A Hybrid Knowledge- and Data-driven Model for Automatic Assessment of Chemically Induced Spiking Patterns in C-fiber Microneurography

Alina Troglio[1,2], Andrea Fiebig[3], Anna Maxion[4], Ekaterina Kutafina[5,6,†], Barbara Namer[2,†]

[1]Institute of Neurophysiology, RWTH Aachen University Hospital, Aachen, Germany
[2]Department of Anesthesiology, Intensive Care, Emergency and Pain Medicine, University Hospital Würzburg, Center for Interdisciplinary Pain Medicine, Würzburg, Germany
[3]Research Group Neuroscience, Interdisciplinary Centre for Clinical Research (IZKF), Faculty of Medicine, RWTH Aachen University, Aachen, Germany
[4]Institute for Computational Biomedicine, RWTH Achen University, Aachen, Germany
[5]Institute for Biomedical Informatics, Faculty of Medicine, University Hospital Cologne, University of Cologne, Cologne, Germany
[6]Scientific Center for Neuropathic Pain Research Aachen, SCN^{AACHEN}, Uniklinik RWTH Aachen University, Aachen, Germany

***Correspondence:**
Ekaterina Kutafina
ekaterina.kutafina@uni-koeln.de

[†]These authors share the last authorship.

Abstract

Analyzing temporal spike patterns in nociceptors recorded via microneurography is challenging due to the use of a single recording electrode, waveform variability, and high similarity of spike shapes across neurons limiting interpretation of sensory coding such as pain and itch. We present a data-driven, supervised spike sorting approach to improve the analysis of nociceptive discharges, identified through activity-dependent conduction velocity changes.

Our method integrates three feature sets and applies machine learning models including one-class SVM, SVM, and XGBoost. Validation used experimentally derived ground truth data acquired by controlled electrical stimulation, allowing precise spike time-locking. Compared to Spike2 software, our approach achieved higher F1-scores and reduced false positives, indicating improved spike sorting. Although XGBoost achieved the highest median F1-scores, optimal performance was dependent on individual combinations of feature sets and models for each recording. In some recordings with many nerve fibers and a low signal-to-noise ratio, reliable sorting was not feasible. This highlights the necessity to determine sortability and optimal configures using a ground truth protocol for each recording.

These findings represent an important step toward reliable analysis of nociceptive activity. The openly accessible framework supports analyzing pruritogen-induced and spontaneous activity in neuropathic pain patients, advancing tools for peripheral neural decoding.

Keywords: microneurography, spike trains, machine learning, pain, nociceptor, C-fibers

Introduction

Sensory perception in the peripheral nervous system depends not only on the presence or absence of action potentials (spikes) but also on the precise temporal patterns in which they are fired [1, 2]. Spike patterns defined by firing regularity, bursting, and adaptation, carry information about stimulus quality and intensity [3]. In the central nervous system, the temporal dynamics of neural activity are known to shape synaptic strength and influence network plasticity [4]. Moreover, spikes in peripheral afferents can modulate the behavior of extraneural cells, such as immune or glial cells, through activity-dependent neuroimmune interactions [5]. These insights suggest that spike timing and patterns in peripheral neurons are not merely epiphenomena but may play a critical role in encoding and modulating sensations, particularly in complex modalities like pain and itch.

Despite this, the spike patterns of peripheral nociceptors and pruriceptors, especially in humans, have received relatively little attention. Most single-electrode recordings of peripheral nerve fibers in human (microneurography) studies have focused on indirect assessment of spike rates or thresholds rather than temporal structure, leaving a major gap in our understanding of how these neurons convey nuanced sensory information [6]. This analysis is especially relevant in pathological conditions such as neuropathic pain, as some primary afferent fibers exhibit spontaneous, continuous, and irregular firing patterns that are not evoked by external stimuli [7]. These chaotic firing patterns may have functional consequences for downstream processing and symptom manifestation. For more general neuroscientific questions of sensory encodings of itch and pain, it must be noted that the same nerve fibers are able to encode itchy and painful heat stimuli [8, 9]. However, without reliable spike pattern analysis, those hypotheses and questions cannot be explored and answered.

Microneurography enables direct extracellular recordings from single afferent fibers in awake humans, offering insights into the dynamics of sensory encoding at the level of individual neurons [10]. However, because single tungsten microelectrodes capture activity from multiple nearby fibers, spike detection and sorting are essential before any meaningful analysis of spike patterns can be performed. This process is particularly difficult in peripheral nerve recordings, especially for unmyelinated nerve fibers, which are also abundantly present in the spinal cord and brain. In these recordings, spike amplitudes are small with typically low signal-to-noise ratios, conduction velocities are activity-dependent, and waveform shapes are very similar between different fibers and variable in a single nerve fiber [11, 12].

While numerous spike sorting algorithms exist, ranging from traditional clustering techniques to automated frameworks such as SpikeInterface [13] and SpikeForest [14], which facilitate comparison and validation across methods, most are optimized for high-density or multi-site electrode arrays [15]. These tools often rely on spatial separation of units, timing effects, or stable waveform shapes across channels, assumptions that do not hold in single-electrode

microneurography. Unsupervised spike sorting methods have shown limited reliability when applied to single-channel microneurography recordings [16].

In this manuscript, we present an open-source spike detection and sorting pipeline for noisy single-electrode recordings in peripheral nerves, focusing on C-fiber nociceptors relevant to pain and itch. Our approach is optimized for recordings with low-amplitude spikes. Building on prior work showing that spikes evoked by regular low-frequency stimuli can train supervised classifiers with different feature sets [16], we introduce two key extensions. First, we designed a specialized stimulation protocol that enhances the traditional marking method [17], establishing reliable ground truth data for all electrically evoked spikes. Second, we developed an improved computational pipeline combining peak detection with supervised classification, utilizing latency and activity-dependent conduction velocity changes to constrain and refine spike detection. We benchmark our methods against Spike2 using ground truth data. To illustrate real-world applicability, we tested our pipeline on a recording with pruritogen injection eliciting chemically induced spikes, despite the absence of ground truth data. By combining experimental ground truth data with supervised classification, this work provides a foundation for decoding peripheral nerve activity and advancing the study of sensory coding.

Methods

Ethical approval

Five healthy female volunteers and one healthy male volunteer participated in the microneurography experiments. The experiments involving humans were reviewed and approved by the Ethics Board of the University Hospital RWTH Aachen with numbers Vo-Nr. EK141-19 and Vo-Nr. EK143-21. The participants were comprehensively informed about the experimental procedure, provided their written informed consent, and the study was conducted according to the Declaration of Helsinki.

Experimental setup

With microneurography, spikes from individual C-fibers within the cutaneous C-fiber fascicles of the superficial peroneal nerve were recorded. A detailed explanation of this technique can be found in [8, 11]. For the experiments, a recording electrode (Frederick-Haer, Bowdoinham, ME, USA) was positioned near the nerve bundle, and innervation territories were mapped using a 0.5-diameter tipped electrode. Once the targeted territories were identified, 0.2 mm diameter needle electrodes (Frederick-Haer) were inserted into the skin for intracutaneous electrical stimulation via a Digitimer DS7 constant current stimulator. Neural signals were amplified and processed in parallel using a Neuro Amp EX (ADInstruments) amplifier, followed by an additional bandpass filter (500-1000 Hz) and a 50 Hz notch filter to minimize electrical noise. The data acquisition was performed with two systems in parallel: 1. with a sampling rate of 10 kHz using an analog-to-digital converter

from National Instrument and the software Dapsys (www.dapsys.net) by Brian Turnquist [12], and 2. Power1401 as a digital analogue converter from Cambridge Electronics Design and the software Spike2 v10.08.

Marking method

To differentiate individual C-fiber responses, low-frequency electrical stimulation was applied to repeatedly activate nerve fibers [17]. These background stimuli delivered at a fixed time (here, every 4 seconds), created a predictable time window for expected spike occurrences due to a constant conduction velocity of C-fibers when stimulated with low-frequencies. The signal is segmented relative to each background stimulus onset, allowing spikes originating from a single C-fiber to align temporally at a consistent latency. This temporal alignment resulted in vertically stacked spikes, referred to as *tracks*. The effect of the marking method is visualized in a raster-like format known as a "waterfall plot" (see Figure 1a).

The resulting tracks enabled reliable identification and sorting with tracking algorithms [12], even for spikes near noise level (see Figure 1b, blue rectangle). When a fiber is activated shortly before an electrical background pulse, its conduction velocity temporarily increases, leading to a delayed response to the subsequent background pulse (see Figure 2). This latency shift, which scales approximately with the number of preceding spikes, is known as activity-dependent conduction velocity slowing (ADS) [18]. ADS serves as a functional marker for individual fibers and allows for semi-quantitative estimation of fiber activity (line 4 in Figure 2A).

