance, we cleaned the extensor digitorum communis (EDC) muscle with alcohol pads. Then, as shown in Fig. 1, an 8 × 20 high-density electrode array (3 mm diameter, 10 mm inter-electrode distance) was attached to the EDC muscle. The array placement area was determined by palpating the EDC muscle during finger extensions. During the experiment, monopolar sEMG data were collected at a sampling frequency of 2048 Hz using the EMG-USB2+ (OT Bioelettronica, Torino, Italy) with a gain of 1000 and a pass band of 10–900 Hz. Subsequently, we applied an sEMG-specific artifact removal approach [39] to eliminate motion artifacts with a maximal fidelity of sEMG signals preserved.

III. METHODS

A. Unsupervised Neural Decoding Algorithm

As shown in Fig. 3, our developed algorithm consisted of two main parts: 1) the training and refinement procedures, and 2) the testing procedure. For a fair comparison with the previous study, we adopted the same validation strategy (stratified 4-fold validation protocol) to avoid the in-sample optimization bias. Multi-finger trials were divided into the training dataset (three folds for the channel selection, initial MU extraction, and MU refinement) and the testing dataset (one fold for the performance evaluation). Each fold took turns to serve as the testing data, and we obtained results from all four folds, which were then averaged as the results of individual subjects. To implement multi-finger force predictions, we first obtained the raw MU pool (corresponding separation matrix: $\boldsymbol{B}_{1,l}$) using the FastICA-based sEMG decomposition for finger l, where $l \in \{index, middle, ring-little\}$. Then, we conducted the probability-based MU pool refinement procedure to obtain the refined MU pool (corresponding separation matrix: $\boldsymbol{B}_{2,l}$) by calculating the probability of each MU and retaining those exceeding a predefined threshold. Instead of using the FastICA-based decomposition algorithm, we directly applied $B_{2,l}$ to decompose the testing dataset for potential real-time applications. The detailed steps are described below.

1) **Initial MU Extraction**

The initial MU extraction was performed following the flowchart shown in Fig. 4. As suggested by previous studies [24], [40], [41], [42], we conducted the initial MU extraction using the FastICA algorithm [43] because of its high computational efficiency and decomposition accuracy. For finger l, we first calculated the average root mean square (RMS) value of sEMG signals during the plateau period of this finger across the training dataset with finger l activated. Then, we ranked the 160 channels by RMS values in descending order. The 60 channels with the highest RMS values were selected for decomposition [30]. The channel selection procedure can improve computational efficiency without sacrificing the decomposition accuracy [24]. To increase the number of observations, we performed the channel extension by a factor of 10 [30]. The extended signals were then whitened to reduce the inter-observation correlation. Lastly, we applied the FastICA algorithm to decompose the whitened signals. The parameter setting of the FastICA algorithm was based on previous studies [24], [30]; Specifically, the contrast function $G(x) = \frac{1}{3}x^3$ was used for fast convergence. The number of iterations was set to 200 with K-means++ applied for the clustering procedure. Because not all decomposed MUs were of high quality, we removed those with low silhouette (SIL) values, a metric correlated with decomposition accuracy. The SIL threshold was set as 0.5 to ensure decomposition accuracy while reserving sufficient MUs for neural-drive estimation. In addition, among the decomposed MUs, some MUs may be duplicates due to the channel-extension and iterative separation procedures. Therefore, we identified duplicate MUs if over 80% of their synchronized spike trains matched within a ±2.5 ms window for any pair of MUs [30]. The duplicate MU with a smaller SIL value was removed. After the low-quality

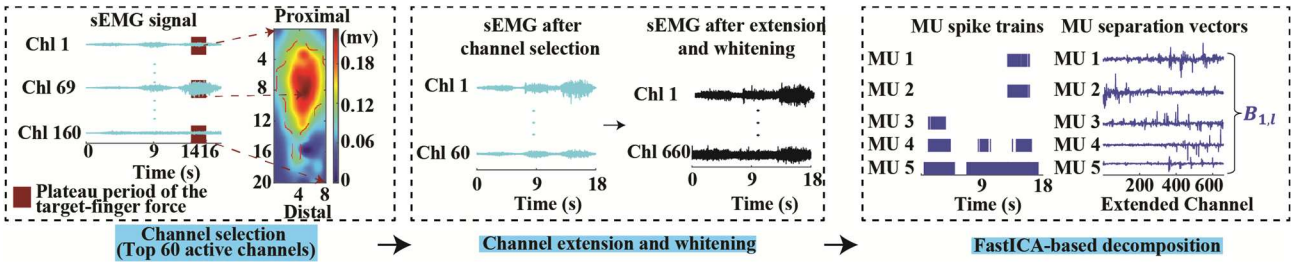

Fig. 4. Initial MU extraction. $l \in \{index, middle, ring-pinky\}$.

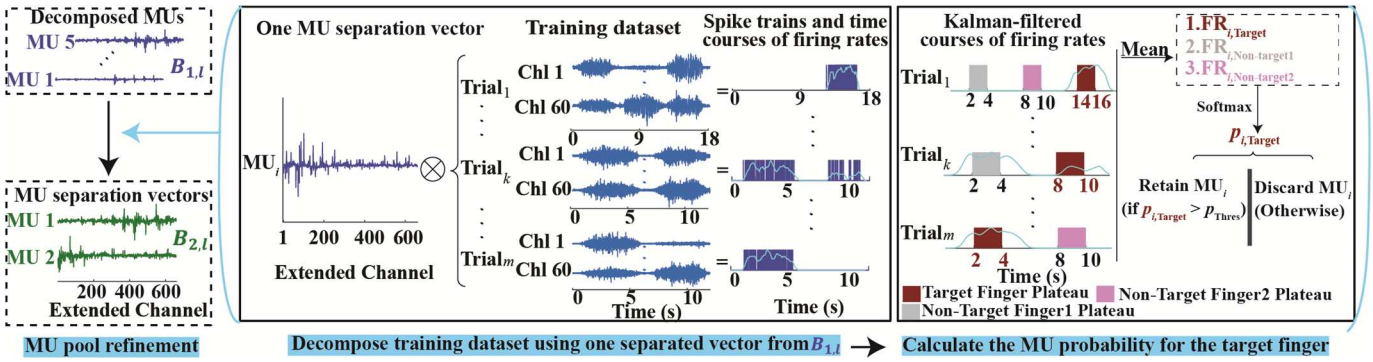

Fig. 5. MU pool refinement. $l \in \{index, middle, ring-pinky\}$, and p_{Thres} denotes the probability threshold.

and duplicate MU removal procedure, the decomposed MUs were pooled together to form the raw MU pool. The corresponding separation vectors were concatenated as a separation matrix (termed $\boldsymbol{B}_{1,l}$). Detailed algorithmic steps are presented in Section A of the supplementary material.

2) MU Pool Refinement

As at least two fingers were activated in each multi-finger trial and co-activations were allowed, the inclusion of MUs associated with non-target fingers was inevitable. Therefore, we conducted an MU pool refinement procedure to identify and exclude these MUs based on their probabilities of association with the target finger. As shown in Fig. 5, the following steps were conducted:

Step 1: Apply the separation matrix ($\boldsymbol{B}_{1,l}$) to decompose the training dataset. Although the FastICA algorithm is efficient for sEMG decomposition, the computation time remained too high to satisfy real-time implementations. Therefore, we directly employed the separation matrix $\boldsymbol{B}_{1,l}$ to decompose sEMG signals and obtain the spike train $\boldsymbol{t}_{i,k,1l}$ ($i = 1,2,\dots,n_{1,l}$; $k = 1,2,\dots,m$), where $n_{1,l}$ denotes the number of separation vectors in $\boldsymbol{B}_{1,l}$, m denotes the number of trials used in the training dataset.

Step 2: Convert $\boldsymbol{t}_{i,k,1l}$ to the time course of firing rates ($\boldsymbol{f}_{i,k,1l}$). The spike train $t_{i,k,1l}$ was segmented using a 0.5-s sliding window with a 0.1-s sliding step. Then, the firing rate within each sliding window was calculated and concatenated over time as $\boldsymbol{f}_{i,k,1l}$.

Step 3: Smooth the time course of firing rates. The obtained $\boldsymbol{f}_{i,k,1l}$ was then smoothed using a Kalman filter to address sporadic, large-amplitude, and isolated fluctuations in the time series [44]. The observation covariance, system covariance, observation matrix, and system matrix were set to 0.5, 0.1, 1, and 1, respectively. The same parameter settings were applied in subsequent analyses.

