Decomposing motor units through elimination for real-time intention driven assistive neurotechnology

Nicholas Tacca[1] **Bryan R. Schlink**[1] **Jackson T. Levine**[2,3] **Mary K. Heimann**[1]
Collin Dunlap[1] **Samuel C. Colachis IV**[1] **Philip Putnam**[1] **Matthew A. Zeglen**[1]
Daniel J. Brobston[4] **Austin M. Bollinger**[4] **José L. Pons**[2,3]
Lauren Wengerd[1,5] **Eric C. Meyers**[4] **David A. Friedenberg**[1]

[1]Battelle Memorial Institute [2]Shirley Ryan AbilityLab [3]Northwestern University
[4]University of Texas at Dallas [5]The Ohio State University

{tacca,friedenbergd}@battelle.org

Abstract

Extracting neural signals at the single motor neuron level provides an optimal control signal for neuroprosthetic applications. However, current algorithms to decompose motor units from high-density electromyography (HD-EMG) are time-consuming and inconsistent, limiting their application to controlled scenarios in a research setting. We introduce MUelim, an algorithm for efficient motor unit decomposition that uses approximate joint diagonalization with a subtractive approach to rapidly identify and refine candidate sources. The algorithm incorporates an extend-lag procedure to augment data for enhanced source separability prior to diagonalization. By systematically iterating and eliminating redundant or noisy sources, MUelim achieves high decomposition accuracy while significantly reducing computational complexity, making it well-suited for real-time applications. We validate MUelim by demonstrating its ability to extract motor units in both simulated and physiological HD-EMG grid data. Across six healthy participants performing ramp and maximum voluntary contraction paradigms, MUelim achieves up to a 36× speed increase compared to existing state-of-the-art methods while decomposing a similar number of high signal-to-noise sources. Furthermore, we showcase a real-world application of MUelim in a clinical setting in which an individual with spinal cord injury controlled an EMG-driven neuroprosthetic to perform functional tasks. We demonstrate the ability to decode motor intent in real-time using a spiking neural network trained on the decomposed motor unit spike trains to trigger functional electrical stimulation patterns that evoke hand movements during task practice therapy. We show that motor unit-based decoding enables nuanced motor control, highlighting the potential of MUelim to advance assistive neurotechnology and rehabilitation through precise, intention-driven neuroprosthetic systems.

1 Introduction

Electromyography (EMG) provides a natural motor interface for humans to interact with machines, capturing the electrical activity of muscles to facilitate intuitive control. At the core of physiological motor control is the motor unit, the smallest functional unit of muscle activation, consisting of a single motor neuron and the fibers it innervates [Heckman and Enoka, 2012]. EMG signals recorded at the periphery consist of a summation of motor unit action potentials and noise, with the input signal originating from the central nervous system, providing a direct link to neuromotor intent [Farina and Negro, 2015]. As a result, decomposing the physiological motor input to muscles via motor unit

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

activity may provide a more advanced and intuitive control signal for neuroproshetics [Tanzarella et al., 2023, Chen et al., 2020, Kapelner et al., 2019].

Recently, there have been significant advances in motor unit decomposition from non-invasive high-density EMG (HD-EMG), leveraging swarm contrastive decomposition (SCD) [Grison et al., 2025], convolutive blind source separation (BSS) [Negro et al., 2016, Holobar et al., 2014], convolutional kernel compensation (CKC) [Holobar and Zazula, 2007], and deep learning-based approaches [Lin et al., 2024, Wen et al., 2023]. Open-source tools [Grison et al., 2025, Avrillon et al., 2024, Formento et al., 2021] have made it possible for researchers to analyze motor unit activity in controlled research settings during both isometric and dynamic movements [Osswald et al., 2025, Tanzarella et al., 2023, Chen et al., 2020]. However, these methods remain computationally intensive and inconsistent, often requiring significant processing time or manual intervention [Del Vecchio et al., 2020, Negro et al., 2016, Farina et al., 2014], prohibiting their use in real-time neuroprosthetic systems, where rapid and reliable decoding of motor intent is essential.

Existing decomposition methods typically use blind source separation (BSS), such as independent component analysis, in which the contrast functions used for source separation measure sparseness of spike trains, rather than independence [Negro et al., 2016]. While effective, these higher-order statistical methods are computationally demanding and sensitive to noise [Congedo et al., 2008]. Second-order statistical (SOS) methods [Belouchrani et al., 1997, Belouchrani and Amin, 1998] offer a more efficient alternative by leveraging spectral and temporal signatures of sources through joint diagonalization. However, SOS methods face two key challenges, namely joint diagonalization, a core component of SOS methods, has historically been computationally expensive, and SOS methods are constrained to decompose at most as many sources as there are recording channels [Congedo et al., 2008, Pham and Cardoso, 2001, Pham, 2001].

Recent advancements in joint diagonalization techniques have significantly improved computational efficiency [de Vlaming and Slob, 2021, Ablin et al., 2018], making SOS methods more practical for real-world applications. To address the challenge of limited source separability, we propose using strategies based on existing decomposition methods [Grison et al., 2025, Negro et al., 2016] and electroencephalography data augmentation techniques [Carrara and Papadopoulo, 2024] to add time-delayed extensions prior to joint diagonalization. This augmentation increases the effective dimensionality and enhances source separability by capturing temporal dynamics, improving the conditioning of the source separation problem. Taking inspiration from compressive sensing [Candès et al., 2006, Donoho et al., 2005], we further accelerate the identification of candidate sources by leveraging the sparsity of motor unit source activity contained within symmetric positive definite (SPD) matrices. By randomly sampling SPD matrices for joint diagonalization, we can perform an iterative BSS to find unique sources while significantly reducing computational overhead.

With these considerations in mind, we propose MUelim, a motor unit decomposition algorithm that leverages approximate joint diagonalization to extract many candidate motor unit sources all at once and then eliminates suspected noise sources. We validate MUelim on both simulated and physiological HD-EMG datasets, demonstrating its ability to extract motor units with high accuracy and speed. Furthermore, we showcase its application in a clinical setting, where it enables a spinal cord injury (SCI) participant to control a neuroprosthetic device in real time, highlighting its potential to advance assistive neurotechnology and rehabilitation.