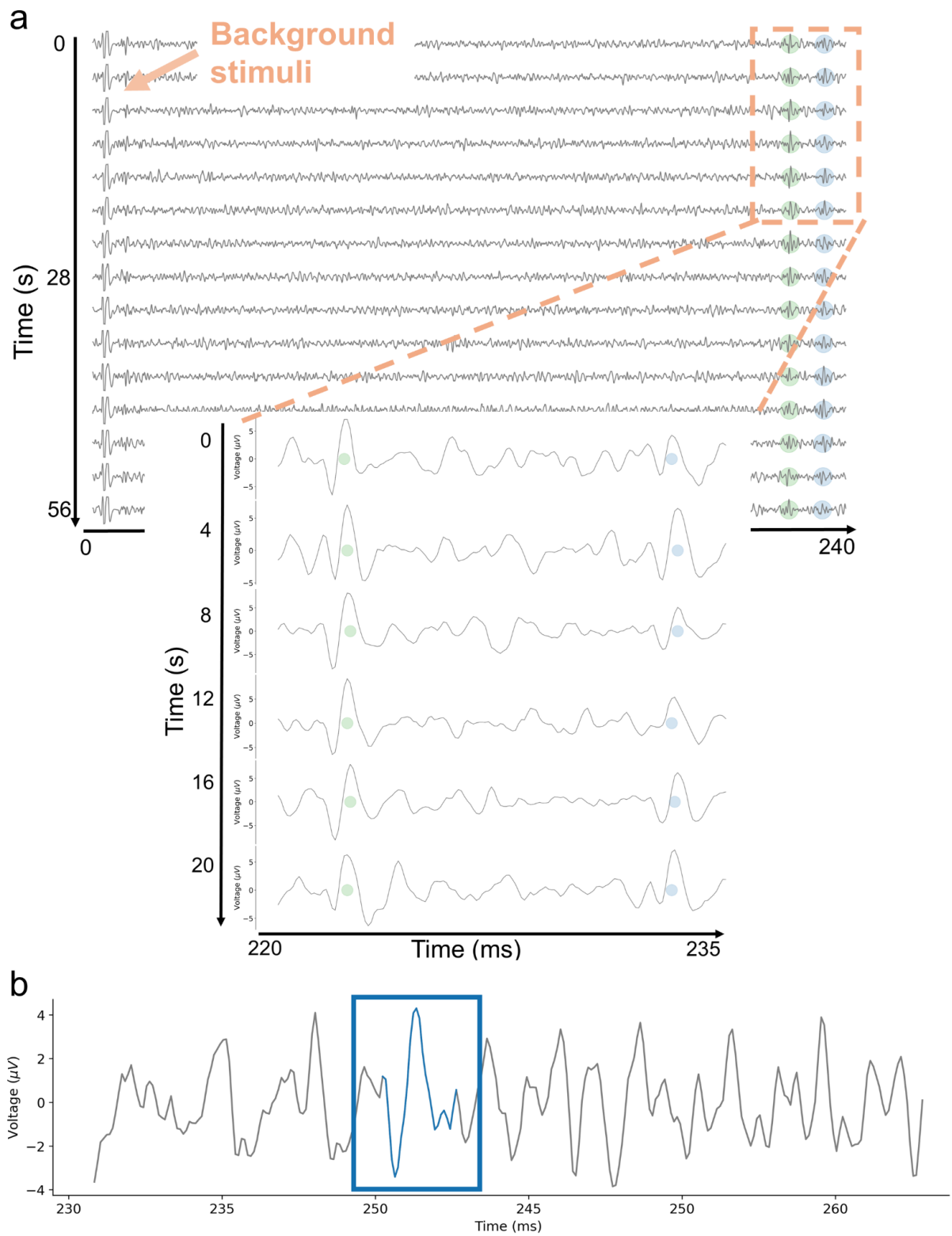


Figure 1. (a) Example waterfall plot of two tracks**.** The signal is segmented every 4 seconds, beginning with the electrical background stimulus. Two tracks are visible, marked by the green and blue dots, indicating the activity of two C-fibers. The conduction velocity stays almost constant leading to vertical alignment of the elicited spikes. **(b)** An example spike at noise level. The spike in the blue box would not be detectable without the vertical alignment of repeated responses enabled by the marking method and waterfall representation. This temporal alignment facilitates the identification of low-amplitude spikes by leveraging consistent latencies evoked by background stimuli.

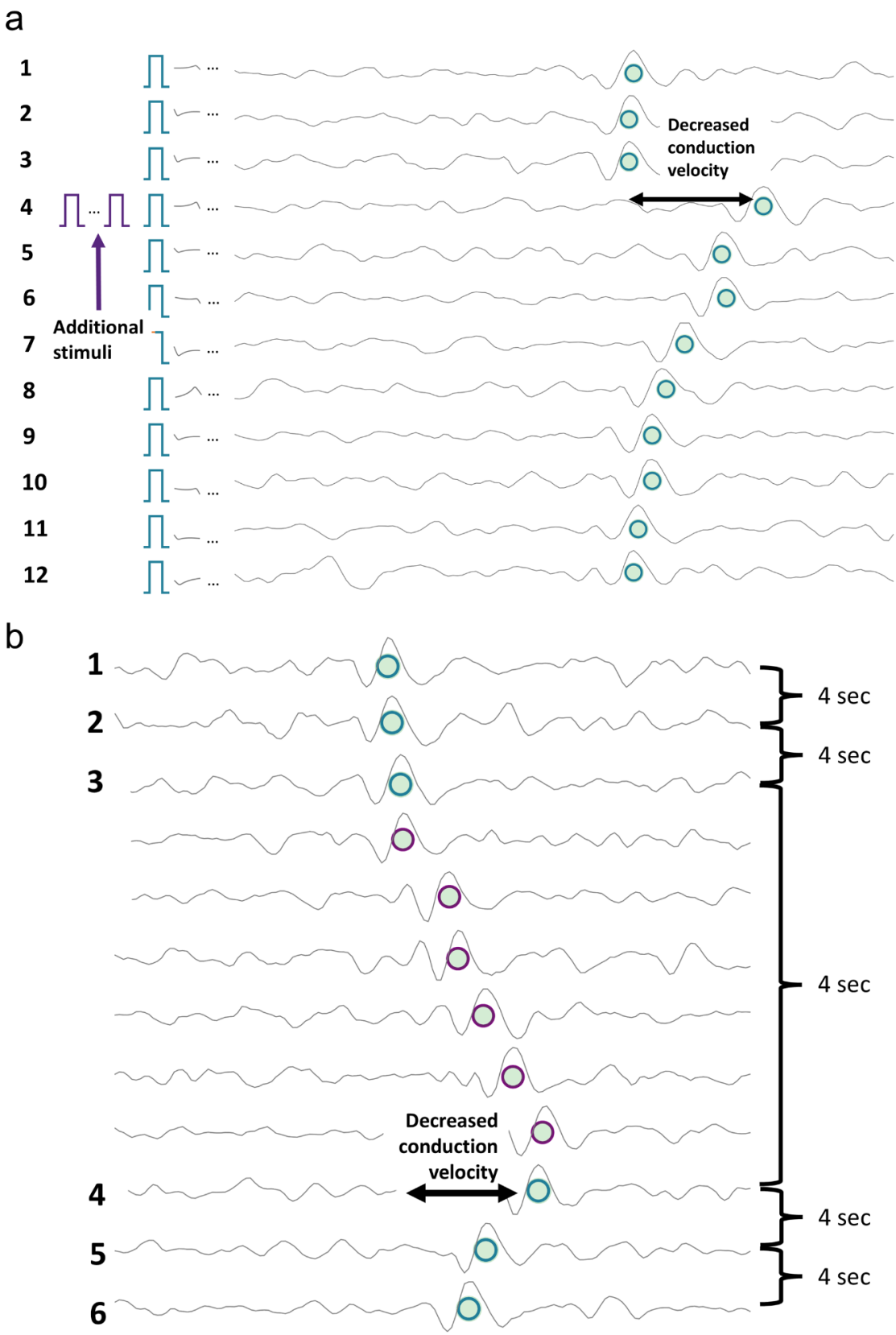


Figure 2. (a) Example of activity-dependent slowing. Each line begins with the periodic background pulse indicated by the blue rectangle, resulting in signal segments with a duration of 4 seconds. When no extra activity is observed (lines 1-3) the spikes (green dots) align vertically due to the almost constant conduction velocity.

When additional (electrical) stimuli are applied, there is a decrease in conduction velocity (line 4). Responses to background stimuli are marked by green dots with a blue circle. **(b)** Example of activity-dependent slowing stimulated with the ground truth protocol. The signal is cut just before each stimulation pulse, whether it is a background pulse or an additional pulse. This ensures that all spikes are aligned on the green track. Responses to background stimuli remain marked by green dots with blue circles, while responses to additional stimuli are indicated by purple rectangles and circles, both for the stimuli and the evoked spikes.

While the marking method enables reliable spike differentiation between fibers based on the tracks, it is inherently constrained to spikes that are time-locked to the background stimuli. Spikes that are elicited from other activity despite the background stimulation, such as those induced by spontaneous or pruritogen-evoked activity, cannot be reliably detected and assigned to fibers. To address these constraints, we implemented an extended version of the marking method, incorporating a modified electrical stimulation protocol designed to capture both background-evoked and non-time-locked spikes.

“Ground truth” stimulation protocol for model development and validation

To allow the detection and sorting of all electrically evoked spikes, we implemented a modified electrical stimulation with standard background pulses every 4 seconds and additional stimuli in between two background stimuli designed to mimic fiber activity and to induce activity-dependent slowing. In previous protocols, such conditioning pulses are not designated to time-lock spikes, and their effects are only indirectly observable through changes in conduction velocity in response to subsequent background stimuli (see Figure 2A, line 4). This limitation restricts tracking capabilities since only spikes directly elicited by background stimuli temporally align on tracks in the waterfall plot.