Step 4: Calculate the mean firing rate across the plateau periods. For the i-th separation vector in $\boldsymbol{B}_{1,l}$, we obtained m time courses of firing rates. Then, we calculated $FR_{i,Target}$ by averaging the mean firing rates during the plateau target activation periods of the target finger across m time courses of firing rates. Note that the plateau only reflected the desired or target force level not the actual force. Similarly, $FR_{i,Non-target1}$ and $FR_{i,Non-target2}$ were calculated for the i-th MU during the other two non-target finger plateau periods, respectively.

Step 5: Compute the probability of each MU specific to finger l. We used the softmax function to compute the probability ($p_{i,Target}$) as

$$p_{i,Target} = \frac{e^{FR_{i,Target}}}{e^{FR_{i,Target}} + e^{FR_{i,Non-Target1}} + e^{FR_{i,Non-target2}}} \quad (1)$$

Step 6: Refine the raw MU pool. A probability threshold (p_{thresh}) was set to retain MUs with $p_{i,Target} > p_{thresh}$. We performed a sensitivity analysis on the threshold as described later in Section III-B. The retained MUs were pooled together to form the refined MU pool, and the corresponding separation vectors were concatenated to form the separation matrix ($\boldsymbol{B}_{2,l}$) of the refined MU pool.

3) Force Prediction

Considering that neural-drive signals typically exhibit a linear correlation with finger force [30], we built three linear regression models to individually predict the forces of the three fingers. Although measured forces were employed for mapping the neural-drive signals to the forces, the measured forces were not involved in any critical steps of the training and refinement procedures. In practice, users can customize the force prediction models to their subjective perception of force levels without the measured forces. In addition, to offer

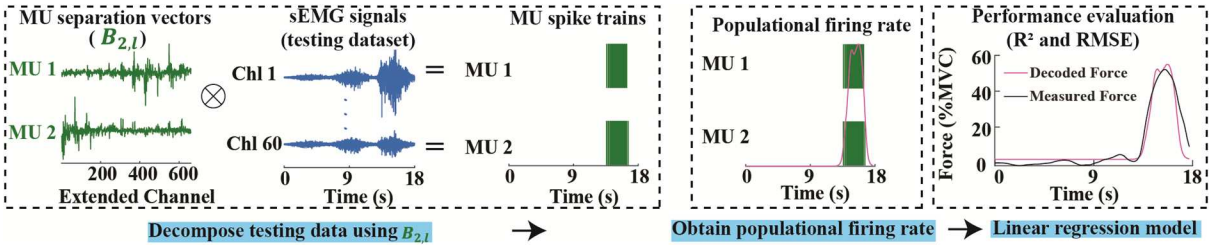

Fig. 6. Force prediction for finger l. $l \in \{index, middle, ring-pinky\}$. ⊗: apply the separation matrix to sEMG signals after channel extension and signal whitening procedures.

a fair quantitative comparison with other approaches and an intuitive presentation of the prediction performance, we also used the measured forces for the performance evaluation of force predictions, which is not needed for practical use. Specifically, we conducted the following steps for the force prediction of finger l (as shown in Fig. 6):

Step 1: Decompose the testing dataset. Similar to the refinement procedure, we directly employed the refined separation matrix $\boldsymbol{B}_{2,l}$ to obtain source signals for a time-efficient decomposition.

Step 2: Convert the decomposed source signals to spike trains.

Step 3: Obtain the time courses of firing rates of individual MUs $\boldsymbol{F}_{2,l} = [\boldsymbol{f}_{1,2l}, \boldsymbol{f}_{2,2l}, \cdots, \boldsymbol{f}_{i,2l}, \cdots, \boldsymbol{f}_{n_2,2l}]$, where n_2 denotes the number of retained MUs.

Step 4: Calculate the neural-drive signals (populational firing rate) as

$$\boldsymbol{D}_{2,l} = \sum_i \boldsymbol{f}_{i,2l} \tag{2}$$

Step 5: Smooth $\boldsymbol{D}_{2,l}$ using the Kalman filter.

Step 6: Use the same sliding window strategy to process the measured force of finger l.

Step 7: Predict the force of finger l using a linear regression model.

$$Force_l = a_l \boldsymbol{D}_l + b_l \tag{3}$$

where $Force_l$ represents the predicted force for finger l, a_l and b_l denote the corresponding slope coefficient and intercept, respectively.

We used four complementary metrics for force prediction performance evaluation: the coefficient of determination (R^2), the root mean square error (RMSE), the Pearson correlation coefficient (PCC), and mean absolute error (MAE), which can be defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

$$PCC = \frac{\sum_{i=1}^{n}(y_i - \overline{y})(\hat{y}_i - \overline{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})^2}} \tag{6}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

where y_i and $\hat{y}_i$ represent the measured value and the predicted value of the i th observation, respectively. $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, $\overline{\hat{y}} = \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i$ represent the average measured value and predicted value, respectively. n is the number of observations.

TABLE I
PROTOCOLS FOR THE INCLUSION OF NON-TARGET TRIALS. THE SYMBOLS "√" AND "×" REPRESENT THE INCLUSION AND EXCLUSION OF NON-TARGET TRIALS IN THE CORRESPONDING PROTOCOLS, RESPECTIVELY.

	Initial MU extraction with non-target trials	MU probability assignment with non-target trials
Protocol 1	√	√
Protocol 2	√	×
Protocol 3	×	√
Protocol 4	×	×

B. Decoder Robustness Evaluation

1) **Sensitivity Analysis of Probability Thresholds**

In our study, to enhance the accuracy of force predictions, each MU was evaluated to determine its probability of association with the target finger l, and a probability threshold was established to retain MUs with probabilities above the threshold. To comprehensively explore the effects of different probability thresholds on force predictions, we experimented with a wide range of threshold values, ranging from 0 to 0.9 in increments of 0.1 (10 thresholds), and included finer granularity thresholds such as 0.95, 0.99, 0.995, and 0.999 as the threshold approached 1. This systematic investigation can provide valuable insights into the robustness of force prediction under different probability thresholds. It also enabled us to identify an optimal threshold that minimized prediction errors while maintaining a sufficient number of MUs, ensuring the generalizability and practical utility of the developed decoder.

2) **Inclusion of Non-Target Finger Trials**

In our study, we defined the term 'robustness' as the ability to ensure accurate and reliable force predictions despite changes in probability thresholds, task variations and background noise. Specifically, in the training dataset, the two-finger trials that excluded the target finger l were termed non-target trials. The force prediction performance may be affected by the trial source for two key procedures (i.e., the initial MU extraction and the MU probability assignment).

- In the initial MU extraction stage, we pooled the MUs extracted from the training dataset together to form the raw MU pool. However, most MUs decomposed from non-target trials were not closely associated with the finger l. Therefore, we explored the decoder robustness to non-target MUs by comparing force prediction performance with and without including the non-target trials for the initial MU extraction.
- For the MU probability assignment, the probability calculation was based on the average firing rates observed during the force plateau periods of three fingers. Correspondingly, we investigated the decoder robustness by including and excluding the non-target trials.

We assessed decoder robustness against non-target MUs and trials by evaluating the effects of including or excluding them during the initial MU extraction and probability assignment procedures. This involved comparing force prediction performances across four different protocols, as detailed in Table I.

3) **Robustness Under Background Noise Conditions**

In practice, the neural decoder is intended for use in common daily scenarios, where the environment often contains uncontrollable interference, with background noise being one of the most common sources. Therefore, we tested our neural decoder under different noise conditions to evaluate

its robustness. Since the acquisition of training data was performed under a minimum level of background noise to avoid low-quality signals, we intentionally introduced Gaussian noise to the testing dataset to simulate the unpredictable and challenging conditions of real-world environments. A total of four noise levels, defined by signal-to-noise ratios (SNRs) of 5 dB, 10 dB, 15 dB, and 20 dB, were examined. For each SNR level, Gaussian noise was added to each channel individually, ensuring that the sEMG signal in each channel did not exceed the target SNR. The noisy sEMG data were then fed into the trained neural decoders for evaluation.