2 Methods

2.1 Overview of EMG decomposition as a blind source separation problem

EMG signals represent the summation of motor unit action potentials (MUAPs) and noise (Figure 1A). These signals can be modeled as a linear instantaneous mixture of sparse sources, where each source corresponds to the discharge timings of a motor unit. The observed EMG signals $\mathbf{X} \in \mathbb{R}^{C \times N}$, where C is the number of channels and N is the number of samples, can be expressed as:

$$\mathbf{X}(t) = \mathbf{H}\mathbf{S}(t) + \mathbf{N}(t), \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{C \times M}$ is the mixing matrix with M motor unit sources, $\mathbf{S}(t) \in \mathbb{R}^{M \times N}$ represents the spike trains, and $\mathbf{N}(t)$ is additive noise. The goal of EMG decomposition is to estimate the spike trains $\mathbf{S}(t)$ from the observed signals $\mathbf{X}(t)$.

Motor unit decomposition can be framed as a BSS problem, where the objective is to estimate the separating matrix $\mathbf{B} \in \mathbb{R}^{M \times C}$ such that:

$$\hat{\mathbf{S}}(t) = \mathbf{B}\mathbf{X}(t), \tag{2}$$

where $\hat{\mathbf{S}}(t)$ is the estimated motor unit source activity.

2.2 MUelim algorithm

2.2.1 Preprocessing and data augmentation

The input EMG data $\mathbf{X}$ is first divided into non-overlapping windows of size L for SPD matrix computation. The windowed data is represented as $\mathbf{X}_{\text{binned}} \in \mathbb{R}^{W \times C \times L}$, where W is the number of windows.

To incorporate temporal information, the data is extended using lagged versions of each channel. This augmentation creates an extended dataset $\mathbf{X_{ext}} \in \mathbb{R}^{W \times (C \cdot R) \times L}$, where R is the extension factor. The extend-lag procedure increases the ratio of observations to sources, improving the conditioning of the source separation problem [Holobar and Zazula, 2007]. By embedding the data into a higher-dimensional space, this approach captures both spatial and temporal dependencies, which are critical for resolving sources with overlapping activity.

The extend-lag procedure is inspired by the success in decoding from the tangent space of augmented covariance matrices [Carrara and Papadopoulo, 2024]. The augmented covariance matrix combines spatial covariance with temporal information, effectively embedding the original dataset into a higher-dimensional space. This embedding enhances the separability of sources by capturing their nonlinear dynamics [Takens, 2006]:

$$\mathbf{X}_{\text{ext}}(t) = [\mathbf{X}_{\text{binned}}(t), \mathbf{X}_{\text{binned}}(t - \tau), \ldots, \mathbf{X}_{\text{binned}}(t - (R-1)\tau)]^{\top}, \tag{3}$$

where τ is the lag parameter, and R is the extension factor.

2.2.2 Iterative blind source separation (BSS)

MUelim employs an iterative BSS approach to identify and refine sources. The algorithm assumes that the sources are sparse in the transformed domain. The iterative process continues until a stopping criterion is met, such as reaching the maximum number of iterations, finding the maximum number of sources, or failing to identify a minimum number of new sources.

SPD matrix computation In each iteration, the extended dataset is sampled ($\mathbf{X}'_{ext}$) for SPD matrix computation. Depending on the characteristics of the data and the assumptions about the sources, several types of SPD matrix estimators can be used. Simple covariance matrices, channelwise kernels, such as linear, polynomial, or laplacian kernels, and cospectral matrices can be used to capture nonlinear dynamics of the system [Congedo et al., 2008, Belouchrani and Amin, 1998, Belouchrani et al., 1997].

In this study, cospectral matrices are used as the primary estimator due to their ability to capture both spatial and spectral dependencies. For each frequency f in the range $[f_{\text{min}}, f_{\text{max}}]$, the Fast Fourier transform is applied to each window of the extended data to compute the cospectral matrix $\mathbf{C}_f$:

$$\mathbf{C}_f = \frac{1}{W} \sum_{k=1}^{W} \mathbf{X}'_{\text{ext},f}[k] \mathbf{X}'_{\text{ext},f}[k]^H, \tag{4}$$

where H denotes the Hermitian transpose. The diagonal elements of $\mathbf{C}_f$ represent the power (auto-spectra) of each channel, while the off-diagonal elements represent the in-phase SOS dependency [Congedo et al., 2008].

Whitening To improve the numerical conditioning of the SPD matrices, whitening is applied. Whitening ensures that the mean SPD matrix $\bar{\mathbf{C}}$ has an identity covariance structure, simplifying the subsequent diagonalization step. This is achieved through eigenvalue decomposition:

$$\mathbf{W}_{\text{whiten}} = \mathbf{V}\mathbf{\Lambda}^{-1/2}\mathbf{V}^T, \quad \mathbf{C}_f^{\text{whiten}} = \mathbf{W}_{\text{whiten}} \mathbf{C}_f \mathbf{W}_{\text{whiten}}^T, \tag{5}$$

where $\mathbf{V}$ and $\mathbf{\Lambda}$ are the eigenvectors and eigenvalues of $\bar{\mathbf{C}}$.

Approximate joint diagonalization The separation of non-stationary signals can be achieved by joint diagonalization of a set of autocorrelation matrices [Belouchrani and Amin, 1998, Belouchrani et al., 1997]. In the context of MUelim, we consider a set of n SPD matrices $\{\mathbf{C}_1, \ldots, \mathbf{C}_n\}$ of size $p \times p$, where each matrix $\mathbf{C}_i$ represents a whitened SPD matrix computed from the extended EMG data (Eq. 4). The goal of joint diagonalization is to find a matrix $\mathbf{B} \in \mathbb{R}^{p \times p}$ such that the transformed set $\{\mathbf{B}\mathbf{C}_1\mathbf{B}^\top, \ldots, \mathbf{B}\mathbf{C}_n\mathbf{B}^\top\}$ contains matrices that are as diagonal as possible. This is achieved by minimizing the following joint diagonalization criterion:

$$\mathcal{L}(\mathbf{B}) = \frac{1}{2n} \sum_{i=1}^{n} \left[\log\det \operatorname{diag}\left(\mathbf{B}\mathbf{C}_i\mathbf{B}^\top\right) - \log\det\left(\mathbf{B}\mathbf{C}_i\mathbf{B}^\top\right) \right], \tag{6}$$

where $\operatorname{diag}(\cdot)$ extracts the diagonal elements of a matrix, and $\det(\cdot)$ denotes the determinant. This criterion, introduced by Pham [2001], is derived as the negative log-likelihood of a source separation model for Gaussian stationary sources.