In our "ground truth” protocol, all stimuli, inducing the additional stimuli, are explicitly defined as background stimuli, allowing evoked spikes to align within existing tracks. The segmentation windows in the waterfall representation are dynamically adjusted to create the intervals between two stimuli, rather than being at fixed 4-second intervals (see Figure 2B). The resulting recordings provide verified timestamps and constituent temporal alignment for each spike, establishing a reliable ground truth timestamp and label for spike detection and sorting, and hence algorithm validation in noisy single-electrode data.

Recording with pruritogen injection

To evaluate the practical applicability of our computational approach to spike detection and sorting, we included one representative recording in which electrical background stimulation was combined with intracutaneous injection of 50 µL bovine adrenal medulla peptide 8-22 (BAM 8-22, Cat. No. SML0729, Sigma, Taufkirchen, Germany), as previously explained in [8]. Unlike the ground truth datasets, this recording did not include labels for spikes elicited by additional activity, making it a realistic illustration of the challenges posed by the detection and sorting of chemically induced spikes.

Ground truth labeling with Dapsys via manual spike sorting

Spike identification and sorting for background stimulation-evoked activity were performed using the integrated tracking algorithm in Dapsys [19] followed by human expert corrections. In the ground truth recordings, this approach enabled the detection and sorting of spikes elicited by both background and additional stimuli. In contrast, for the recording with chemical stimulation, only spikes time-locked to background stimuli could be reliably identified with Dapsys.

A knowledge- and data-driven spike sorting pipeline

We developed a Snakemake-based [20] computational pipeline for microneurography recordings (see Figure 3), integrating knowledge-driven steps (for example, background spike alignment and constraining the detection space) with data-driven machine learning for spike detection and classification.

Data pre-processing

All recordings were read in and parsed into structured pandas DataFrames (v2.2.3) containing the raw signal, spike labels and timestamps, and simulation employing our custom Python package PyDapsys [21].

Spike waveforms were extracted from the raw signal with a fixed window of 30 data points, corresponding to a 3 ms interval at a sampling frequency of 10,000 Hz. Waveform alignment was performed according to the method described by Caro-Martín et al. [22], where each spike is temporally centered based on the time point corresponding to the largest downward slope (i.e., the negative peak in the first derivative of the signal). This alignment step ensures that extracted spike features are computed consistently across spikes.

Template-based identification of optimal units

For each track, a template was computed by averaging all spikes aligned on that track. This template serves both as a visual representation of spike morphology and as a quantitative measure of spike amplitude, aiding in the identification of high-amplitude tracks where thresholding-based detection is more reliable. The track with the highest amplitude template was selected as the track of interest for subsequent spike sorting.

To evaluate the general quality, we computed the signal-to-noise ratio (SNR) for the track of interest. A noise segment of 40 ms was extracted from the raw signal preceding the first evoked spike to ensure the absence of fiber activity. The SNR was then calculated as the ratio of the spike template amplitude to the noise standard deviation, estimated using the median absolute deviation (MAD), following the implementation in SpikeInterface [13].

Spikes were further categorized by their consecutive temporal differences. Background-evoked spikes were defined as those occurring at least 3.8 seconds apart, consistent with the 4 second-interval of the stimuli. In contrast, spikes with shorter inter-spike intervals were classified as responses to additional stimulation. Notably, background-evoked spikes are

consistently available in recordings obtained via the marking method, whereas spikes evoked by additional stimuli are exclusive to datasets acquired using the ground truth stimulation protocol.

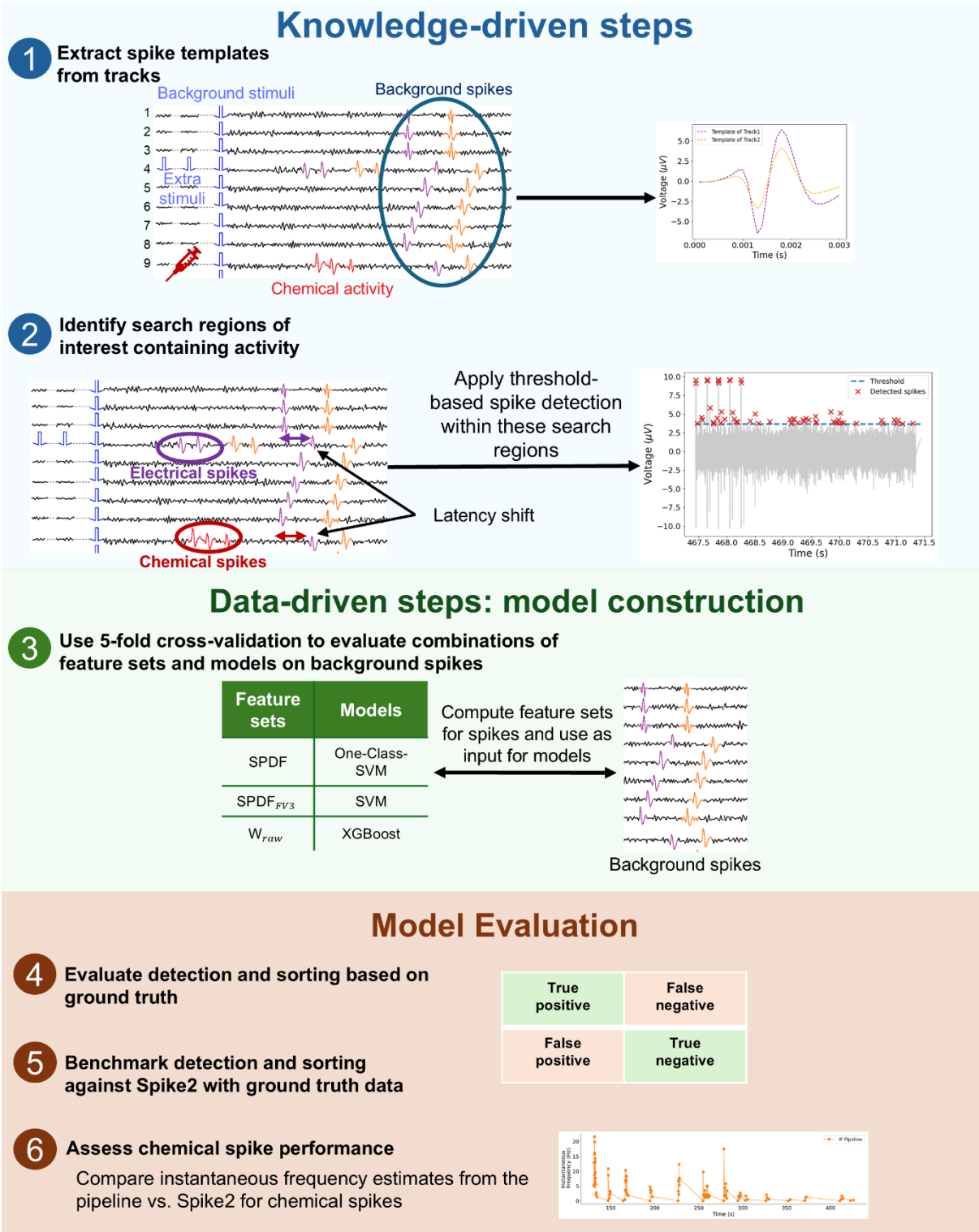


Figure 3. Overview of a hybrid spike detection and evaluation pipeline combining knowledge- and data-driven steps. (1) Spike templates are extracted from background spikes (blue circle) using low-frequency stimulation to identify the track of interest for further analysis. (2) Search spaces containing activity are identified, and spikes are detected using threshold-based methods. (3) Feature sets are computed from background spikes and used as

input to different classification models (One-Class SVM, SVM, XGBoost). Model performance is evaluated through 5-fold cross-validation to determine the optimal model-feature set combination for the background spikes. (4) Evaluate detection and sorting based on ground truth. First, spike detection is assessed by comparing detected spikes to known ground-truth spikes, classifying outcomes as true positives, false positives, false negatives, or true negatives. Next, sorting performance is evaluated by verifying whether detected spikes are correctly assigned to their originating fibers. (5) Benchmark detection and sorting against Spike2 with ground truth data. The pipeline's detection and sorting performance is directly compared to Spike2 by applying both methods independently to the same recording segments. (6) Assess chemical spikes performance. For recordings with the pruritogen injection, we compute the instantaneous frequency and latency of chemically evoked spikes sorted by both the pipeline and Spike2. Since no ground truth is available for these spikes, we compare the resulting patterns to assess how plausible and physiologically consistent the detected activity is across methods.

Spike detection with constrained search space

Spike detection thresholds were determined by manual analysis of the peak amplitudes of background-evoked spikes for the track of interest. To establish these thresholds, the peak amplitudes were first sorted in ascending order to facilitate the identification of the minimum threshold at which spikes occur, and the resulting distribution was examined for variance and outliers. In recordings exhibiting high variance, we manually selected the threshold, otherwise the smallest amplitude value was selected. The final thresholds used are reported in Table 2. We intentionally decided not to use fully automated thresholding algorithms, as such methods are often sensitive to fluctuations and noise. Manual threshold selection allowed careful tailoring to the specific signal characteristics of each recording, thereby reducing the rate of false-positive spike detections.

To constrain the search space for threshold-based spike detection, we calculated spike latencies relative to the background stimuli for the track of interest. Latency shifts larger than 0.9 ms were used to identify time windows from all 4-second signal segments. These latency shifts indicated spikes evoked by additional stimuli and provided knowledge, which constrained the search space for detection.