C. Comparison with sEMG-Amplitude-Based Force Predictions

As indicated by the early work [45], the sEMG amplitude has been preferred over various time-domain and frequency-domain features. In this study, we conducted two different sEMG-amplitude-based approaches for comparison. We applied the same validation protocol and parameter settings (if any) as the developed unsupervised neural drive approach if not specified.

For the first approach, the number of sEMG channels for the muscle strength calculation was fixed at 60 (termed the sEMG60 method), after removing motion artifacts and powerline noise. The rationale for selecting 60 channels rather than using all 160 channels is that most sEMG activities can be captured by 60 channels [30]. Additionally, channels located outside or on the edges of the activation area, which often have low RMS values, contributed little to finger force predictions. Specifically, the following steps were conducted for the force prediction of finger $l \in \{index, middle, ring-pinky\}$.

Step 1: Calculate the average RMS value of sEMG signals during the plateau period of the finger l across the training dataset.

Step 2: Rank all channels from high to low based on the calculated RMS values and select the top 60 channels ($C_{60,l}$) with the highest RMS values.

Step 3: For a testing trial, calculate sEMG amplitude (RMS) for each channel using the same sliding window strategy (0.5-s sliding window with a 0.1-s sliding step) as the neural-drive approach, and average across the 60 channels, which generated an overall time course of the sEMG amplitude for finger l ($\boldsymbol{A}_{60,l}$).

Step 4: Smooth the $\boldsymbol{A}_{60,l}$ using the same Kalman filter as in the neural-drive approach.

Step 5: Use the same sliding window strategy to process the corresponding measured force of finger l, resulting in $\boldsymbol{F}_{\boldsymbol{l}}$.

Step 6: Build a linear regression model between the filtered RMS of sEMG $\boldsymbol{A}_{60,l}$ and the segmented measured force $\boldsymbol{F}_{\boldsymbol{l}}$.

Following the basic procedure of the first approach (sEMG60 method), the second approach further refined the 60-channel pool, termed the sEMG-Ref method. We conducted the refinement to reduce crosstalk interference that could bias force predictions due to substantial channel overlap between fingers [38]. Furthermore, the number of active channels for different fingers may vary due to the different spatial distributions of muscle compartments. The channel refinement strategy was conducted after *Step 2* of the sEMG60 method as:

Step 1: For the target finger l_1 (l_1=Index, Middle, or Ring-Pinky), calculate and smooth the time course of the sEMG amplitude, resulting in $\boldsymbol{a}_{i,l1}$, where $i = 1,2,\dots,60$.

Step 2: Process measured forces of three fingers using the same sliding window procedure, obtaining $\boldsymbol{F}_{\text{Index}}$, $\boldsymbol{F}_{\text{Middle}}$ and $\boldsymbol{F}_{\text{Ring-Pinky}}$.

Step 3: Conduct a linear regression between $\boldsymbol{a}_{i,l1}$ and processed finger forces, deriving the coefficient of determination R^2_{i,l_1-l_2} (l_1, l_2 =Index, Middle or Ring-Pinky) between the i-th channel in $\boldsymbol{A}_{60,l1}$ and the measured force of finger l_2.

Step 4: Exclude the i-th channel from $C_{60,l1}$ if $R^2_{i,l_1-l_1} < R^2_{i,l_1-l_2}$ and $l_1 \neq l_2$. The retained channels were used for force prediction following procedures outlined in the sEMG60 method.

D. Statistical Analysis

For the confidence interval (CI) analysis, we employed a bootstrap technique to randomly select 1000 sample sets from the obtained results for calculating the 95% CI of correlation and prediction error values. For the significance analysis, we adopted the Repeated Measures Analysis of Variance (ANOVA) and *t*-test for pairwise comparisons if the compared groups satisfied the normal distribution ($p > 0.05$ in the Shapiro-Wilk test) and the sphericity assumption ($p > 0.05$ in the Mauchly's test) [46]. Otherwise, we employed non-parametric analysis approaches (Friedman test and Wilcoxon signed-rank test) [47]. To minimize errors from multiple comparisons, we applied the Holm-Bonferroni correction where applicable. In this study, we set the significance level at 0.05 and reported only adjusted p-values.

IV. Results

A. Decoder Robustness Evaluation

To evaluate the decoder robustness against various probability thresholds, we analyzed force prediction performance across a range of thresholds. Similarly, to assess the decoder robustness against the non-target MUs and trials, we evaluated force prediction performance variations across four protocols, distinguished by whether non-target trials were included or excluded during the initial MU extraction and MU probability assignment procedures. Fig. 7 shows the trends of R^2, RMSE, PCC, and MAE, respectively, as the probability threshold increased from 0 towards 1. Accordingly, we summarized the ranges of the four metric values, along with their corresponding probability thresholds, in Table II for all protocols.

In Fig. 7(a) and Fig. 7(c), the mean R^2 and PCC increased rapidly across all four protocols to reach a plateau from p_{Thres} = 0 to 0.1. This plateau was sustained across the probability threshold range up to 0.9 for all protocols, indicating strong

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3520990

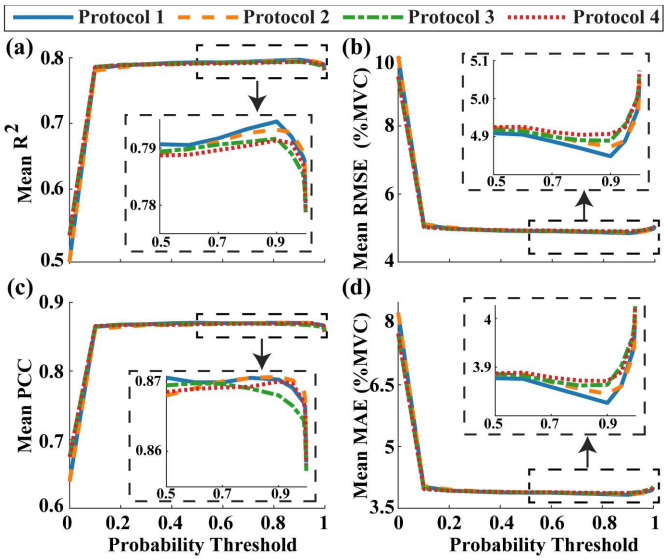

Fig. 7. Multi-finger force predictions using different probability thresholds. Performance metrics R^2, RMSE, PCC, and MAE are shown sequentially in (a)-(d). The dashed box in the inset of each subfigure indicates enlarged areas of the probability threshold ranging from 0.5 towards 1 for a clear demonstration of minor variations.

decoder robustness to probability thresholds for the R^2 and PCC metrics. Then, the mean R^2 and mean PCC decreased slightly after p_{Thres} = 0.9 for all protocols.

In Fig. 7(b) and Fig. 7(d), the mean RMSE and MAE for all four protocols decreased sharply from p_{Thres} = 0 to 0.1. Then, the mean RMSE and mean MAE tended to stabilize across the remaining probability threshold range up to 0.9 for all protocols, indicating strong decoder consistency for the RMSE and MAE metrics. Following that, the mean RMSE and mean MAE exhibited a slight increase for all protocols.

Overall, for all performance metrics, the force prediction performance using all four protocols demonstrated strong robustness across a wide range of probability thresholds. In addition, the results revealed that the force prediction performances using all four protocols were similar after $p_{Thres} \geq 0.1$, demonstrating the decoder effectiveness against non-target MUs and trials used for the MU probability calculation.

B. Computational Efficiency Evaluation

The term "lightweight" in this study was used to describe our computational processes requiring minimal computation time and resources. Considering the critical importance of computational efficiency in IoMT applications, we evaluated the computation time of our developed approach. The computational efficiency analyses were performed on an AMD Ryzen 7 6800H @ 3.2 GHz, using MATLAB R2023a (The MathWorks Inc., USA) as the implementation software.

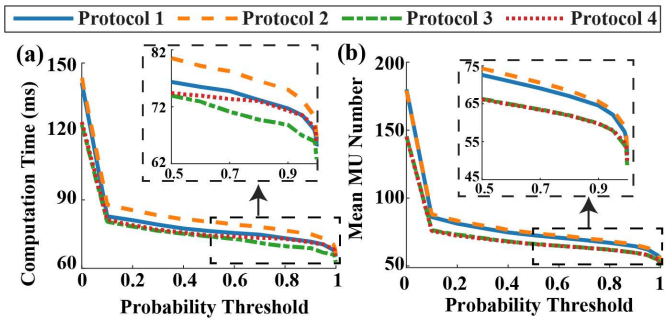

Fig. 8. Evaluation of computational efficiency. (a) Variation in computation time. (b) Variation in MU numbers. The dashed boxes in the insets of (a) and (b) indicate enlarged areas on the probability threshold ranging from 0.5 towards 1 for a clear demonstration of minor variations.