After solving the joint diagonalization problem, the forward and backward filters are computed. The forward filters $\mathbf{W}_{\text{forward}}$ are used to project the extended EMG data into the source space, while the backward filters $\mathbf{W}_{\text{backward}}$ are used to reconstruct the original signals from the source space. Let $\mathbf{W}_{\text{whiten}}$ and $\mathbf{W}_{\text{whiten}}^{-1}$ represent the whitening and inverse whitening filters, respectively. The forward and backward filters are computed as:

$$\mathbf{W}_{\text{forward}} = \mathbf{B}\mathbf{W}_{\text{whiten}}^\top, \tag{7}$$

$$\mathbf{W}_{\text{backward}} = \mathbf{W}_{\text{whiten}}^{-1}\mathbf{B}^{-1}. \tag{8}$$

Peak detection and source refinement After obtaining the forward filters from the joint diagonalization step, the full extended dataset is transformed into the source domain. For each source, the source power $\gamma_j(k)$ is computed as:

$$\gamma_j(k) = (\mathbf{W}_{\text{forward}}\mathbf{X}_{\text{ext}})^2, \tag{9}$$

Peaks are detected in the source power to refine each source filter $\mathbf{w}_j$ as the mean of the corresponding impulse indices:

$$\mathbf{w}_j = \frac{\sum_{k \in \text{peaks}} \mathbf{X}_{\text{ext}}(k)}{\left\| \sum_{k \in \text{peaks}} \mathbf{X}_{\text{ext}}(k) \right\|_2}. \tag{10}$$

The updated filter is then orthogonalized and normalized with respect to the previously identified filters to ensure independence:

$$\mathbf{w}_j \leftarrow \mathbf{w}_j - \mathbf{W}_{\text{forward}}^\top(\mathbf{W}_{\text{forward}}\mathbf{w}_j), \tag{11}$$

$$\mathbf{w}_j \leftarrow \frac{\mathbf{w}_j}{\|\mathbf{w}_j\|_2}. \tag{12}$$

This process is repeated iteratively until convergence, ensuring that the filters are optimized for the identified sources. During this improvement iteration step, k-means clustering is used to separate signal peaks from noise peaks detected in the source power. The silhouette score (SIL) for each source is then calculated from the detected peaks. Sources that do not converge or that have a low SIL are eliminated from the final forward filters ($\mathbf{W}_{\text{forward}}$).

Unique source identification Newly identified spatial filters per BSS iteration are compared with previously found filters to ensure uniqueness. Spatial similarity is assessed using the cosine similarity between filters. Temporal similarity between spike trains is evaluated based on the percentage of coincident spike timings between motor unit spike trains. Newly identified candidate sources that exceed predefined similarity thresholds are discarded.

Optionally, to remove the influence of already identified sources, existing sources can be peeled off from the data [Chen and Zhou, 2015]. This step may improve the detection of weaker sources in subsequent iterations by removing the contribution of already-identified sources. Refer to Supplementary Algorithm 1 for pseudocode and Supplementary Figure 1 for an overview of the algorithm.

2.2.3 Computational complexity

MUelim's computational complexity is primarily determined by the joint diagonalization step, which is performed on the extended data matrix. The data is extended using R lagged versions of each channel, resulting in $C \times R$ extended channels, where C is the number of original channels and R is the extension factor. The per-iteration computational cost is $\mathcal{O}(K(C \times R)^2 + (C \times R)^3)$, where K is the number of matrices being diagonalized. For large $C \times R$, the $(C \times R)^3$ term dominates, so the complexity is approximately $\mathcal{O}((C \times R)^3)$ per iteration. This scaling arises because the most expensive operation is matrix multiplications of size $(C \times R) \times (C \times R)$. As channels or extension factor increases, decomposition time increases accordingly. However, MUelim's advantage becomes more pronounced with more motor unit sources, as it efficiently extracts many sources simultaneously rather than incrementally.

2.3 Algorithm evaluation

To validate the proposed algorithm, we tested the decomposition method on simulated EMG data. The simulated EMG signals were generated using a Poisson neuron model to create spike trains, which were then convolved with a MUAP template to produce signals across multiple channels. Random noise and amplitude modulation were incorporated to mimic the variability and complexity of real physiological signals. We tested MUelim, SCD [Grison et al., 2025], and MUEdit [Avrillon et al., 2024] on identical simulated datasets to enable direct comparison of decomposition performance.

We evaluated two channel configurations spanning the physiologically realistic range for motor unit decomposition: 32 channels with 5, 10, and 15 motor unit sources, and 64 channels with 5, 10, 15, 20, 25, and 30 motor unit sources. These configurations test up to $N/2$ sources across all three methods. Signals were generated using a sampling rate of 2,048Hz with average motor unit firing rate of 10Hz for 30s total. We assessed decomposition accuracy by comparing detected spike timings with ground truth within a 25ms window, and calculated false positive and false negative rates for each method.

To assess the feasibility of using the MUelim algorithm in application, we decomposed motor units from a 8x8 HD-EMG grid (GR10MM0808; 64 monopolar EMG channels) and Quattrocento amplifier (OT Bioelettronica) across different parameters. EMG signals were recorded from the flexor digitorum superficialis muscle in six healthy participants (N=6). Participants performed two experimental paradigms: a ramped contraction for 45s, with the first 15s ramping up, the next 15s plateaued at approximately 30% of the maximum voluntary contraction, and the last 15s ramping down, and a maximum voluntary contraction (MVC) sustained for 10s. EMG measurements were taken at a sampling rate of 2,048Hz. An initial bandpass filter for the incoming signal was set at 10-900Hz for each recording, and then an offline filter of 20-500Hz was applied to the recordings. Force measurements were taken using a digital hand dynamometer at 10Hz and was synced with the EMG signal offline.

Following decomposition, we plotted spike trains superimposed over target and force trajectories. We extracted MUAP waveforms using spike-triggered averaging (STA), which averages the EMG signal around detected spike times to isolate the waveform associated with each motor unit. For each spike timing, we extracted a 30ms window centered on the spike impulse. To reduce cross-contamination, we reconstructed the EMG signal from source space using the BSS inverse filters while suppressing other motor unit sources. Waveforms were aligned to their global maxima across channels to handle temporal jitter before averaging. We extracted waveforms from all channels and displayed results from the 5 most active channels based on signal amplitude. Individual traces show the average computed over a sliding window spanning one fifth of all detected MUAPs, with a stride equal to one sixth of the window length. Subsequent inverse spatial filters were mapped to the 2D grid to visualize source activity dipoles.