Spike detection was then performed using the find_peaks function from SciPy (v1.15.2), applying the manually determined thresholds. To exclude stimulation artifacts, we excluded any detected peaks occurring within 100 ms of the stimulation onset.

Spike classification using machine learning

To assign detected spikes to their corresponding track, we tested three machine learning classifiers: a one-class support vector machine (One-Class-SVM) and a standard support vector classifier (SVM), implemented in scikit-learn (v1.6.1) [23], and eXtreme Gradient Boosting classifier (XGBoost, v3.0.2) [24]. Feature vectors were extracted from the 30 datapoints describing the raw waveform. Three feature sets were computed: SPDF, a 23-dim feature vector, adjusted and implemented from the SS-SPDF features presented by Caro-Martín et al. [22], $\mathrm{SPDF}_{\mathrm{FV3}}$, a reduced three-dimensional subset of SPDF, and $\mathrm{W}_{\mathrm{raw}}$, the raw waveform feature vector, the 30 datapoints extracted from the raw signal. Detailed definitions of these feature sets are described in our other paper currently under review [16].

For the SVM and XGBoost classifiers, models were trained on background-evoked spikes from all available tracks in a recording, enabling multi-class discrimination across fibers. The One-Class-SVM model was trained using only background spikes from the track of interest, aiming to identify whether newly detected spikes match the known distribution of that track.

Hyperparameter optimization for all models was performed using the Optuna framework (v 4.3.0) [25], applying 5-fold cross-validation on the background spikes. Maximizing the mean F1-score across folds was used as the optimization objective.

Validation and evaluation methods

Detection and sorting performance were evaluated independently. For both, performance metrics included true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) rates, from which F1-score, precision, and recall were derived.

In the case of detection, we additionally report the false positive per true positive (FP/TP) rate as an additional indicator for reliability. Detection performance was assessed by comparing the timestamps of detected spikes against ground truth spike annotations using a temporal tolerance window of 2 ms.

For sorting, the final models were retrained on all background spikes using the previously selected optimal hyperparameters. Each model was then applied to the set of detected spikes, after extracting each of the three feature sets from those spikes to serve as input. Classifier performance was assessed with spikes evoked by extra electrical stimuli, which were excluded from training and served as ground truth.

Statistical Analysis

Data was collected from multiple datasets and different performance metrics (F1-score, precision, recall, TP, FP, TN, and FN) were calculated. The metrics F1-score, precision, and recall are bounded in [0, 1]. As such proportions typically deviate from normality, we considered models suitable for bounded data. The metrics TP, FP, TN, and FN are count data and we therefore applied separate modeling strategies. All statistical analysis was done in R.

Data Inspection and Preprocessing

We first conducted exploratory analyses (histograms, boxplots) to examine the overall distribution of each metric and identify any potential outliers or anomalies. The bounded metrics often approached boundary values (0 or 1). To avoid issues with values exactly at the boundary, each 0 or 1 was shifted slightly ($\varepsilon = 0.0001$) into the interval (0, 1).

Comparing Statistical Models

For the bounded metrics, we compared two different statistical models: a Linear mixed model with logit transformation and a Beta mixed model, explicitly suitable for [0,1] data, fitted

with a logit link. The model formula was expressed by $\text{OutcomeMetric} \sim \text{Model} * \text{FeatureSet} + (1 \mid \text{Dataset})$.

For the count data, we compared a Poisson mixed model with a log link with a negative Binomial model. To test for overdispersion, we calculated the Pearson residuals and their dispersion ratio (sum of squared Pearson residuals divided by the residual degrees of freedom). The model formula was expressed by $\text{OutcomeMetric} \sim \text{Model} * \text{FeatureSet} + \text{offset}(\log(\text{RecordingDuration})) + (1 \mid \text{Dataset})$.

To compare the different models for the bounded as well as for the count data, we compared the information criteria (AIC/BIC) and conducted an analysis of the plots of the residuals compared to the fitted values.

Baseline comparison with Spike2

To create a baseline for our algorithm, spike detection and sorting were also performed with Spike2 (v10.08), which implements a template-matching approach. The "New WaveMark" function was used with default parameters to define a reference spike, based on negative peak thresholds. This reference spike corresponds to a response evoked by a background stimulus and was temporally identified by the latency obtained from Dapsys, ensuring accurate association with a selected track. The identified waveform served as the initial template for spike sorting. Spike2 applied a template to detect and sort all subsequent spikes, the data was exported as spreadsheets and further processed in pandas DataFrames.

To remove stimulation artifacts from the Spike2 result, spikes occurring within a 10 ms window following each stimulus pulse (stimulus onset + 10 ms) were excluded. Simultaneously recorded datasets from Dapsys were used to obtain ground truth labels. A key challenge was the temporal misalignment between the two systems: Dapsys recording sessions were segmented across multiple files, while Spike2 generated a single recording file. A dynamic offset correction was applied to align Spike2 spike timestamps with their corresponding Dapsys timestamps. This offset was computed individually for each stimulus accounting for time drift between systems.

After alignment, spike sorting performance was assessed by comparing Spike2-detected spikes against Dapsys ground truth labels. A Spike2 spike was considered a true positive if it occurred within ± 3 ms of a Dapsys spike. This tolerance window corrected inherited differences in timestamp definitions: Spike2 timestamps correspond to the spike onset, whereas Dapsys timestamp approximate the midpoint of the spike. This evaluation enabled the assessment of Spike2's template-based sorting performance relative to the manually labeled Dapsys result.

Validation for chemically evoked spikes in the absence of direct ground truth

For comparing the spike sorting of Spike2 and our computational pipeline for the recording with pruritogen injection, we assessed the proportion of correctly classified spikes and

calculated the instantaneous firing frequency (IF), defined as the inverse of the interspike interval between consecutive spikes. This firing served as an indirect measure of fiber activity evoked by chemical stimulation, enabling us to assess consistency between the two sorting approaches. To further validate the spike sorting quality, we visualized the latency relative to the background stimuli as a physiological marker of fiber activation. Latency shifts were evaluated alongside instantaneous frequency to detect inconsistencies and assess the plausibility of detected bursts in both methods.

Software Accessibility

The complete spike sorting pipeline, including data preprocessing, feature extraction, and classification, is openly available at https://github.com/Digital-C-Fiber/SpikeSortingForSpikingPatterns. The repository includes source code, documentation, and an example recording.

Results

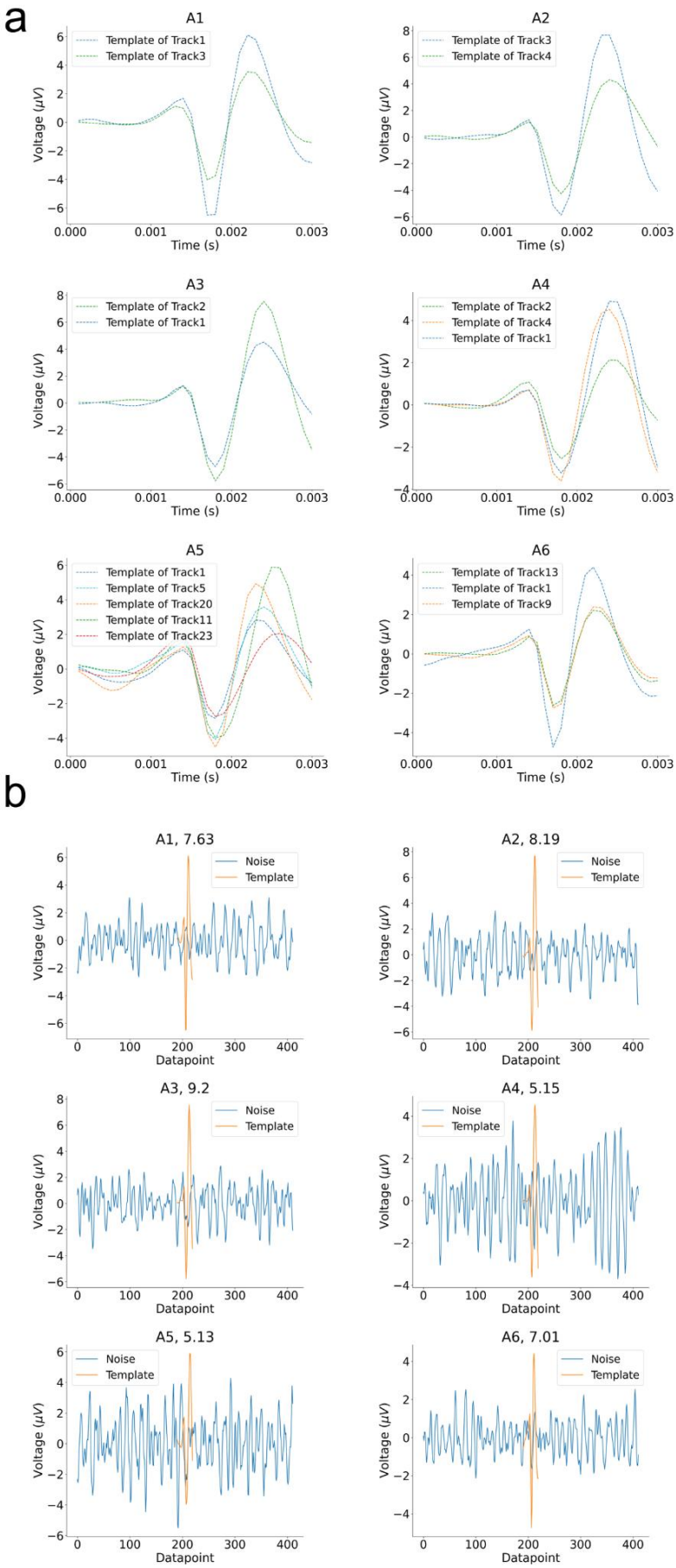
Data characteristics

In this work, we systematically evaluated spike sorting performance across six microneurography recordings, including challenging scenarios with low signal-to-noise ratios (SNR) and a high number of nerve fibers. The recording durations range from 375 to 1,774 seconds with SNR values between 5.13 and 9.20, and the number of tracks varying from 2 to 5.