Real-time performance was evaluated via pseudo real-time testing, where data segments (simulating data buffer in the real-time data acquisition system) were fed to the algorithm sequentially. As illustrated in Fig. 8(a), computation time decreased sharply across all four protocols as the probability threshold increased from 0 to 0.1, then continued to decline at a reduced rate beyond this point. The longest computation time for $p_{Thres} \geq 0.1$ across all four protocols was 87.8±12.4 ms, which met the real-time requirements (an ideal loop delay ranges between 100 ms and 150 ms [48], [49]), demonstrating high computational efficiency across all four protocols and a wide range of probability thresholds ($p_{Thres} \geq 0.1$).

Generally, the number of MUs used for the decomposition has a large impact on the sEMG decomposition duration and the required computational resources. Therefore, we investigated the MU number variation as the probability threshold increased. Fig. 8(b) shows that the number of MUs significantly decreased as the probability threshold increased from 0 to 0.1, then continued to decrease at a slower rate, indicating a stable computational load in terms of MU numbers for $p_{Thres} \geq 0.1$ across all protocols.

In summary, a clear pattern of improved computational efficiency was present as the probability threshold increased, especially from 0 to 0.1. The acceptable computation time and MU numbers for p_{Thres} underscored the practical applicability of our developed decoder across all four protocols, justifying the term "lightweight".

C. Effects of the Probability Thresholding Strategy

Fig. 9 and Table III present force prediction performance without the probability thresholding strategy (i.e., using the raw MU pool) and using this strategy with critical probability

TABLE II
MEAN VALUE RANGES OF FOUR METRICS (R^2, RMSE, PCC, and MAE). THE VALUES IN PARENTHESES REPRESENT THE PROBABILITY THRESHOLDS USED TO ACHIEVE METRIC VALUES LISTED OUTSIDE THE PARENTHESES.

	R^2	RMSE (%MVC)	PCC	MAE (%MVC)
Protocol 1	0.4872 (0) – 0.7947 (0.9)	4.8484 (0.9) – 10.0471 (0)	0.6400 (0) – 0.8703 (0.5)	3.8265 (0.9) – 8.2514 (0)
Protocol 2	0.4866 (0) - 0.7932 (0.9)	4.8732 (0.9) – 10.0497 (0)	0.6392 (0) – 0.8704 (0.9)	3.8451 (0.9) – 8.2496 (0)
Protocol 3	0.5273 (0) - 0.7917 (0.9)	4.8893 (0.9) – 9.4415 (0)	0.6751 (0) – 0.8695 (0.6)	3.8623 (0.8) – 7.7349 (0)
Protocol 4	0.5273 (0) - 0.7914 (0.9)	4.9046 (0.8) – 9.4497 (0)	0.6748 (0) – 0.8698 (0.9)	3.8715 (0.9) – 7.7408 (0)

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3520990

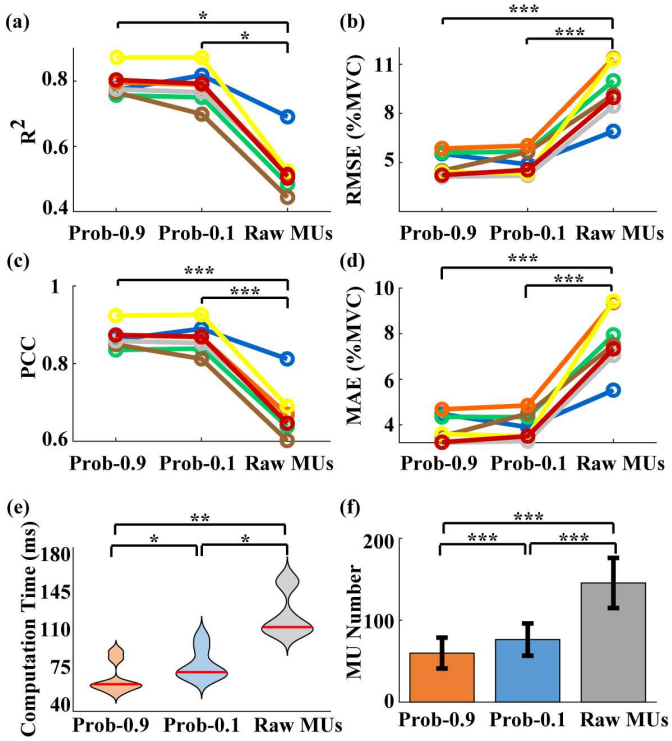

Fig. 9. Force prediction performance evaluation using the probability thresholding strategy. Specifically, performance without the probability threshold and with the probability threshold using two critical thresholds of 0.1 and 0.9 are evaluated using (a) R^2 metric, (b) RMSE metric, (c) PCC metric, (d) MAE metric, (e) computation time, and (f) MU number analyses. In (a) to (d), results in the same color denote the same subject. In (e), red lines indicate median values. In (f), the length of the error bars represents the standard deviation. * denotes $0.01<p<0.05$, ** denotes $0.001<p<0.01$, *** denotes $p<0.001$.

thresholds, termed Prob-0.1 (probability threshold of 0.1, which is the inflection point) and Prob-0.9 (probability threshold of 0.9, which represents the point of best performance in most cases). Although similar performance was achieved using all four protocols, we selected protocol 3 here due to its relatively low computation time and a small number of MUs.

As shown in Fig. 9(a) to Fig. 9(d), for all performance metrics, force prediction performance using both the Prob-0.9 and Prob-0.1 approaches were significantly better than without the probability threshold (all $p < 0.05$). Similar conclusions were observed for the comparisons of computation time and MU numbers (as shown in Fig. 9(e) and Fig. 9(f)), demonstrating the effectiveness of the developed strategy.

For the force prediction performance comparison using critical probability thresholds, no significant difference was detected in the four metrics between the Prob-0.9 and Prob-0.1 approaches (all $p < 0.05$). In terms of computation time and MU numbers, the Prob-0.9 approach outperformed the Prob-0.1 approach (computation time: $p = 0.023$, MU numbers: $p < 0.001$), indicating that satisfactory results can be achieved with less computation time and resources.

D. Comparison with Prevailing Approaches

To further demonstrate the superiority of the developed approach, we conducted force prediction performance evaluations using the developed neural-drive, sEMG-Ref, and sEMG60 approaches, as shown in Fig. 10 and Table IV. The parameter configuration for the neural-drive approach was a probability threshold of 0.9 under protocol 3 (same setting as the Prob-0.9 in Section 3.3) due to its ability of accurate prediction performance and high computational efficiency.

Fig. 10(a) and Fig. 10(b) present two representative force prediction examples using the three approaches to visually demonstrate their predictive capabilities. The neural-drive approach can accurately predict the force of each finger in scenarios where only one finger was activated at a time, as shown in Fig. 10(a). In the case of co-activation of multiple fingers, Fig. 10(b) demonstrates that the neural-drive approach can also accurately predict the force exerted by each finger in scenarios where fingers were activated simultaneously. In contrast, the force predictions using the sEMG-Ref and sEMG60 approaches were greatly affected by the activation of non-target fingers, leading in a large deviation from the measured forces.