We conducted two tests to benchmark the number of decomposed sources and decomposition time across different parameters in a single representative participant. First, we assessed the influence of the extension factor and lag on decomposition performance. The extension factor was set between 1 and 6, with the lag between 1 and 10 for a total of 60 permutations. A cospectral estimator and bin size of 20ms was used across all permutations. In the second test, we evaluated the influence of the SPD estimator, as well as the variability in bin size to compute SPD matrices. We evaluated a cospectral estimator and laplacian channelwise kernel over a range of bin sizes (10-2000ms) using open-source methods from pyRiemann [Barachant et al., 2025]. For all cospectral matrices, we computed the FFT using a window size of 8 with 75% overlap between 0 and $60Hz$. For this test, we held the extension factor and lag constant at 2 extensions and 6 lag samples, respectively. For both tests, we ran each decomposition permutation over 5 seeds to determine the consistency of the algorithm with the given set of parameters. The average number of sources decomposed and decomposition time ± standard error of the mean are reported. A SIL threshold of 0.85 was used to determine high signal-to-noise sources. We found sources over 3 BSS iterations using a random sampling of 50% of the data for diagonalization of SPD matrices. For all tests, JADOC [de Vlaming and Slob, 2021] was used for joint diagonalization with a regularization strength of 0.85.

We then compared our results to current state-of-the-art open-source motor unit decomposition algorithms across all six participants, namely SCD [Grison et al., 2025] and MUEdit [Avrillon et al., 2024]. We assessed both number of sources decomposed and decomposition time. We held all other parameters constant based on the open-source implementations, with 0.85 SIL threshold used to determine acceptable sources. For SCD, we used the built-in GPU capability to increase decomposition speed. All tests were evaluated using a HP ZBook Power 15.6 inch G8 Mobile Workstation PC (Intel Core i7-11850H 2.50GHz, NVIDIA T1200 GPU, CUDA 12.2) python 3.11 or MATLAB 2021 for MUEdit.

2.4 Application: Intention driven neuroprosthetic

To demonstrate MUelim's utility in a realistic use-case, we evaluated its performance in an ongoing registered clinical trial (NCT06087445) in an individual with SCI. All study procedures were conducted under Institutional Review Board approval. The participant was informed of potential risks and monitored for adverse events throughout the study. In the study, a C3 incomplete SCI participant (ASIA D) used an EMG-based neuroprosthetic system to control movement-specific functional electrical stimulation (FES) patterns based on motor intent. The participant could partially open and close his hand, though not fully, and exhibited limited grip strength. Abnormal muscle synergies, particularly overactivity in the wrist flexors, limited his ability to achieve wrist extension. FES facilitated more complete hand opening, enhancing functional performance and reducing reliance on compensatory strategies. The complete training and inference workflow is illustrated in Supplementary Figure 3.

The neuroprosthetic system consists of 150 electrodes (75 bipolar EMG channels) embedded into a stretchable fabric. An operator manually calibrated FES patterns at the beginning of the session to evoke both hand open and hand close movements (Supplementary Video 1: Heatmap / Stim Patterns). A sinusoidal waveform with $20Hz$ stimulation frequency was used for both patterns. Next, a block of operator-guided stimulation was used to find spatial filters for motor unit decomposition and train a spiking neural network (SNN) to decode motor intent from the decomposed motor unit spike trains. The operator manually cued the participant to perform different tasks guided by an occupational therapist (OT). The corresponding FES pattern was automatically stimulated with an offset of 1,000ms to record intent with/without FES active. The BSS forward filters were computed in-session after collecting this training block (fit phase), then applied for real-time inference (transform phase), following standard scikit-learn convention (Supplementary Algorithm 1). EMG was recorded with a sampling rate of 3,000Hz, collected in non-overlapping $100ms$ bins to generate final predictions at $10Hz$ used to switch/turn-off FES patterns when in inference mode.

Following this training block, EMG data was filtered for subsequent motor unit decomposition. First, FES artifact was removed using a template artifact filter, whereby a template of the artifact from the previous pulse window was subtracted from the current pulse window. Next, a $20ms$ blanking window was used to blank out any remaining artifact in between FES pulses. The subsequent signal was then filtered using notch and bandpass filters similar to previous studies [Tacca et al., 2024, Meyers et al., 2024]. Following the preprocessing pipeline, motor units were decomposed using the proposed MUelim algorithm to compute the BSS forward filters from the training data. For enhanced speed and accuracy, motor unit spike trains were downsampled within a bin to five sub-bin splits, with a shape of $n_{samp} \times n_{MUs} \times n_{splits}$. Next, the previous four bins were concatenated with the current bin to provide additional time information to the SNN. This was then fed into a fully-connected SNN developed using the LAVA framework. Hidden layers consisted of 1000 x 500 current based leaky integrate and fire neurons. Default neuron parameters from LAVA were used for training. A dropout of 40% was applied at each layer. The SNN was trained natively using the SLAYER module [Shrestha and Orchard, 2018] within LAVA for 15 epochs with batch size of 128 and an initial learning rate of 0.001. The Adam optimizer with cosine annealing warm restarts was used to optimize the error function. The built-in spike rate error function with true rate of 0.8 and false rate of 0.001 was used to align the output SNN neuron activity with the three classes, namely Hand Open, Hand Close, and Rest. Both SNN training and inference utilized a GPU to enhance speed for application. All steps within the pipeline were wrapped in a sklearn transformer [Pedregosa et al., 2011] containing both fit and transform methods to be used in the existing clinical software.

Once a decoder was trained, the full pipeline was used in inference mode, allowing the participant to control FES based on motor intent decoded from the decomposed motor unit activity. The operator manually labeled the task as the participant performed it with guidance from the OT to assess decoder performance offline. Bin-wise accuracy was used to assess how many $100ms$ bins matched the ground truth manually labeled by the operator. A live user-interface was generated to visualize EMG activity, motor unit activity, SNN neuron output activity, and MUAP waveform generation. A Unity-engine virtual hand shows the real-time predictions and active FES provided to the participant.

3 Results

3.1 MUelim decomposition validation

Representative decomposition results from simulated EMG data are shown in Figure 1B-D. MUelim successfully decomposed the simulated signals into the three ground truth motor unit sources. Peak detection in the source power domain (Figure 1B) identified spike timings that closely matched ground truth (Figure 1C). Reconstructing the EMG signal at these spike timings with other sources suppressed produced clear motor unit action potential waveforms (Figure 1D), demonstrating successful recovery of the simulated MUAP shapes.