Table 1 provides a comprehensive summary of the recording characteristics. By selecting these diverse recordings, our analysis provides a robust test of spike sorting methodologies under varied and demanding conditions. To facilitate the identification of a track of interest, we computed a template representing the average of all tracked spikes for each recording. Figure 4a displays these templates. Additionally, to provide an overall impression of the signal quality and recording conditions, we computed the SNR for each dataset (see Table 1). A comparison of the computed template with a short segment of background noise is illustrated in Figure 4b, highlighting the differences between signal and noise characteristics. This visualization offers an immediate qualitative impression of the recording challenges encountered in each dataset.

Table 1. Data overview. The table summarizes the key properties of each dataset, including the total recording duration, the number of tracks (i.e., distinct recorded nerve fibers), and the total number of extracted spikes. For each track, individual spike counts are listed. Additionally, we specify the track selected for detailed analysis, and the signal-to-noise ratio (SNR) computed for the track of interest.

Dataset	Recording duration in sec	Number of Tracks	Total number of spikes	Track labels and distribution	Track label of interest	SNR	Threshold
A1	375	2	293	Track1: 148 Track3: 145	Track1	9.51	3.60
A2	923	2	597	Track3: 299 Track4: 298	Track3	6.82	4.64
A3	581	2	346	Track1: 173 Track2: 173	Track2	5.54	5.12
A4	1774	3	1780	Track1: 585 Track2: 570 Track4: 625	Track4	3.06	3.50
A5	1390	5	2167	Track1: 436 Track5: 437 Track20: 437 Track11: 436 Track23: 421	Track11	6.77	5.00
A6	740	3	796	Track1: 288 Track9: 220 Track13: 288	Track1	5.54	3.00


a
A1
Template of Track1
Template of Track3
A2
Template of Track3
Template of Track4
A3
Template of Track2
Template of Track1
A4
Template of Track2
Template of Track4
Template of Track1
A5
Template of Track1
Template of Track5
Template of Track20
Template of Track11
Template of Track23
A6
Template of Track13
Template of Track1
Template of Track9
Voltage (μV)
Time (s)
b
A1, 7.63
A2, 8.19
A3, 9.2
A4, 5.15
A5, 5.13
A6, 7.01
Noise
Template
Voltage (μV)
Datapoint

381

Figure 4. (a) Average spike templates for each track in all recordings**.** For each dataset, we extracted spike waveforms from each track and aligned them according to the time point corresponding to the largest downward slope (time point of the maximum negative peak in the first derivative of the signal as the temporal midpoint) [22]. We averaged all spikes to generate the templates. Each subplot shows the templates for all tracks within a single recording. These templates reflect the characteristic spike shapes, enable the visual comparison of waveform morphology, and support the selection for the track of interest. **(b)** Signal-to-noise (SNR) visualization for each recording and track of interest. For each recording, the track containing the spikes with the largest amplitudes was used to calculate the SNR. We calculated the SNR by dividing the peak amplitude of the template by the standard deviation of a noise segment extracted from the raw recording. The noise standard deviation was estimated using the median absolute deviation (MAD). The computed SNR for each dataset is indicated next to its name in the figure title.

A computational pipeline for adaptive spike detection and sorting

Constraining the search space with latency information

To evaluate the detection rate of our pipeline, we employed a thresholding approach to identify spikes in the recordings. The detection performance was analyzed for each individual recording. We constrained the search space, focusing on segments preceding a latency jump to improve detection results.

The absolute numbers of detected spikes are visualized in Figure 5. The false positive rate per true positive (FP/TP) varies significantly across datasets. A4 and A5 exhibit the highest values for FP per TP with 20 and 14, respectively. Our approach demonstrated a high recall rate, suggesting that by separating the spike detection and sorting steps, we can first reliably detect spikes before proceeding to classification.

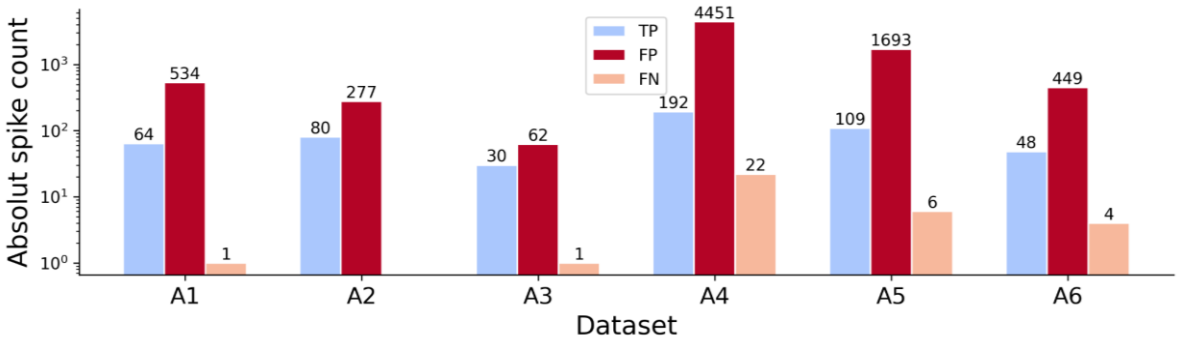


Figure 5. Spike detection performance using thresholding and peak detection. TP, FP, and TN are visualized on a log scale for the detected spikes in the constrained search space. All recordings have higher false positive values compared to the TP values. A4 exhibits the most false positives and 22 spikes could not be detected with the set threshold.

However, in certain cases, increasing the detection threshold proved beneficial and slightly led to a reduction in false positives, at the cost of tolerating a few more false negatives, for example, in the case of A4. This balance is crucial for optimal spike sorting performance. All detection thresholds and computed metrics are listed in Table 2. All metrics are collected in Supplementary Table S1.

Table 2. Spike detection performance. For each recording, the detection threshold for the Dapsys recordings, F1-scores, precision, and recall are reported, along with the number of FP/TP. We set the thresholds based on the amplitudes of the spike templates and optimized the balance between reducing the number of false negatives and false positives.

Dataset	Detection Threshold	F1-Score	Precision	Recall	Number of FP per TP
A1	3.60	0.19	0.11	0.98	8
A2	4.64	0.37	0.22	1.00	3
A3	5.12	0.49	0.33	0.97	2
A4	3.50	0.08	0.04	0.90	23
A5	5.00	0.11	0.06	0.95	16
A6	3.00	0.17	0.10	0.90	9

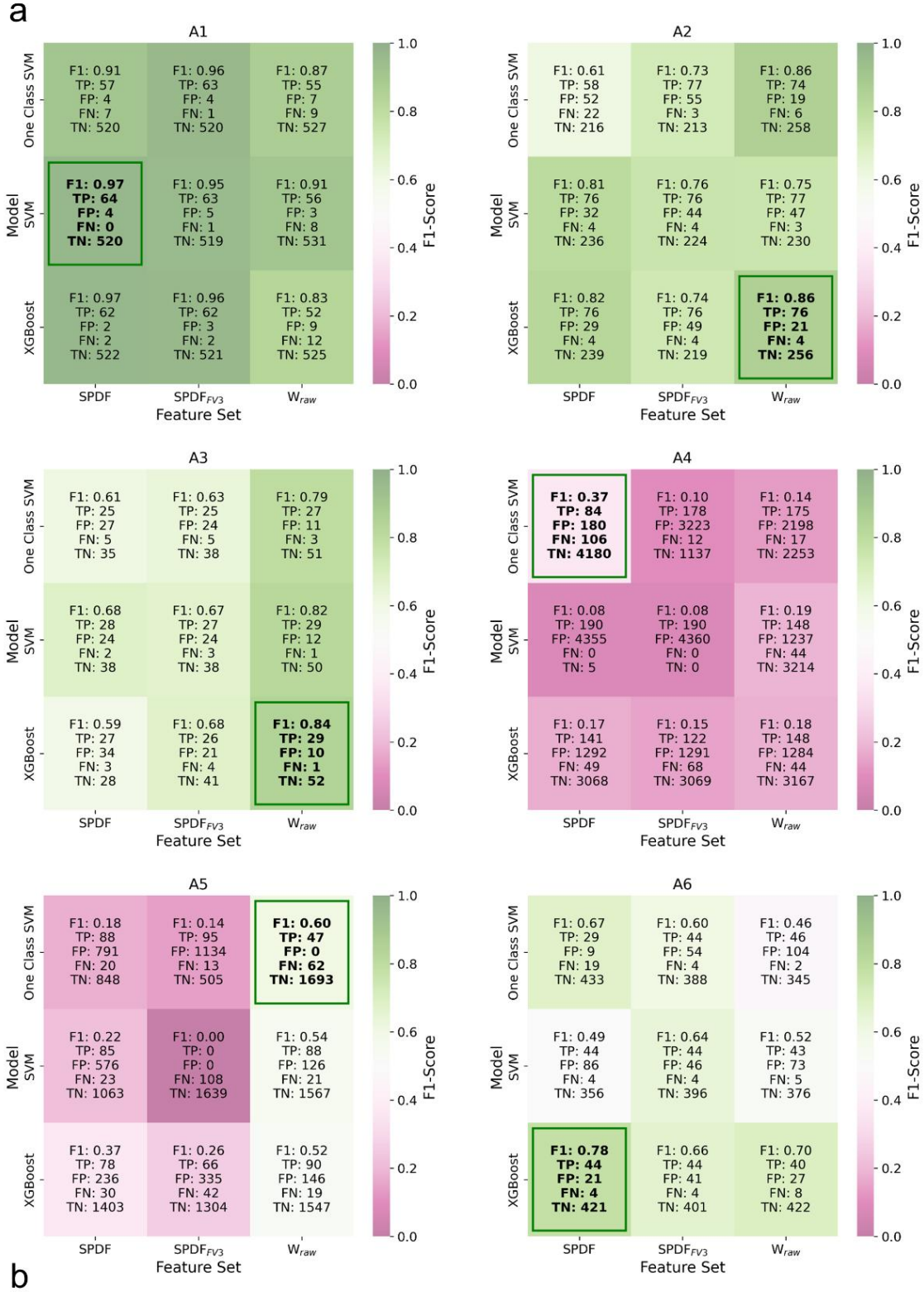
Individual recordings require individual combinations of feature sets and classifying algorithms