As shown in Fig. 10(c) to Fig. 10(f), the statistical analysis revealed significant differences for R^2 (Friedman test: $\chi^2(2) = 12.29$, $p = 0.0021$, η²_partial=0.7347), RMSE (ANOVA: $F(2,12)=20.7$, $p < 0.001$, η²_partial=0.7753), PCC (Friedman test: $\chi^2(2) = 12.29$, $p = 0.0021$, η²_partial=0.7347), and MAE (ANOVA: $F(2,12)=22.8$, $p <$

TABLE III
EVALUATION OF FORCE PREDICTION PERFORMANCE WITHOUT PROBABILITY THRESHOLD AND WITH PROBABILITY THRESHOLDS OF 0.1 AND 0.9 (RESULTS PRESENTED AS MEAN ± STANDARD ERROR, VALUES IN PARENTHESES REPRESENT 95% CI)

	Prob-0.9	Prob-0.1	Raw MUs
R^2	0.79±0.039 (0.77-0.82)	0.78±0.055 (0.75-0.82)	0.53±0.078 (0.48-0.59)
RMSE (%MVC)	4.89±0.73 (4.43-5.43)	5.04±0.73 (4.53-5.54)	9.44±1.59 (8.32-10.50)
PCC	0.87±0.028 (0.85-0.89)	0.87±0.037 (0.84-0.89)	0.68±0.067 (0.64-0.72)
MAE (%MVC)	3.86±0.62 (3.47-4.32)	3.98±0.59 (3.58-4.40)	7.73±1.36 (6.79-8.64)
Computation Time (ms)	68.83±13.63 (60.31-78.91)	80.35±13.43 (72.18-90.54)	122.92±19.70 (110.68-137.59)
Number of MUs	59.84±18.90 (47.96-73.81)	76.36±19.69 (64.16-91.23)	145.46±30.57 (126.17-167.35)

TABLE IV
EVALUATION OF FORCE PREDICTION PERFORMANCE USING THE DEVELOPED NEURAL-DRIVE METHOD, THE SEMG-REF METHOD, AND THE SEMG60 METHOD (RESULTS PRESENTED AS MEAN ± STANDARD DEVIATION, VALUES IN PARENTHESES REPRESENT 95% CI)

	Neural drive	sEMG-Ref	sEMG60
R^2	0.79±0.039 (0.77-0.82)	0.64±0.080 (0.59-0.69)	0.50±0.12 (0.44-0.59)
RMSE (%MVC)	4.89±0.73 (4.43-5.43)	7.31±1.88 (6.05-8.57)	9.41±2.47 (7.75-11.05)
PCC	0.87±0.028 (0.85-0.89)	0.76±0.060 (0.72-0.80)	0.65±0.095 (0.60-0.72)
MAE (%MVC)	3.86±0.62 (3.47-4.32)	6.08±1.51 (5.07-7.09)	7.89±2.09 (6.47-9.28)

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3520990

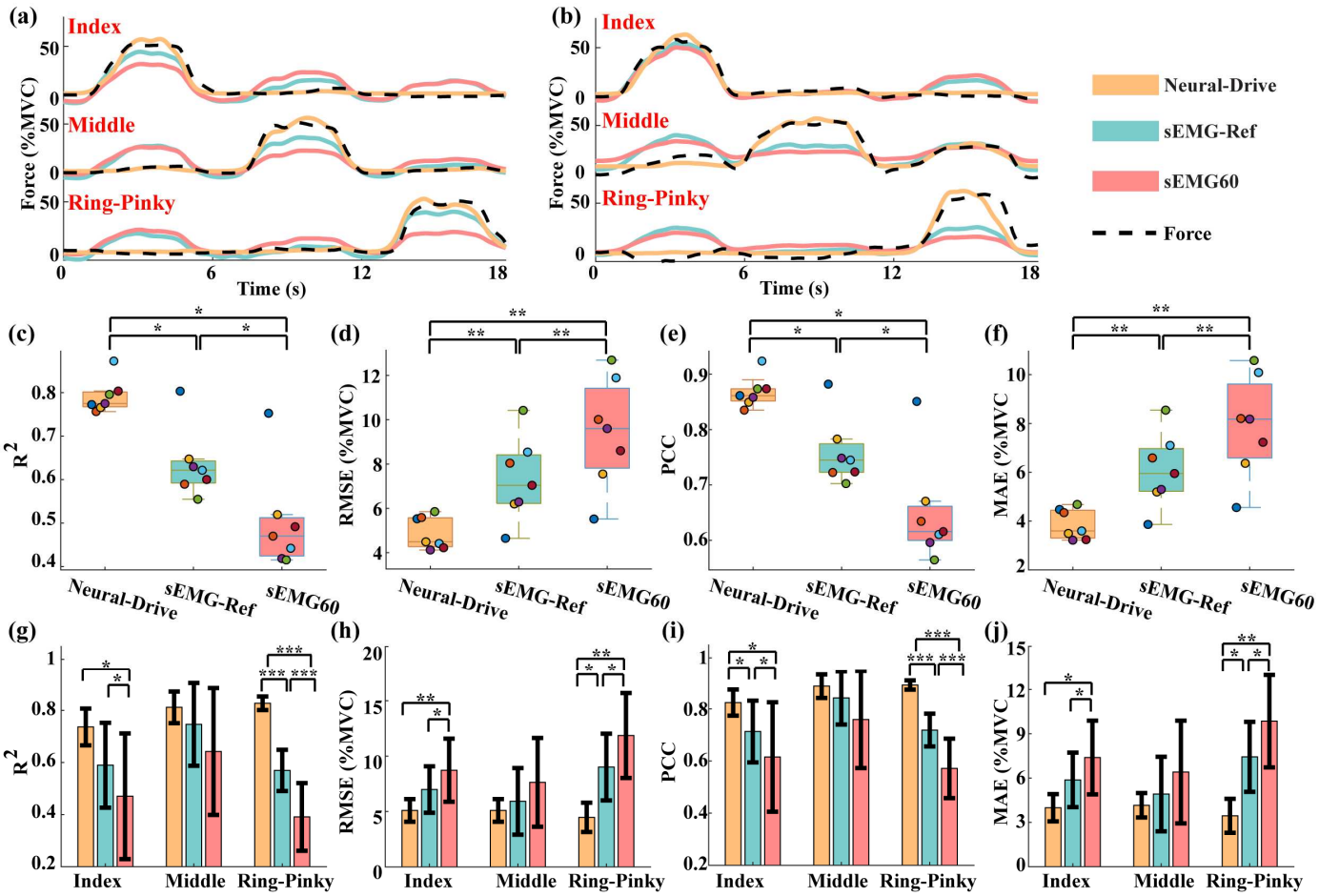

Fig. 10. Comparison of the developed neural-drive method, the sEMG-Ref method, and the sEMG60 method. (a) A representative demonstration of the force prediction where only one finger was activated at a time. (b) A representative demonstration of the force prediction where more than one finger was activated simultaneously. (c) to (f) show R^2, RMSE, PCC, and MAE values for force predictions, respectively. In (c) to (f), dots of the same color represent results from the same subject. (g) to (j) illustrate R^2, RMSE, PCC, and MAE values for the force prediction performance of each finger, respectively. In (g) and (j), the error bars represent the standard deviation. * denotes $0.01<p<0.05$, ** denotes $0.001<p<0.01$, *** denotes $p<0.001$.

0.001 , η²_partial=0.7917), indicating exceptionally large effect sizes (as suggested by [50]). Further post-hoc analyses demonstrated that the R^2 and PCC values obtained using the neural-drive method was significantly higher than those using the sEMG-Ref and sEMG60 methods (all $p < 0.05$). Similarly, the RMSE and MAE values obtained using the neural-drive method was significantly lower than those using the sEMG-Ref and sEMG60 methods (all $p < 0.01$).

As shown in Fig. 10(g) to Fig. 10(j), force prediction performance was also compared for individual fingers. For the R^2 metric in Fig. 10(e), significant differences were found for the index finger ($\chi^2(2) = 8.86$, $p = 0.012$) and the ring-pinky finger ($F(2,12)=20.7$, $p < 0.001$). Further post-hoc analyses revealed that R^2 values obtained using the neural-drive method was significantly higher than those using the sEMG60 method for both the index finger and the ring-pinky finger (both $p < 0.05$). In addition, the R^2 values obtained using the neural-drive method was significantly higher than that using the sEMG-Ref method for the ring-pinky finger ($p < 0.001$). Similar results were observed for the RMSE (Fig. 10(h)), PCC (Fig. 10(i)), and MAE (Fig. 10(j)).