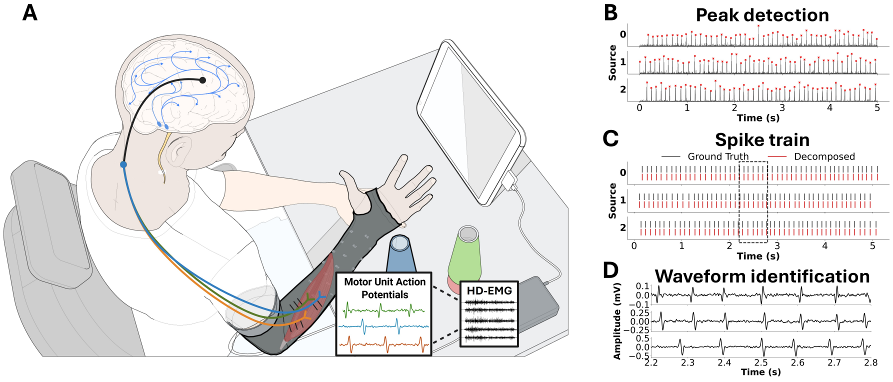

Figure 1: **Overview of MUelim decomposition algorithm. A.** Motor units, each consisting of a motor neuron and the muscle fibers it innervates, generate action potentials that sum to form high-density electromyography (HD-EMG). In this work, we present MUelim, an efficient motor unit decomposition algorithm, that decomposes EMG into motor unit spike trains to provide an intuitive control signal for assistive and rehabilitative neurotechnology. **B.** Peak detection from source power of the identified sources decomposed from simulated EMG signals. Identified peak impulses have high signal to noise ratio. **C.** Spike timings align with ground truth spike timings. **D.** Waveform identification in the first EMG channel through the reconstruction of EMG signal from source domain with other sources suppressed.

We compared MUelim against state-of-the-art decomposition methods on identical simulated EMG datasets. Across all simulation scenarios (32 and 64 channels with 5 to 30 motor units), all three methods successfully identified the correct number of sources. When comparing spike timings with ground truth, MUelim achieved 99.99 ± 0.01% accuracy with 0.99 ± 0.14 false positives per source and 0.03 ± 0.02 false negatives per source, closely matching SCD (99.99 ± 0.01% accuracy, 0.93 ± 0.13 FP per source, 0.03 ± 0.02 FN per source) and MUEdit (98.98 ± 0.07% accuracy, 0.00 ± 0.00 FP per source, 3.06 ± 0.20 FN per source) (Supplementary Table 1).

Next, motor units were decomposed from a HD-EMG grid as a healthy individual performed a ramped contraction (Figure 2). Varying parameters were used to assess the impact on the number of sources decomposed and decomposition time (Supplementary Figure 2). Using an extension factor of 4 and lag of 6, 19 unique motor unit sources were found in just under 5 minutes that align closely to the force output (Figure 2A). Representative MUAP waveforms (Figure 2B) demonstrate clean action potentials, while the inverse spatial filters (Figure 2C) show the spatial localization of motor unit sources across the electrode grid.

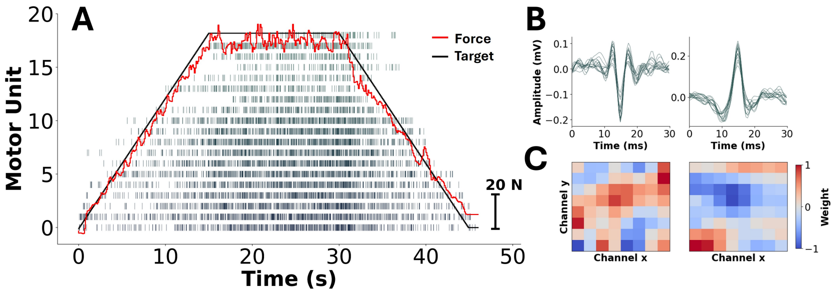

Figure 2: **Decomposition results align with force output in a healthy individual A.** Spike train decomposed using MUelim algorithm aligned with target (black) and force (red) trajectories **B.** Motor unit action potential (MUAP) waveforms for two representative motor unit sources. MUAP waveforms were identified by taking the spike-triggered average at the spike timings from the five highest weighted channels of the reconstructed EMG signal. During reconstruction, other motor unit sources were suppressed for cleaner MUAP waveforms. **C.** First extension of the inverse spatial filters used to reconstruct EMG signal from motor unit source activity mapped to the HD-EMG grid. Dipoles show weighting of motor unit source.

Across all six participants in both ramp and MVC paradigms, MUelim consistently decomposed motor units with substantially reduced computational time compared to alternative methods (Table 1). For ramp contractions, MUelim achieved up to a 36× speed increase compared to MUEdit (3.1 vs 112.3 minutes, Supplementary Table 2). The motor unit yields observed are consistent with prior reports for high-density grids in hand and forearm muscles [Farina et al., 2008, Negro et al., 2016]. Full timing results from the parameter sweep in a single participant are shown in Supplementary Figure 2. Varying bin size for SPD matrix computation affected the number of sources decomposed, while having a minimal effect on decomposition time (Supplementary Figure 2A-B). The optimal bin size range to extract the most sources was 20-30ms. When comparing SPD matrix estimators, cospectral matrices yielded more sources in the optimal bin size range, while taking less time to decompose than a laplacian kernel estimator.

Table 1: Comparison of motor unit decomposition methods across N=6 healthy participants

Method	**Ext.**	**Lag**	**Ramp Experiment**			**MVC Experiment**		
			MUs	**Time (min.)**	**SIL**	**MUs**	**Time (min.)**	**SIL**
MUelim	2	4	12.8 ± 2.7	1.3 ± 0.1	0.91 ± 0.01	7.8 ± 1.8	0.3 ± 0.0	0.91 ± 0.01
	4	4	18.8 ± 4.2	3.1 ± 0.2	0.92 ± 0.01	11.2 ± 2.2	0.7 ± 0.0	0.91 ± 0.01
SCD	6	1	2.4 ± 0.5	5.5 ± 0.6	0.89 ± 0.01	5.6 ± 1.4	9.2 ± 3.0	0.90 ± 0.02
	16	1	3.2 ± 0.8	14.6 ± 3.6	0.90 ± 0.01	5.6 ± 1.7	5.7 ± 1.1	0.90 ± 0.01
MUEdit	6	1	3.2 ± 2.0	72.3 ± 4.5	0.88 ± 0.01	24.3 ± 9.6	10.9 ± 2.2	0.89 ± 0.00
	16	1	18.2 ± 5.7	112.3 ± 28.5	0.89 ± 0.03	12.5 ± 3.2	30.9 ± 5.5	0.93 ± 0.01

Both extension factor and lag influenced the number of sources decomposed and decomposition time (Supplementary Figure 2C-D). The number of sources decomposed began to saturate with an extension factor greater than 2. A lag between 4-8 samples per extension yielded the most sources. Across all permutations, SIL was greater than 0.85 for every source with a grand average SIL of 0.955 ± 0.001. The maximum decomposition time was just under eight minutes with extension factor of 6 and lag of 8 samples. Using a small extension factor and lag > 1, a majority of sources can still be found in a fast decomposition time. For example, with an extension factor of 2 and lag of 5, there was an average of 15.4 ± 0.98 sources decomposed across the 5 iterations (81% of the peak number of sources). Decomposition time for this setting was only 1.47 ± 0.01 minutes.