After the initial detection step, we employed different input feature sets (SPDF, $\mathrm{SPDF}_{\mathrm{FV3}}$, and $\mathrm{W}_{\mathrm{raw}}$) and classification models (One-Class-SVM, SVM, XGBoost) for spike sorting. A comprehensive comparison of all tested approaches is presented in Figure 6a, which shows the values of F1-score, TP, FP, TN, and FN for each recording. The rectangle highlights the best-performing method for each dataset. The results clearly demonstrate that no single sorting approach yields universally optimal performance across all recordings. Instead, each dataset exhibits its own optimal configuration, underscoring the variability in recording conditions and variance for microneurography. Among the datasets, A1 consistently produced the highest overall performances across methods, while A4 presented the most challenging conditions, with generally lower F1-scores and more frequent misclassifications.

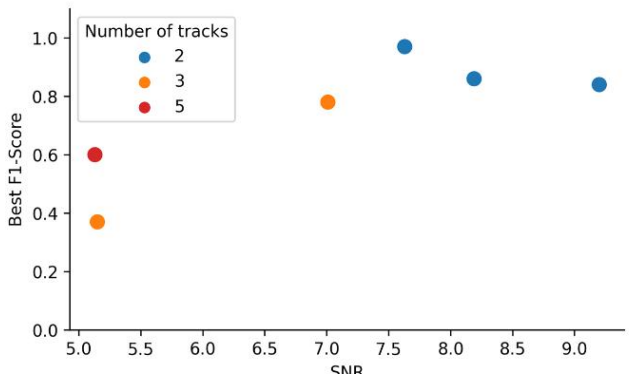
Similar to our findings in the detection phase, the sorting results show strong dataset dependency, with performance varying considerably across different recording conditions and microneurography signal characteristics. To investigate how recording properties influence sorting performance, Figure 6b presents the relationship between SNR and best F1-score. As expected, recordings with lower SNR exhibit lower F1-scores, demonstrating a clear dependency between noise levels and classification performance. However, beyond an SNR of approximately 7.5, the F1-score no longer improves.

Recordings with fewer fibers generally yield higher F1-scores with all recordings containing a small number of fibers achieving scores above 0.8. Interestingly, the dataset with the highest number of fibers (A5) does not produce the worst result.

a


A1
One Class SVM
SVM
XGBoost
Model
F1: 0.91 TP: 57 FP: 4 FN: 7 TN: 520
F1: 0.96 TP: 63 FP: 4 FN: 1 TN: 520
F1: 0.87 TP: 55 FP: 7 FN: 9 TN: 527
F1: 0.97 TP: 64 FP: 4 FN: 0 TN: 520
F1: 0.95 TP: 63 FP: 5 FN: 1 TN: 519
F1: 0.91 TP: 56 FP: 3 FN: 8 TN: 531
F1: 0.97 TP: 62 FP: 2 FN: 2 TN: 522
F1: 0.96 TP: 62 FP: 3 FN: 2 TN: 521
F1: 0.83 TP: 52 FP: 9 FN: 12 TN: 525
SPDF
SPDF$_{FV3}$
W$_{raw}$
Feature Set
F1-Score
A2
F1: 0.61 TP: 58 FP: 52 FN: 22 TN: 216
F1: 0.73 TP: 77 FP: 55 FN: 3 TN: 213
F1: 0.86 TP: 74 FP: 19 FN: 6 TN: 258
F1: 0.81 TP: 76 FP: 32 FN: 4 TN: 236
F1: 0.76 TP: 76 FP: 44 FN: 4 TN: 224
F1: 0.75 TP: 77 FP: 47 FN: 3 TN: 230
F1: 0.82 TP: 76 FP: 29 FN: 4 TN: 239
F1: 0.74 TP: 76 FP: 49 FN: 4 TN: 219
F1: 0.86 TP: 76 FP: 21 FN: 4 TN: 256
A3
F1: 0.61 TP: 25 FP: 27 FN: 5 TN: 35
F1: 0.63 TP: 25 FP: 24 FN: 5 TN: 38
F1: 0.79 TP: 27 FP: 11 FN: 3 TN: 51
F1: 0.68 TP: 28 FP: 24 FN: 2 TN: 38
F1: 0.67 TP: 27 FP: 24 FN: 3 TN: 38
F1: 0.82 TP: 29 FP: 12 FN: 1 TN: 50
F1: 0.59 TP: 27 FP: 34 FN: 3 TN: 28
F1: 0.68 TP: 26 FP: 21 FN: 4 TN: 41
F1: 0.84 TP: 29 FP: 10 FN: 1 TN: 52
A4
F1: 0.37 TP: 84 FP: 180 FN: 106 TN: 4180
F1: 0.10 TP: 178 FP: 3223 FN: 12 TN: 1137
F1: 0.14 TP: 175 FP: 2198 FN: 17 TN: 2253
F1: 0.08 TP: 190 FP: 4355 FN: 0 TN: 5
F1: 0.08 TP: 190 FP: 4360 FN: 0 TN: 0
F1: 0.19 TP: 148 FP: 1237 FN: 44 TN: 3214
F1: 0.17 TP: 141 FP: 1292 FN: 49 TN: 3068
F1: 0.15 TP: 122 FP: 1291 FN: 68 TN: 3069
F1: 0.18 TP: 148 FP: 1284 FN: 44 TN: 3167
A5
F1: 0.18 TP: 88 FP: 791 FN: 20 TN: 848
F1: 0.14 TP: 95 FP: 1134 FN: 13 TN: 505
F1: 0.60 TP: 47 FP: 0 FN: 62 TN: 1693
F1: 0.22 TP: 85 FP: 576 FN: 23 TN: 1063
F1: 0.00 TP: 0 FP: 0 FN: 108 TN: 1639
F1: 0.54 TP: 88 FP: 126 FN: 21 TN: 1567
F1: 0.37 TP: 78 FP: 236 FN: 30 TN: 1403
F1: 0.26 TP: 66 FP: 335 FN: 42 TN: 1304
F1: 0.52 TP: 90 FP: 146 FN: 19 TN: 1547
A6
F1: 0.67 TP: 29 FP: 9 FN: 19 TN: 433
F1: 0.60 TP: 44 FP: 54 FN: 4 TN: 388
F1: 0.46 TP: 46 FP: 104 FN: 2 TN: 345
F1: 0.49 TP: 44 FP: 86 FN: 4 TN: 356
F1: 0.64 TP: 44 FP: 46 FN: 4 TN: 396
F1: 0.52 TP: 43 FP: 73 FN: 5 TN: 376
F1: 0.78 TP: 44 FP: 21 FN: 4 TN: 421
F1: 0.66 TP: 44 FP: 41 FN: 4 TN: 401
F1: 0.70 TP: 40 FP: 27 FN: 8 TN: 422

b


Number of tracks
2
3
5
Best F1-Score
SNR

442

Figure 6. (a) Heatmaps of spike sorting performance across feature sets, classification models, and evaluation metrics. Each cell is color-coded according to the F1-score with magenta indicating low performance and green indicating high performance. The heatmaps display results for all tested combinations of three feature sets and classification models, evaluated using multiple metrics (F1-score, TP, FP, FN, and TN). A rectangle highlights the best-performing configurations based on the F1-score for each recording. **(b)** Relationship between SNR and best F1-score for recordings with varying number of tracks. Each dot represents a recording colored by the number of simultaneously recorded tracks (2, 3, or 5). As expected, lower SNR values correspond to reduced F1-scores. Notably, the F1-score plateaus beyond an SNR of approximately 7.5, indicating no further improvement in performance.

These findings suggest improving the signal quality and reducing the number of active fibers in an active recording session are crucial for good sorting results, which should be considered during the experiments. Additionally, while SNR is a helpful indicator, it should not be used in isolation to predict spike sorting quality as the recording with the highest computed SNR does not have the best sorting results.