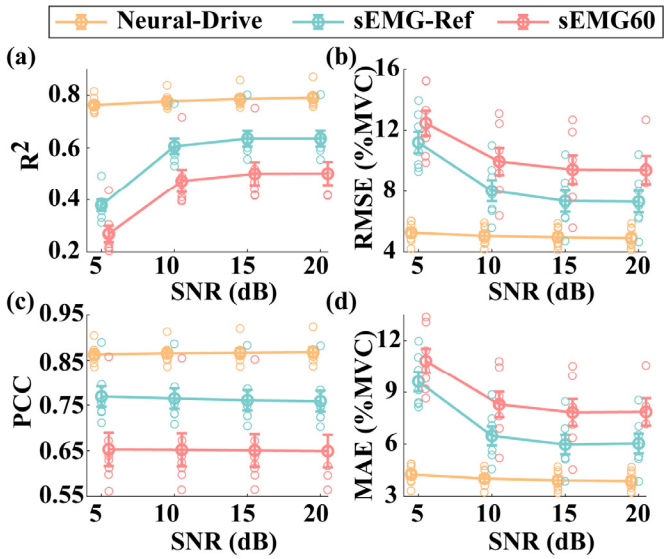

Fig. 11. Performance evaluation of finger force prediction under varying noise levels. Each dot represents the average result for each subject. Lines connect the average results of each method across varying noise levels.

As shown in Fig. 11, the prediction performance of our neural decoder remained stable across varying noise levels, demonstrating its strong robustness to noise interference. In contrast, both the sEMG-Ref and sEMG60 approaches exhibited a decline in prediction performance, as indicated by the R^2, RMSE, and MAE metrics, especially when the SNR decreased from 10 dB to 5 dB. For the PCC metric, the prediction performance of both sEMG-amplitude-based approaches remained relatively stable across varying SNR levels. This was mainly because the PCC metric measured relative trends between the predicted and measured forces, rather than their absolute accuracy. For each evaluation metric, our neural decoder significantly outperformed both sEMG-Ref and sEMG60 approaches across all examined SNR levels (all $p < 0.01$).

V. DISCUSSION

In this study, we aimed to develop an unsupervised, robust, and lightweight neural-drive decoder capable of efficiently predicting dexterous multi-finger forces. Building on a previous study [30], we took a critical step forward by customizing the decoder for potential IoMT-based finger force monitoring applications. Specifically, we derived the raw MU pool directly from multi-finger data, without intentionally avoiding non-target MUs. Our developed decoder effectively quantified the degree of association (probability) between each MU and the target finger using its firing rate distribution. Subsequently, MUs with probabilities exceeding the predefined threshold were retained for dexterous multi-finger force predictions. The results demonstrated the feasibility of our developed decoder for reliable finger force monitoring.

A. Effects of Probability Thresholds

Compared with force predictions using raw MUs, force prediction performance significantly improved with the application of a probability threshold, primarily due to the effective removal of interfering MUs. Although raw MUs were derived using sEMG data collected from specific channels of the target-finger, the inclusion of numerous non-target MUs was unavoidable due to the following reasons: 1) We acquired the raw MU pool using multi-finger data, which included the activation of at least two fingers. 2) Different fingers shared several common active electrode channels, because the muscle compartments of the fingers are spatially close. In general, these interfering MUs, being closely related to other fingers, contribute little to the target-finger force prediction. To illustrate this point more clearly, we calculated R^2 values to quantify the correlation between the firing rate of each MU and the corresponding target-finger force, then averaged the R^2 values of all MUs retained for the force prediction. As shown in Fig. 12, the average R^2 value was the lowest when $p_{Thres} = 0$ under any protocol, revealing a large presence of interfering MUs. As p_{Thres} increased from 0 to 0.1, the average R^2 value increased significantly, indicating the effective removal of interfering MUs.

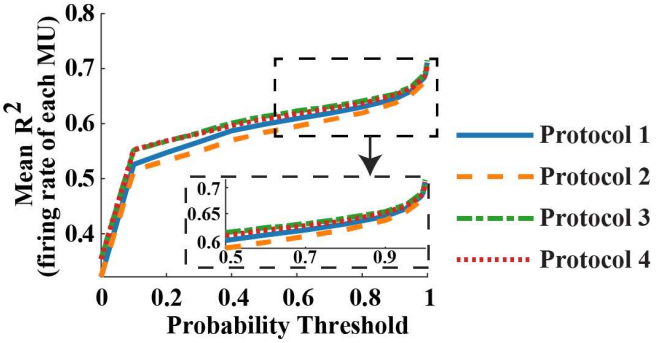

Fig. 12. Average R^2 values between the firing rate of each MU and the corresponding target-finger force.

B. Robust Prediction Across Probability Thresholds

Finger force prediction performance (across all four metrics) tended to be stable for $p_{Thres} \geq 0.1$, indicating a generalized performance across a wide range of thresholds. This can be explained as follows. Some MUs contributed to multi-finger forces, resulting in the co-activation effect. Relatively low MU probabilities indicated high co-activation levels with the target finger. However, the quantity of those MUs was far smaller than that of MUs specific to the target finger, as shown in Fig. 12. For example, the average MU numbers only decreased from 76 to 60 when p_{Thres} increased from 0.1 to 0.9 under protocol 3. Thus, the effect of these MUs was limited to some extent, resulting in stable performance for $p_{Thres} \geq 0.1$.

Fig. 7 shows high robustness to the interferences of non-target MUs and trials. For $p_{Thres} = 0$ (i.e., the raw MU pool without refinement), better prediction performance can be achieved when only trials with the active target finger were used for the initial MU extraction (protocols 3 and 4), compared with when all trials were used (protocols 1 and 2). This improvement was due to a reduced number of non-target-specific MUs, which contributed little to prediction performance. These MUs can be largely removed after applying the probability threshold (i.e., $p_{Thres} > 0$), thereby effectively eliminating the performance gaps between protocols. In addition, the MU probability was calculated based on the average firing rates during the plateau activation of each finger. The average firing rates during plateau periods were not affected substantially by the inclusion of non-target trials, thus the calculated prediction probability did not change much.

C. Comparison with Prevailing sEMG-Amplitude-Based Approaches

The overall force prediction performance of our neural-drive approach was significantly better than that of sEMG-amplitude-based approaches, which can be attributed to inherent limitations of the sEMG-amplitude-based method. As mentioned above, the muscle compartments of the fingers are spatially close, and they partially overlap when observed from the skin surface. Crosstalk from non-target compartments can corrupt some sEMG signals, leading to inaccurate force predictions. Although channel refinement in the sEMG-Ref approach can reduce the crosstalk effects, the remaining channels may still be affected by this interference. In contrast, the neural-drive approach evaluated MU probabilities and removed MUs closely associated with non-target fingers to

TABLE V
COMPARISON OF OUR NEURAL DECODER WITH PREVIOUS STUDIES.

	Zheng et al. [30]	Rubin et al. [31]	Roy et al. [32]	Meng et al. [33]	This study
Unsupervised?	No	No	No	Yes	Yes
Training Data Fully Allowing Co-activation?	No	No	No	No	Yes
Predicted Force Type	Multi-finger	Single-finger	Single-finger	Multi-finger	Multi-finger
R^2	0.71±0.11	Unreported	Unreported	0.77±0.036	0.79±0.039
RMSE (%MVC)	5.88±1.34	5.40±0.23	5.01±0.56	5.16±0.58	4.89±0.73

minimize crosstalk from non-target muscle compartments. Furthermore, sEMG-amplitude-based approaches were built on sEMG amplitudes, which were easily affected by background noise. This led to performance degradation, especially in challenging scenarios with high noise levels, as illustrated in Fig. 11. In contrast, the neural-drive approach predicted finger forces from a microscopic perspective. During sEMG decomposition, we identified two distinct clusters for each source signal. The cluster with a higher amplitude corresponded to the MU discharge events, while the other represented baseline noise, which was excluded from further analyses. The effective noise removal contributed to stable prediction performance under noisy conditions, demonstrating the robustness of the neural-drive approach in maintaining decoding accuracy despite varying levels of noise interference.

D. Comparison with Neural-Decoding Approaches

To demonstrate the superiority of our decoder, we conducted a comparison on the performance and characteristics of our neural decoder with recent state-of-the-art neural decoders from previous studies, as summarized in Table V. Specifically, Zheng et al. [30] ensured accurate force predictions by developing an effective neural decoding approach, which obtained target-finger-specific MU pools via carefully selected data sources (i.e., isolated single-finger movements) and MU refinement in a supervised manner. Built on this strategy, Rubin et al [31] demonstrated that MU decomposition was robust to forearm postural differences, as evidenced by the similar performance of finger force predictions across postures. To improve the decoding efficiency, Roy et al [32] employed a convolutional neural network using the neural-drive signals as ground truth to directly extract these neural-drive signals from sEMG data for single-finger force predictions. Although the developed neural decoding was promising, we further advanced the neural decoding approach by developing an unsupervised framework that eliminated the need for finger forces during decoder training. In addition, to train the neural decoders, previous studies [30], [31], [32] required subjects to perform single-finger tasks with intentional efforts of avoiding co-activations of non-target fingers. However, such single-finger tasks were inherently challenging to perform due to the intricate neural and mechanical interconnections among fingers, which naturally led to co-activations. Instead, we performed MU extraction directly using multi-finger trials, which were straightforward and natural for users to perform.