3.2 Closed-loop control of a neuroprosthetic by a SCI user

To evaluate whether MUelim could be used to control a neuroprosthetic, we trained a SNN to decode motor intent from a SCI participant's motor unit activity to activate movement-specific FES patterns during task practice therapy. The SCI participant was successfully able to control the neuroprosthetic using the trained decoder (Figure 3 and Supplementary Video 1). The motor unit-based SNN decoder achieved 85% bin-wise accuracy during live control. For comparison, an offline analysis using standard root-mean-square (RMS) features with a neural network (NN) decoder achieved 86% accuracy on the same dataset. All of the intended movements of the 3 tasks were successfully activated by the participant. Discrepancies between manual labels and final FES predictions can be attributed to short delays in transition between movements and movement to rest (Figure 3B - FES predictions above vs. lightly shaded labels on top of the motor unit spike train). Looking at the first transition from rest to hand open to grab the pipe, MUelim detected the initial motor intent, which triggered the hand open FES pattern to activate (Figure 3A). Following this, there was an initial break-in period of the template artifact filter, with subsequent peaks detected after the second FES pulse that oscillated with stimulation frequency. The user retained control of the device and was successfully able to transition between states to allow the FES to assist in completing the task. Waveforms generated from the spike triggered averaging of reconstructed EMG at detected spikes show clean action potentials (Figure 3D), demonstrating the ability of the algorithm to detect spikes in between FES pulses.

A SNN was able to dissociate between motor unit activity during the different movements independent of the task (Figure 3C). The prediction probability calculated from the output neuron firing rate is shown on top of the SNN output spike train. When there was no output neuron firing, the decoder defaulted to a rest prediction. Corresponding snapshots of the tasks are shown below the spike trains (Figure 3E). Refer to Supplementary Video 1 for the full live video demonstrating neuroprosthetic control.

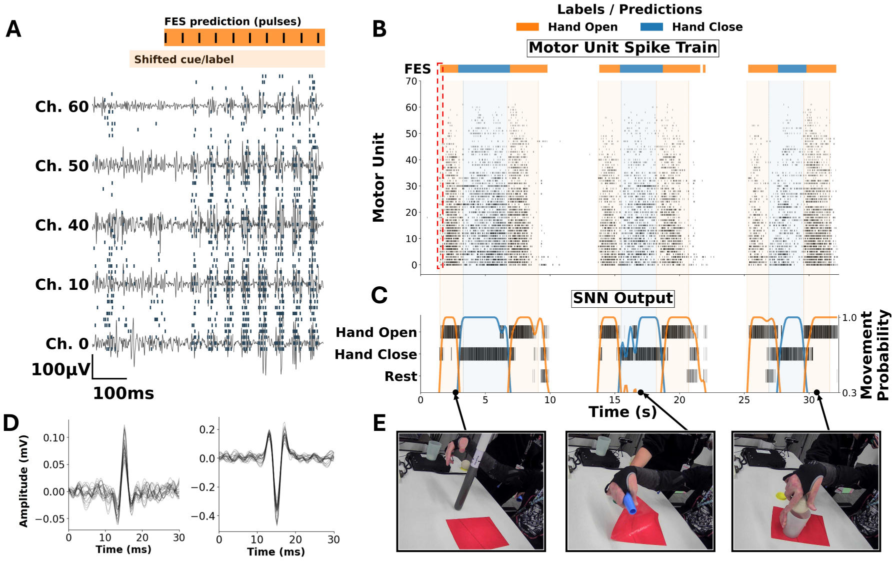

Figure 3: **Spinal cord injury participant controls functional electrical stimulation (FES) in real-time via intention decoded from motor unit activity using a spiking neural network (SNN).** **A.** Zoom-in plot of raw EMG with spike train on top at the transition between rest and hand open. Spikes are detectable in a $50ms$ window between FES pulses with $20Hz$ stimulation frequency. **B.** Full spike train that was decomposed in real-time while using the EMG-FES system in task practice. Rectangles above the spike train indicate the final movement predicted that triggered the corresponding FES movement to evoke assistance. The lighter shaded rectangles across the full spike train indicate the ground truth manually labeled by the operator. **C.** Output of the SNN with 3 classes and the resulting prediction probability based on the output neuron firing rates. **D.** Sample motor unit action potential waveforms averaged across the five highest weighted channels. **E.** Snapshots of SCI participant performing the rehabilitation tasks. See Supplementary Video 1 for a real-time demo.

4 Discussion

We introduced an efficient method for motor unit decomposition that leverages approximate joint diagonalization for rapid extraction of motor units from HD-EMG. Our approach incorporates an extend-lag procedure, previously used in EEG and dynamical systems analysis [Carrara and Papadopoulo, 2024, Takens, 2006], to augment the data and enhance source separability prior to joint diagonalization. While the extend-lag procedure itself is not novel, its application in this context enhances source separability by capturing temporal dynamics and improves the conditioning of the source separation problem. MUelim builds on advances in convolutive BSS and joint diagonalization [Negro et al., 2016, Holobar and Zazula, 2007, Belouchrani et al., 1997, Pham, 2001], and addresses computational bottlenecks that have limited motor unit decomposition for real-time applications [Congedo et al., 2008, Grison et al., 2025].

Comprehensive validation on both simulated and physiological data demonstrated MUelim's effectiveness. Direct comparison on identical simulated datasets showed that MUelim achieved comparable decomposition accuracy to SCD and MUEdit. Validation across six healthy participants in both ramp and MVC contractions demonstrated consistent performance with up to 36× speed improvements compared to open-source methods such as SCD [Grison et al., 2025] and MUEdit [Avrillon et al., 2024], while achieving similar or greater source yield. MUelim's substantially reduced BSS filter computation time (minutes vs up to hours) enables practical deployment in closed-loop neuroprosthetic systems where rapid session setup is required.

We demonstrated this feasibility in a clinical neuroprosthetic control application, in which we decoded motor intent from motor unit firing activity decomposed using MUelim to enable an individual with SCI to volitionally control FES movement patterns. The motor unit-based SNN decoder achieved comparable performance to an offline RMS-NN decoder (85% vs 86% accuracy), although direct comparison is difficult since FES was already triggered during data collection, affecting subsequent motor recruitment and EMG signals. Integrating MUelim with a SNN decoder enables spike-based decoding, which has the potential to be incorporated in neuromorphic hardware for low-power neuroprosthetics [Tanzarella et al., 2023, Chen et al., 2020].