Combo XGBoost + W_{raw} achieves the highest median F1-score

To evaluate the performance across different combinations of feature sets and models, we visualized the F1-score using boxplots (see Figure 7a) and a scatter plot (see Figure 7b). All metrics are collected in Supplementary Table S2. For each feature set, XGBoost (orange) consistently achieved the highest median F1-score and demonstrated lower variability compared to SVM (green) and One-Class-SVM (blue). Among the feature sets, the raw waveform features (W_{raw}) yielded slightly higher and more consistent F1-scores than both the full SPDF set and its reduced subset $SPDF_{FV3}$.

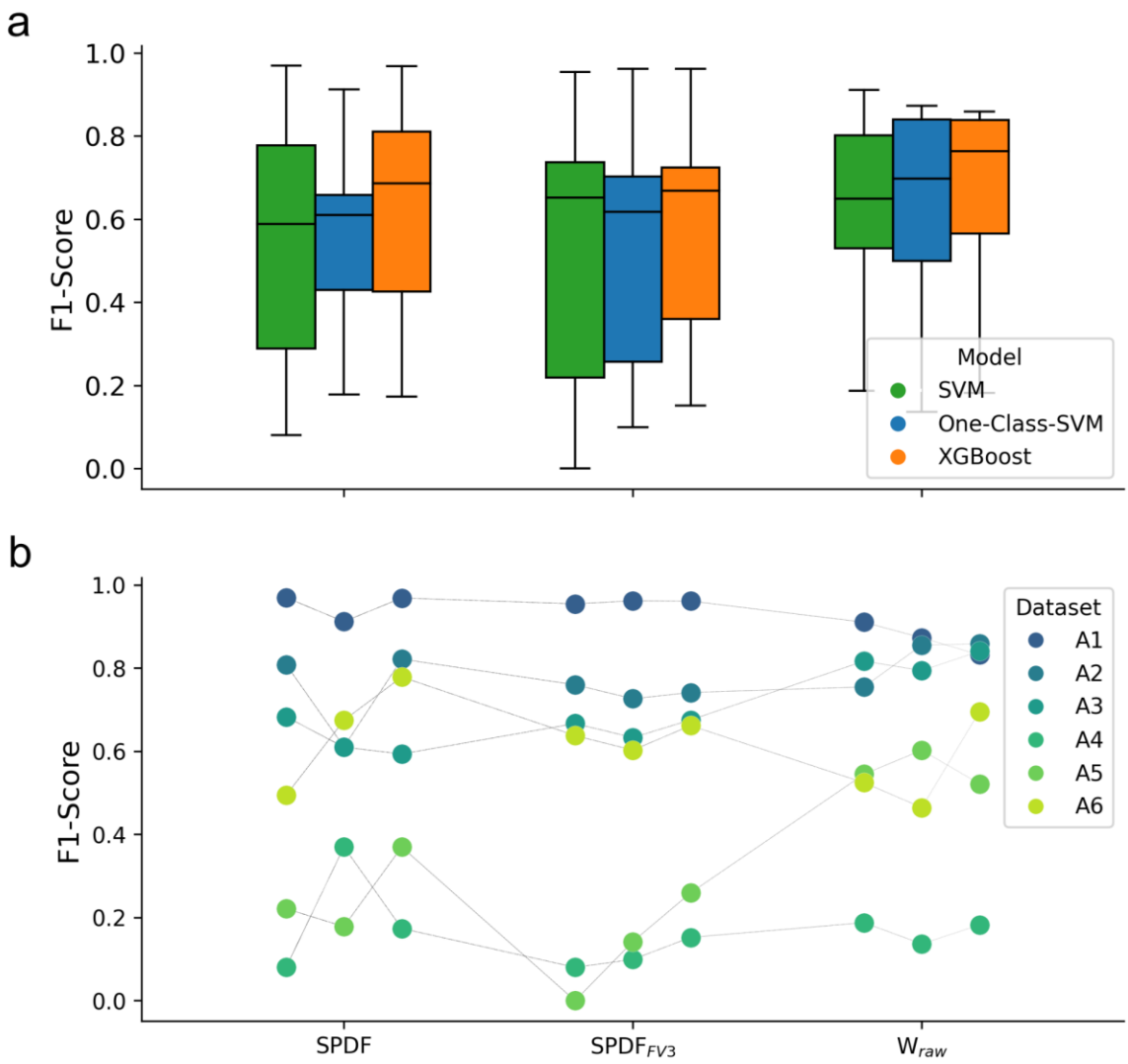


Figure 7. (a) Boxplots of F1-scores are shown for three feature sets (SPDF, $\mathrm{SPDF_{FV3}}$, and $\mathrm{W_{raw}}$), evaluated across three classification models: One-Class-SVM, standard SVM, and XGBoost. Each boxplot summarizes the distribution of F1-scores for all six recordings with the horizontal line indicating the median. **(b)** Dataset-specific trends are illustrated, showing F1-scores for each dataset (A1-A6) across the combinations. The grey lines connect scores for each dataset to illustrate relative changes in performance. While some recordings, such as, A1 show good performance for all combinations, others show greater variability depending on the chosen feature set and model (for example, A5).

For the statistical analysis, the outcome measurements were divided into two categories: bounded metrics (F1-score, precision, and recall) and count metrics (TP, FP, TN, and FN). An initial assessment showed that none of the metrics followed a normal distribution. Consequently, we fitted different mixed-effects models appropriate to each metric type, as described in the Methods section. Model comparisons via information criteria (AIC/BIC) indicated that the Beta mixed model consistently provided the best fit for the bounded metrics, while the negative Binomial model exhibited the lowest AIC/BIC values and favorable dispersion ratios for count metrics. In both final models, plots of residuals versus fitted values revealed a random scatter around zero, suggesting no discernible patterns and confirming that variability in the data was adequately captured.

We applied the mixed-effects models to investigate the influence of the classification models, feature sets, and their interaction on various performance metrics statistically (see Table 3). The One-Class-SVM model and the SPDF feature set were used as a baseline and compared to the other models and feature sets, as well as their interactions. For the F1-score, no main effects or interactions reached statistical significance, suggesting relatively consistent performance across conditions. In contrast, false negatives showed multiple significant effects: both SVM and XGBoost models, as well as the $\mathrm{SPDF_{FV3}}$ feature set, significantly reduced FN counts compared to the baseline. Even though the interaction terms SVM * $\mathrm{SPDF_{FV3}}$, XGBoost * $\mathrm{SPDF_{FV3}}$, and SVM * $\mathrm{W_{raw}}$ increased FN significantly, the total effect, calculated by summing the main effects and the interaction effects for each combination, remained negative (SVM and $\mathrm{SPDF_{FV3}}$= -0.87, XGBoost and $\mathrm{SPDF_{FV3}}$= -0.74, SVM and $\mathrm{W_{raw}}$ = -0.92). A similar effect was observed for true positives, where the interaction between SVM and $\mathrm{SPDF_{FV3}}$ significantly reduced TP counts, but the total effect remained positive (0.05), indicating a potential trade-off between capturing more positives and avoiding false alarms. Models for TN and FP did not yield statistically significant predictors. These findings highlight that while overall F1-scores remain stable, specific configurations influence false positive and false negative distributions differently.

Table 3. Statistically significant parameter estimates ($p < 0.05$) from the Beta and negative Binomial mixed models across the eight-outcome metrics (F1-score, precision, recall, FN, FP, TN, and TP). Estimates for F1-score, precision, and recall are presented on the logit scale. Estimates for FN, FP, TN, and TP are presented on the log scale. Non-significant terms are omitted for clarity. Significance levels are denoted by asterisks (*, **, ***).

Parameter	Outcome Measurement	Parameter Estimate	p-value
$\mathrm{W_{raw}}$	Precision	1.63	0.00639 (**)
SVM	Recall	1.44	0.01922 (*)
SVM * $\mathrm{SPDF_{FV3}}$	Recall	-2.21	0.00751 (**)
SVM	FN	-1.75	0.000243 (***)
XGBoost	FN	-0.96	0.027265 (*)
$\mathrm{SPDF_{FV3}}$	FN	-1.47	0.001504 (**)
SVM * $\mathrm{SPDF_{FV3}}$	FN	2.35	0.000405 (***)
XGBoost * $\mathrm{SPDF_{FV3}}$	FN	1.70	0.007833 (**)
SVM * $\mathrm{W_{raw}}$	FN	1.59	0.016725 (*)
SVM * $\mathrm{SPDF_{FV3}}$	TP	-0.48	0.0416 (*)

Analyses using Spike2 are only reliable in high SNR recordings

As a baseline for comparison, we applied a standard spike sorting pipeline using Spike2 (v10.08), which processes the entire duration of each recording using thresholding and

template matching. When applied to the full recordings (recording durations in Table 1), as intended by Spike2, most datasets exhibited a high number of FP (see Figure 8a, Spike2 (Full Recording)). The only exception was recording A1, where the number of TP exceeded false positives. This can be attributed to a higher SNR (7.63), and only two tracks. In contrast, recordings A4 and A5 showed the highest number of FP per TP (see Table 4), with A4 also exhibiting the highest false negative rate, approximately 25% of spikes were not detected and sorted. A5 contains activity from five tracks, being a possible reason for the high number of TP but relatively low number of FN. To avoid the influence of recording duration giving a false impression, we also report scaled standard metrics, such as F1-score, precision, and recall (see Table 4). While a promising recall remains around 0.9 for almost all recordings except A4, the poor overall performance becomes evident with a mean F1-score of 0.33 and a mean precision of 0.25 across all recordings. All metrics are collected in Supplementary Table S3.