For performance comparison of finger force prediction performances, our decoder achieved a higher R^2 and a lower RMSE. This could primarily be due to the effective calculation of MU probabilities and removal of interfering MUs via our MU refinement procedure, as demonstrated in Fig. 12. In addition, as shown in Table II, the finger prediction performances were stable even when we intentionally introduced non-target finger MUs, further demonstrating the robustness of our decoder to non-target MUs. In comparison, for each MU, the supervised MU refinement strategy calculated the correlation (R^2) between the firing rate of this MU and the force of each finger. If the correlation for the target finger was the highest, the MU was retained regardless of the actual correlation value. This approach could retain interfering MUs with the highest correlation, thereby degrading the performance of finger force predictions. Recently, an unsupervised neural decoder was developed [33]. However, this study faced challenges similar to those encountered in previous supervised research, i.e., the requirement for isolated single-finger data to train the decoder. To further obtain MUs specific to a target finger, the MUs were grouped into three clusters, and the MU set with the highest mean population firing rate was selected. Those selected MUs were weighted based on the frequency of being retained across trials. In comparison, our newly developed MU refinement was more intuitive and straightforward for practical use. Our approach has been proven to be computationally efficient, performed well across a wide range of probability thresholds, robust to task variations, non-target MUs, and background noise.

E. Computational complexity analysis

In the training phase, as demonstrated in Section B of the supplementary material, the total computational complexity for the FastICA iterations was $O(m \cdot n_0 \cdot N^2) + O(m \cdot n_0 \cdot I \cdot D \cdot N)$. The symbols are defined in Table VI.

For the MU refinement procedure, we first decomposed the training trials using the obtained separation matrix ($\boldsymbol{B}_{1,l}$) to accelerate the sEMG decomposition process and extract source signals. The total complexity of this step was $O(m \cdot n_{1,l} \cdot D \cdot N)$. The main computational complexities for the remaining procedures were as follows: 1) Spike train calculation: $O(m \cdot n_{1,l} \cdot N)$; 2) Firing rate calculation: $O(m \cdot n_{1,l} \cdot N)$; 3) Plateau-period mean firing rate calculation: $O(m \cdot n_{1,l} \cdot P)$; 4) MU probability calculation: $O(M)$ and 5) MU refinement: $O(M)$. Compared with the FastICA iterations, our developed refinement procedure did not increase the overall complexity. The dominant complexity remains $O(m \cdot n_0 \cdot N^2) + O(m \cdot n_0 \cdot I \cdot D \cdot N)$.

TABLE VI
DEFINITIONS AND VALUES OF COMPUTATIONAL PARAMETERS

Notations	Descriptions	Value
m	Number of training trials	24
n_0	Number of initially set MU sources per trial	200
N	Number of sEMG data points per trial	24576 or 36864*
D	Number of extended sEMG channels	660
I	Maximum number of iterations for each MU	50
$n_{1,l}$	Number of separation vectors in $\boldsymbol{B_{1,l}}$	145.46±30.57
$n_{2,l}$	Number of separation vectors in $\boldsymbol{B_{2,l}}$	59.84±18.90
P	Number of plateaus in training trials	60
N_P	Number of sEMG data points in each segmentation	1024

* 24576 corresponds to two-finger trials, and 36864 corresponds to three-finger trials.

In the testing phase, 0.5-s segments of sEMG data (N_P =1024) were processed at each step for finger force predictions, with an average of 59.84±18.90 MUs ($n_{2,l}$) retained for finger force prediction, as shown in Table III. The computational complexity is primarily determined by the decomposition of testing trials, which has a complexity of $O(n_{2,l} \cdot D \cdot N_P)$. This results in an average computation time of 68.83±13.63 ms (as shown in Table III), demonstrating its feasibility for real-time applications.

F. Limitations

Although this study has taken a step forward towards finger force monitoring applications, this study focused solely on isometric contractions. In the future, we will validate our developed approach for dynamic contractions. In addition, this study did not include force prediction for the thumb due to its distinct anatomical structure and movement capabilities. The subsequent experiment will consider related physiological structures for the thumb force prediction. Furthermore, the participants were healthy individuals, we will validate the performance of our decoder on populations with hand disabilities or impairment, such as individuals with stroke or amputation. Lastly, computation time was obtained via pseudo real-time testing, where data segments were sequentially fed into the decoder. In future experiments, we will validate our decoder with data collected in real time.

VI. CONCLUSION

In this study, we developed an unsupervised neural-drive decoder for multi-finger force predictions. The new decoder demonstrated robustness across various probability thresholds (for MU pool refinement), data sources (for initial MU extraction and MU probability calculation), and background noise. Our results show that the derived MU probability can effectively capture the association between each MU and the target finger. By employing this strategy, we successfully removed MUs specific to non-target fingers, thereby leading to accurate force prediction performance. Additionally, the highly efficient computation under various protocols and probability thresholds offered a practical solution for real-time force monitoring applications in IoMT, holding potential to facilitate rehabilitation and health monitoring for individuals with hand disabilities. The ability of our decoder to accurately predict multi-finger forces in an unsupervised manner has profound implications for remote rehabilitation. It enables clinicians to monitor the recovery process in real time and adjust therapeutic interventions dynamically, thereby broadening the accessibility of rehabilitation services, particularly for individuals in remote or underserved regions. These advancements underscore the potential of our decoder to be implemented in the IoMT frameworks that require efficient, reliable, and adaptive solutions.

REFERENCES

[1] H. Diao *et al.*, "Real-time and cost-effective smart mat system based on frequency channel selection for sleep posture recognition in IoMT," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21421–21431, 2022.

[2] L. Meng *et al.*, "Automatic upper-limb Brunnstrom recovery stage evaluation via daily activity monitoring," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2589–2599, 2022.

[3] L. Meng *et al.*, "Exploration of human activity recognition using a single sensor for stroke survivors and able-bodied people," *Sensors*, vol. 21, no. 3, p. 799, 2021.

[4] G. J. Snoek *et al.*, "Survey of the needs of patients with spinal cord injury: impact and priority for improvement in hand function in tetraplegics," *Spinal Cord*, vol. 42, no. 9, pp. 526–532, 2004.

[5] M. W. Raad *et al.*, "An IOT based Wearable Smart Glove for Remote Monitoring of Rheumatoid Arthritis Patients.," *Biosignals*, vol. 2019, pp. 224–228, 2019.

[6] J. Vinstrup *et al.*, "Hand strengthening exercises in chronic stroke patients: Dose-response evaluation using electromyography," *J. Hand Ther.*, vol. 31, no. 1, pp. 111–121, 2018.

[7] J. H. Bae *et al.*, "Relationship between grip and pinch strength and activities of daily living in stroke patients," *Ann. Rehabil. Med.*, vol. 39, no. 5, p. 752, 2015.

[8] A. M. Bertrand *et al.*, "Reliability of maximal grip strength measurements and grip strength recovery following a stroke," *J. Hand Ther.*, vol. 28, no. 4, pp. 356–363, 2015.

[9] C. Mercierand *et al.*, "Rlative shoulder fexor and handgrip strength is related to upper limb function after stroke," *Clin. Rehabil.*, vol. 18, no. 2, pp. 215–221, Mar. 2004.

[10] E. T. Wolbrecht *et al.*, "Finger strength, individuation, and their interaction: Relationship to hand function and corticospinal tract injury after stroke," *Clin. Neurophysiol.*, vol. 129, no. 4, pp. 797–808, 2018.

[11] Y. Furudate *et al.*, "Quantification method of motor function recovery of fingers by using the device for home rehabilitation," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2017, pp. 3872–3875.

[12] O. Postolache *et al.*, "Remote monitoring of physical rehabilitation of stroke patients using IoT and virtual reality," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 562–573, 2020.

[13] C. Y. Neng *et al.*, "Internet of things (iot) based flex force smart glove for physical rehabilitation," in *2020 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, IEEE, 2021, pp. 276–280.