There are several limitations to this work. MUelim was validated only on non-invasive HD-EMG and has not yet been tested on intramuscular EMG, which may present different decomposition challenges [Negro et al., 2016, Holobar et al., 2014]. Due to clinical trial constraints, the clinical evaluation period was limited. As a result, spatial filters may require adaptation for long-term or at-home use, as electrode shifts and signal quality changes can degrade performance over time [Del Vecchio et al., 2020, Farina et al., 2014]. Future work should perform more rigorous benchmarking of motor unit-based features against RMS and other standard EMG preprocessing approaches using open-source datasets, such as those used in EMGBench [Yang et al., 2024]. Broader validation across diverse user populations and movement tasks will also be important to establish generalizability of the approach [Osswald et al., 2025, Wen et al., 2023]. As with any assistive neurotechnology, there is a potential risk that MUelim could lead to unintended device actuation if motor units are incorrectly decomposed, or raise privacy concerns if EMG data are not handled securely. Nevertheless, MUelim provides a practical and efficient solution for real-time motor unit decomposition, supporting the development of robust, intention-driven assistive technologies.

5 Conclusion

This work presents MUelim, an efficient algorithm for motor unit decomposition from HD-EMG signals. By leveraging approximate joint diagonalization, MUelim achieves high decomposition accuracy while significantly reducing computational complexity, outperforming state-of-the-art methods with up to a 36× speed improvement. Comprehensive validation on identical simulated datasets demonstrated that MUelim achieves comparable accuracy to existing methods. Validation across six healthy participants in both ramp and maximum voluntary contraction paradigms demonstrates its ability to extract motor units with high signal-to-noise ratios. Furthermore, we showcased its application in a clinical setting, where MUelim facilitated real-time decoding of motor intent to control a neuroprosthetic device by a SCI participant, highlighting its potential for advancing assistive neurotechnology.

Disclaimer

The system used in the study referenced has not been approved or cleared as safe or effective by FDA. This device is limited by U.S. federal law to investigational use.

Acknowledgments

The authors would like to thank the broader Battelle development, quality, and management teams for their engineering work and general project support. Algorithm development, data analysis, and manuscript drafting were funded through Battelle Memorial Institute internal research and development funds.

The spinal cord injury trial was supported by the Assistant Secretary of Defense for Health Affairs endorsed by the Department of Defense, in the amount of $2,004,786, through the Spinal Cord Injury Research Program under Award No. W81XWH-22-1-1119 and W81XWH-22-1-1083. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the Department of Defense.

References

Pierre Ablin, Jean-François Cardoso, and Alexandre Gramfort. Beyond pham's algorithm for joint diagonalization. *arXiv preprint arXiv:1811.11433*, 2018.

Simon Avrillon, François Hug, Stuart N Baker, Ciara Gibbs, and Dario Farina. Tutorial on muedit: An open-source software for identifying and analysing the discharge timing of motor units from electromyographic signals. *Journal of Electromyography and Kinesiology*, 77:102886, 2024.

Alexandre Barachant, Quentin Barthélemy, Jean-Rémi King, Alexandre Gramfort, Sylvain Chevallier, Pedro L. C. Rodrigues, Emanuele Olivetti, Vladislav Goncharenko, Gabriel Wagner vom Berg, Ghiles Reguig, Arthur Lebeurrier, Erik Bjäreholt, Maria Sayu Yamamoto, Pierre Clisson, Marie-Constance Corsi, Igor Carrara, Apolline Mellot, Bruna Junqueira Lopes, Brent Gaisford, Ammar Mian, Anton Andreev, Gregoire Cattan, and Arthur Lebeurrier. pyriemann, feb 2025. URL `https://doi.org/10.5281/zenodo.593816`. Version v0.8.

Adel Belouchrani and Moeness G Amin. Blind source separation based on time-frequency signal representations. *IEEE transactions on signal processing*, 46(11):2888–2897, 1998.

Adel Belouchrani, Karim Abed-Meraim, J-F Cardoso, and Eric Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444, 1997.

Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

Igor Carrara and Théodore Papadopoulo. Classification of bci-eeg based on the augmented covariance matrix. *IEEE Transactions on Biomedical Engineering*, 2024.

Chen Chen, Yang Yu, Shihan Ma, Xinjun Sheng, Chuang Lin, Dario Farina, and Xiangyang Zhu. Hand gesture recognition based on motor unit spike trains decoded from high-density electromyography. *Biomedical signal processing and control*, 55:101637, 2020.

Maoqi Chen and Ping Zhou. A novel framework based on fastica for high density surface emg decomposition. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(1): 117–127, 2015.

Marco Congedo, Cédric Gouy-Pailler, and Christian Jutten. On the blind source separation of human electroencephalogram by approximate joint diagonalization of second order statistics. *Clinical Neurophysiology*, 119(12):2677–2686, 2008.

Ronald de Vlaming and Eric AW Slob. Joint approximate diagonalization under orthogonality constraints. *arXiv preprint arXiv:2110.03235*, 2021.

A Del Vecchio, A Holobar, D Falla, F Felici, RM Enoka, and D Farina. Tutorial: Analysis of motor unit discharge characteristics from high-density surface emg signals. *Journal of Electromyography and Kinesiology*, 53:102426, 2020.

David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2005.

Dario Farina and Francesco Negro. Common synaptic input to motor neurons, motor unit synchronization, and force control. *Exercise and sport sciences reviews*, 43(1):23–33, 2015.

Dario Farina, Francesco Negro, Marco Gazzoni, and Roger M Enoka. Detecting the unique representation of motor-unit action potentials in the surface electromyogram. *Journal of neurophysiology*, 100(3):1223–1233, 2008.

Dario Farina, Roberto Merletti, and Roger M Enoka. The extraction of neural strategies from the surface emg: an update. *Journal of applied physiology*, 117(11):1215–1230, 2014.

Emanuele Formento, Paul Botros, and Jose M Carmena. Skilled independent control of individual motor units via a non-invasive neuromuscular–machine interface. *Journal of Neural Engineering*, 18(6):066019, 2021.

Agnese Grison, Irene Mendez Guerra, Alexander Kenneth Clarke, Silvia Muceli, Jaime Ibáñez, and Dario Farina. Unlocking the full potential of high-density surface emg: novel non-invasive high-yield motor unit decomposition. *The Journal of Physiology*, 603(8):2281–2300, 2025.

CJ Heckman and Roger M Enoka. Motor unit. *Comprehensive physiology*, 2(4):2629–2682, 2012.

Ales Holobar and Damjan Zazula. Multichannel blind source separation using convolution kernel compensation. *IEEE Transactions on Signal Processing*, 55(9):4487–4496, 2007.

Aleš Holobar, Marco Alessandro Minetto, and Dario Farina. Accurate identification of motor unit discharge patterns from high-density surface emg and validation with a novel signal-based performance metric. *Journal of neural engineering*, 11(1):016008, 2014.

Tamás Kapelner, Ivan Vujaklija, Ning Jiang, Francesco Negro, Oskar C Aszmann, Jose Principe, and Dario Farina. Predicting wrist kinematics from motor unit discharge timings for the control of active prostheses. *Journal of neuroengineering and rehabilitation*, 16:1–11, 2019.

Chuang Lin, Chen Chen, Ziwei Cui, and Xiujuan Zhu. A bi-gru-attention neural network to identify motor units from high-density surface electromyographic signals in real time. *Frontiers in Neuroscience*, 18:1306054, 2024.

Eric C Meyers, David Gabrieli, Nick Tacca, Lauren Wengerd, Michael Darrow, Bryan R Schlink, Ian Baumgart, and David A Friedenberg. Decoding hand and wrist movement intention from chronic stroke survivors with hemiparesis using a user-friendly, wearable emg-based neural interface. *Journal of NeuroEngineering and Rehabilitation*, 21(1):7, 2024.

Francesco Negro, Silvia Muceli, Anna Margherita Castronovo, Ales Holobar, and Dario Farina. Multichannel intramuscular and surface emg decomposition by convolutive blind source separation. *Journal of neural engineering*, 13(2):026027, 2016.

Marius Osswald, Andre L Cakici, Daniela Souza de Oliveira, Dominik I Braun, Dario Farina, and Alessandro Del Vecchio. Task-specific motor units in the extrinsic hand muscles control single-and multi-digit tasks of the human hand. *Journal of Applied Physiology*, 2025.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Dinh Tuan Pham. Joint approximate diagonalization of positive definite hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1136–1152, 2001.

Dinh-Tuan Pham and J-F Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. *IEEE Transactions on signal processing*, 49(9):1837–1848, 2001.

Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

Nicholas Tacca, Collin Dunlap, Sean P Donegan, James O Hardin, Eric Meyers, Michael J Darrow, Samuel Colachis IV, Andrew Gillman, and David A Friedenberg. Wearable high-density emg sleeve for complex hand gesture classification and continuous joint angle estimation. *Scientific Reports*, 14(1):18564, 2024.

Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pages 366–381. Springer, 2006.

Simone Tanzarella, Massimiliano Iacono, Elisa Donati, Dario Farina, and Chiara Bartolozzi. Neuromorphic decoding of spinal motor neuron behaviour during natural hand movements for a new generation of wearable neural interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:3035–3046, 2023.

Yue Wen, Sangjoon J Kim, Simon Avrillon, Jackson T Levine, François Hug, and José L Pons. Toward a generalizable deep cnn for neural drive estimation across muscles and participants. *Journal of Neural Engineering*, 20(1):016006, 2023.

Jehan Yang, Maxwell Soh, Vivianna Lieu, Douglas Weber, and Zackory Erickson. Emgbench: Benchmarking out-of-distribution generalization and adaptation for electromyography. *Advances in Neural Information Processing Systems*, 37:50313–50342, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Introduction sections state the contributions of the work regarding development and validation of the MUelim algorithm, its benchmarking against existing methods, and its application in a clinical neuroprosthetic setting. The claims are consistent with the algorithm description in the Methods section, as well as with the experimental results and discussion.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the discussion section, including the demonstration of the algorithm on both simulated and non-invasive HD-EMG, but not on intramuscular EMG, limited clinical evaluation period, and the need for broader benchmarking and adaptation for long-term use.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not introduce new theoretical results or formal proofs. Instead, it builds on established methods and references existing theoretical work for joint diagonalization and blind source separation.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The manuscript provides detailed descriptions of the experimental setup, including data generation, preprocessing, algorithm parameters, evaluation metrics, and benchmarking procedures. These details are found throughout the Methods and Results sections.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code and data used in this study are not currently publicly available. We are actively considering options to make them accessible in the future, but at this time, open access is not provided.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The Methods section provides detailed descriptions of the experimental setup, including data generation, preprocessing steps, parameter sweeps, evaluation metrics, and SNN training details such as optimizer, learning rate, batch size, and dropout.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Error bars representing the standard error of the mean are reported for the parameter sweep experiments, as described in the Methods and shown in Table 1 and Supplementary Figure 2. Statistical comparisons between methods were not performed because only one seed was used for each alternative method due to practical constraints.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Methods section specifies the hardware used for experiments, including CPU and GPU models, as well as the time required for decomposition in minutes for each method.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research was conducted in accordance with the NeurIPS Code of Ethics, including appropriate handling of human subject data and privacy. The clinical study was performed as part of an ongoing registered clinical trial with Institutional Review Board (IRB) approval and participant consent.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The manuscript discusses the positive societal impact of enabling intention-driven assistive neurotechnology for individuals with motor impairments using the MUelim algorithm, as well as limitations and considerations related to generalizability and long-term use. Potential negative impacts, such as risks associated with incorrect decomposition, are acknowledged in the Discussion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release any data or models that pose a high risk for misuse. As a result, safeguards are not applicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external code packages used in this work are properly cited in the manuscript, including references to open-source tools such as swarm contrastive decomposition (SCD) [Grison et al., 2025], MUEdit [Avrillon et al., 2024], pyRiemann [Barachant et al., 2025], sklearn [Pedregosa et al., 2011], JADOC [de Vlaming and Slob, 2021], and LAVA-SLAYER [Shrestha and Orchard, 2018]. All code packages were used in accordance with their respective licenses and terms of use as specified by their original authors.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: At this time, the paper does not release any new datasets, code, or models as assets, so this question is not applicable. If assets are released in the future, appropriate documentation will be provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The clinical trial involving human subjects was conducted with IRB approval and informed consent was obtained from all participants. Participants were compensated for their time in accordance with ethical guidelines. Full instructions and screenshots are not central to the contribution and are available upon request.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The study involving human subjects was conducted with IRB approval, and all participants provided informed consent after being informed of potential risks. Details regarding the clinical trial and participant safety are described in the Methods and Disclaimer sections, without revealing any information that would compromise anonymity.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: No LLMs were used to generate any of the core methods in this work.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.