[14] Y. N. Chang *et al.*, "Restorative Hand Therapy Exercise using IoT-Based Flex Force Smart Glove," *Makara J. Technol.*, vol. 27, no. 1, p. 4.

[15] Q. Wu *et al.*, "Regularized Group Sparse Discriminant Analysis for P300-Based Brain–Computer Interface," *Int. J. Neural Syst.*, vol. 29, no. 06, Art. no. 06, Aug. 2019.

[16] T. Lenzi *et al.*, "Intention-based EMG control for powered exoskeletons," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 8, Art. no. 8, 2012.

[17] S. M. Hosseini *et al.*, "State-Based Decoding of Continuous Hand Movements using EEG Signals," *IEEE Access*, 2023.

[18] V. Aggarwal *et al.*, "State-based decoding of hand and finger kinematics using neuronal ensemble and LFP activity during dexterous reach-to-grasp movements," *J. Neurophysiol.*, vol. 109, no. 12, pp. 3067–3081, Jun. 2013.

[19] A. Ahmadi *et al.*, "State-based decoding of force signals from multi-channel local field potentials," *IEEE Access*, vol. 8, pp. 159089–159099, 2020.

[20] B. Farrokhi *et al.*, "A state-based probabilistic method for decoding hand position during movement from ECoG signals in non-human primate," *J. Neural Eng.*, vol. 17, no. 2, p. 026042, 2020.

[21] A. Y. Paek *et al.*, "Regression-based reconstruction of human grip force trajectories with noninvasive scalp electroencephalography," *J. Neural Eng.*, vol. 16, no. 6, p. 066030, 2019.

[22] A. Y. Paek *et al.*, "Predicting hand forces from scalp electroencephalography during isometric force production and object grasping," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015, pp. 7570–7573.

[23] X. Jiang *et al.*, "Optimizing the Cross-Day Performance of Electromyogram Biometric Decoder," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4388–4402, 2022.

[24] L. Meng *et al.*, "Evaluation of decomposition parameters for high-density surface electromyogram using fast independent component analysis algorithm," *Biomed. Signal Process. Control*, vol. 75, p. 103615, 2022.

[25] D. Leonardis *et al.*, "An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation," *IEEE Trans. Haptics*, vol. 8, no. 2, Art. no. 2, 2015.

[26] A. Fougner *et al.*, "Control of upper limb prostheses: Terminology and proportional myoelectric control—A review," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 5, Art. no. 5, 2012.

[27] J. G. Ngeo *et al.*, "Continuous and simultaneous estimation of finger kinematics using inputs from an EMG-to-muscle activation model," *J. NeuroEngineering Rehabil.*, vol. 11, no. 1, Art. no. 1, Dec. 2014.

[28] D. Farina *et al.*, "Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation," *Nat. Biomed. Eng.*, vol. 1, no. 2, Art. no. 2, 2017.

[29] C. Dai *et al.*, "Prediction of individual finger forces based on decoded motoneuron activities," *Ann. Biomed. Eng.*, vol. 47, no. 6, pp. 1357–1368, 2019.

[30] Y. Zheng *et al.*, "Concurrent prediction of finger forces based on source separation and classification of neuron discharge information," *Int. J. Neural Syst.*, vol. 31, no. 06, Art. no. 06, 2021.

[31] N. Rubin *et al.*, "Finger force estimation using motor unit discharges across forearm postures," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 9, pp. 2767–2775, 2022.

[32] R. Roy *et al.*, "A generic neural network model to estimate populational neural activity for robust neural decoding," *Comput. Biol. Med.*, vol. 144, p. 105359, 2022.

[33] L. Meng *et al.*, "Unsupervised neural decoding for concurrent and continuous multi-finger force prediction," *Comput. Biol. Med.*, vol. 173, p. 108384, 2024.

[34] L. Meng *et al.*, "Unsupervised Decoding of Multi-Finger Forces Using Neuronal Discharge Information with Muscle Co-Activations," in *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, IEEE, 2024, pp. 1–4.

[35] F. Gao *et al.*, "Matrix analyses of interaction among fingers in static force production tasks," *Biol. Cybern.*, vol. 89, no. 6, pp. 407–414, 2003.

[36] M. H. Schieber *et al.*, "Selective Activation of Human Finger Muscles after Stroke or Amputation," in *Progress in Motor Control*, vol. 629, D. Sternad, Ed., in Advances in Experimental Medicine and Biology, vol. 629. , Boston, MA: Springer US, 2009, pp. 559–575.

[37] V. M. Zatsiorsky *et al.*, "Enslaving effects in multi-finger force production," *Exp. Brain Res.*, vol. 131, pp. 187–195, 2000.

[38] X. Hu *et al.*, "Extracting extensor digitorum communis activation patterns using high-density surface electromyography," *Front. Physiol.*, vol. 6, p. 279, 2015.

[39] Y. Zheng *et al.*, "Interference removal from electromyography based on independent component analysis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 5, Art. no. 5, 2019.

[40] M. Chen *et al.*, "Two-Source Validation of Progressive FastICA Peel-Off for Automatic Surface EMG Decomposition in Human First Dorsal Interosseous Muscle," *Int. J. Neural Syst.*, vol. 28, no. 09, Art. no. 09, Nov. 2018.

[41] M. Chen *et al.*, "A novel framework based on FastICA for high density surface EMG decomposition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 1, Art. no. 1, 2015.

[42] F. Negro *et al.*, "Multi-channel intramuscular and surface EMG decomposition by convolutive blind source separation," *J. Neural Eng.*, vol. 13, no. 2, Art. no. 2, 2016.

[43] A. Hyvärinen *et al.*, "Independent component analysis: algorithms and applications," *Neural Netw.*, vol. 13, no. 4–5, Art. no. 4–5, 2000.

[44] C. Dai *et al.*, "Independent component analysis based algorithms for high-density electromyogram decomposition: Systematic evaluation through simulation," *Comput. Biol. Med.*, vol. 109, pp. 171–181, 2019.

[45] E. A. Clancy *et al.*, "Probability density of the surface electromyogram and its relation to amplitude detectors," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 6, Art. no. 6, 1999.

[46] Z. Zeng *et al.*, "Unsupervised Transfer Learning Approach With Adaptive Reweighting and Resampling Strategy for Inter-subject EOG-based Gaze Angle Estimation," *IEEE J. Biomed. Health Inform.*, 2023.

[47] Z. Zeng *et al.*, "A robust gaze estimation approach via exploring relevant electrooculogram features and optimal electrodes placements," *IEEE J. Transl. Eng. Health Med.*, 2023.

[48] J. E. Downey *et al.*, "Motor cortical activity changes during neuroprosthetic-controlled object interaction," *Sci. Rep.*, vol. 7, no. 1, Art. no. 1, 2017.

[49] P. Heo *et al.*, "Current hand exoskeleton technologies for rehabilitation and assistive engineering," *Int. J. Precis. Eng. Manuf.*, vol. 13, pp. 807–824, 2012.

[50] J. Cohen, *Statistical power analysis for the behavioral sciences*. Academic press, 2013

Long Meng received his B.S. degree in applied physics from Northeastern University, Shenyang, China, in 2011. He then received his M.S. degree in circuits and systems and his Ph.D. degree in Biomedical Engineering from Fudan University, Shanghai, China, in 2018 and 2023, respectively. He has worked as a postdoctoral fellow at the Penn State Neuromechanics Lab since 2023.

His research interests lie in biomedical engineering, focusing on wearable sensor systems, neural prosthetic control, stroke rehabilitation, and personalized health monitoring.

Xiaogang Hu (Senior Member, IEEE) received the B.E. degree in mechanical engineering from Tsinghua University. He was trained in motor control and biomechanics at The Pennsylvania State University during his Ph.D. study. He completed his postdoctoral training with the Rehabilitation Institute of Chicago. He is currently the Dorothy Foehr Huck and J. Lloyd Huck Chair of Neurorehabilitation and an Associate Professor with the Department of Mechanical Engineering, Department of Kinesiology, and Department of Physical Medicine and Rehabilitation, the Huck Institutes of the Life Sciences, and the Center for Neural Engineering, The Pennsylvania State University.

His research interests include neural-machine interface and neural stimulation, targeting upper limb sensorimotor functions of individuals after stroke, and traumatic brain injury, or limb loss.