Table 4. Spike sorting performance using Spike2. For each recording, the detection threshold used in Spike2 is reported along with the resulting F1-score, precision, and recall values. The last column indicates the number of FP/TP, providing a measure of spike sorting specificity. In well-sorted recordings, this value should ideally be below 0.5, indicating fewer than one FP for every two TPs.

Dataset	Detection Threshold	F1-Score	Precision	Recall	Number of FP per TP
A1	-1700	0.91	0.88	0.95	0.14
A2	-68	0.40	0.26	0.91	3
A3	-54	0.33	0.20	0.95	4
A4	-640	0.04	0.02	0.76	44
A5	-248	0.03	0.02	0.91	62
A6	-8130	0.26	0.15	0.91	6

Furthermore, a core challenge in the analysis of our results is the dynamic of spike waveforms during long recording sessions. Initially, a template was created for the spikes of the track of interest, while other spikes that fall within the threshold range but do not match the template trigger the creation of new templates. In theory, these additional templates could be ignored. However, as the experiment progresses, the waveform of the spike of interest changes. This shift means that some templates initially classified as incorrect may later contain spikes from the track of interest. Consequently, post-hoc deletion of these later templates would result in missing relevant spikes. Table 5 shows that as an example for A2.

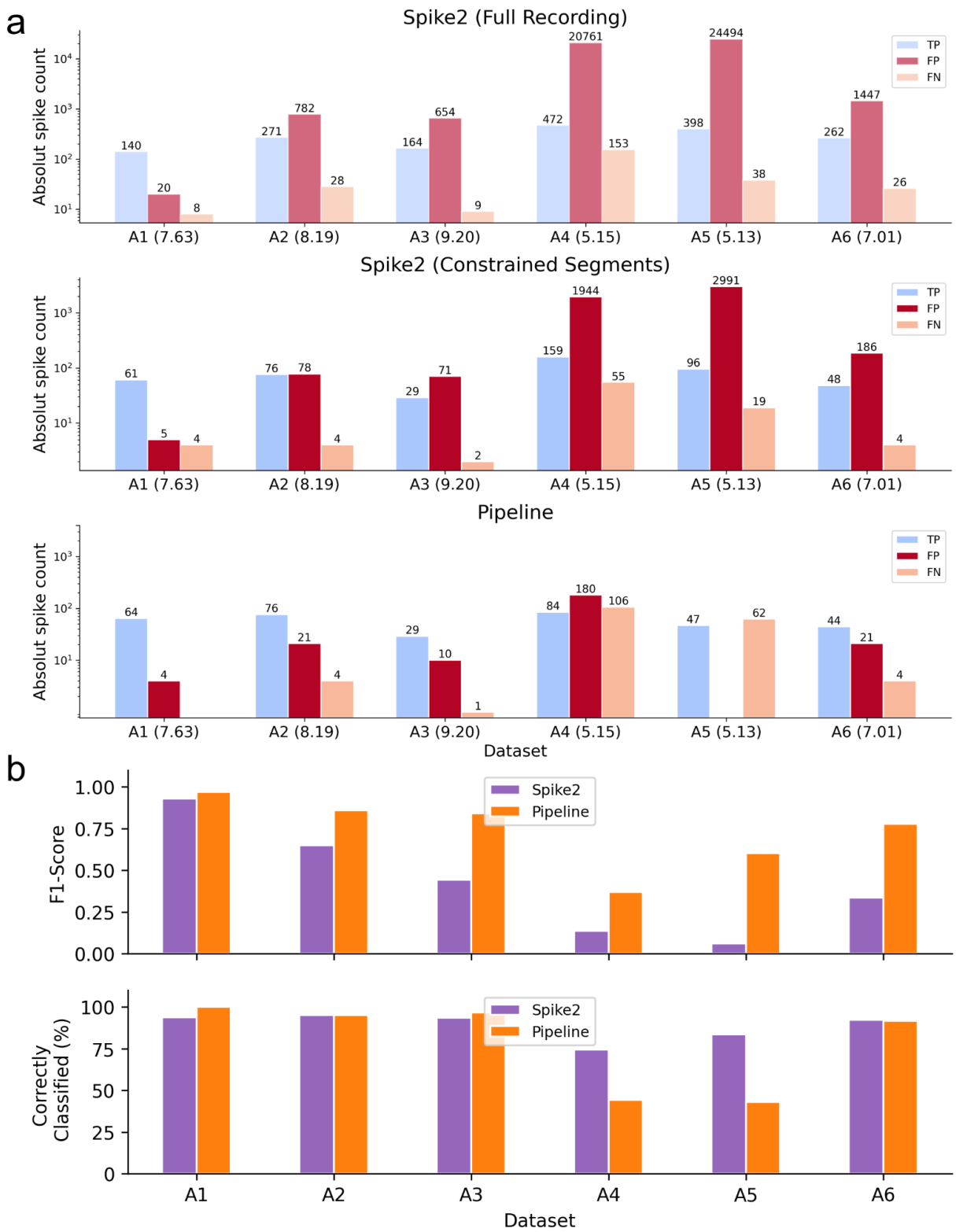


Figure 8. Spike detection and sorting performance comparison between Spike2 and our pipeline across datasets A1-A6. **(a)** Bar plots show absolute counts of TP, FP, and FN for each dataset on a log scale. The top panel shows results from Spike2 on the full recordings, the middle panel shows Spike2 results restricted to the same segments found via the pipeline, and the bottom panel shows results from our pipeline using the best-performing feature set-model combination (based on F1-score). The computed SNR for each dataset is shown in parentheses

next to the dataset name. **(b)** Quantitative comparison of sorting performance between Spike2 and our pipeline. The top plot displays F1-scores, while the bottom plot shows the percentage of correctly classified spikes.

Table 5. Impact of template selection strategy on performance in recording A2. The comparison of performances for two approaches is shown: using only the initial spike templates vs. retaining all templates generated during the analysis. We summarized TP, FP, FN, F1-scores, precision, and recall values for both strategies. Due to waveform drift over time, retaining all templates improves the spike detection and sorting that would otherwise be missed (recall improves to 0.91 from 0.55).

Template selection method	TP	FP	FN	F1-Score	Precision	Recall	Number of FP per TP
Only initial template used	164	181	135	0.51	0.48	0.55	1
All templates retained	271	782	28	0.40	0.26	0.91	3

This issue becomes particularly evident now that we have access to ground truth data, allowing us to verify how template evolution affects classification. In real-life scenarios, without ground truth validation, we would have to accept the limitations of a single-template approach, which would lead to an increased number of FNs. Alternatively, accepting all detected templates would capture more TP but at the cost of potential misclassification. This trade-off highlights the challenges in spike sorting and the difficulty in determining the optimal balance between sensitivity and specificity.

Based on these results and the observed recall values, we decided to retain all templates. Without ground truth data, it is impossible to determine with certainty which spikes belong to the fiber class and which do not. Prioritizing high recall ensures that we capture as many true spikes as possible, even at the cost of more false positives. While false positives could be possibly removed later through additional filtering or manual inspection, missed spikes cannot be recovered, making this approach the more reliable choice for our analysis.

To enable a direct comparison with our own pipeline, we restricted the search space to the same intervals where we observed latency jumps (see Figure 8a, Spike2 (Constrained Segments) and Pipeline). All metrics are collected in Supplementary Table S4.

In recordings with high SNR (A1-A3, A6), our pipeline demonstrated a clear advantage over Spike2, correctly identifying and sorting the majority of spikes while generating fewer false positives. In contrast, Spike2 exhibited higher false positive rates. Under more challenging recording conditions, for example, A4 and A5, both approaches showed limitations: Spike2 continued to produce a large number of false positives, whereas our pipeline was more restrictive, which resulted in fewer incorrect classifications but missing a substantial number of spikes, reflected as an increase in false negatives.

To provide a final performance comparison, we evaluated both F1-score and the proportion of correctly classified spikes for each dataset (see Figure 8b). Our pipeline outperformed Spike2 in terms of F1-scores across all datasets, indicating a better balance between true

positive and false positive classifications, though Spike2 yielded a higher percentage of correctly classified spikes in A4-A6. Notably, for dataset A1, both methods achieved similar performance with nearly equivalent F1-scores and high classification accuracy. Spike2 maintained a higher percentage of correctly classified spikes, likely due to its tendency to classify more events overall, including many false positives. This highlights a key difference: our pipeline establishes conservative but precise classification, whereas Spike2 prioritizes higher coverage at the expense of precision.

Using the described mixed-effect models from before, we compared each classification method against Spike2 (see Table 6). One-Class-SVM and XGBoost significantly outperformed the Spike2 baseline across multiple outcome metrics (F1-score, precision, and FP). The standard SVM showed a significant improvement only in terms of false positives (FP).

Table 6. Parameter estimates (with significance levels) for each model (One-Class-SVM, Standard SVM, and XGBoost) compared to the Spike2 baseline across six outcome metrics (F1-score, Precision, Recall, FN, FP, and TP). Estimates for F1-score, precision, and recall are presented on the logit scale, while those for FN, FP, and TP are on the log scale. In both scales, a positive estimate indicates a higher mean outcome compared to Spike2, and a negative estimate indicates a lower mean outcome. Significance levels are denoted by asterisks (*, **, ***).

Model/Outcome Metric	F1-Score	Precision	Recall	FN	FP	TP
One-Class-SVM	0.6834 (*)	0.9705 (*)	-0.20010	0.42670	-1.0024 (*)	-0.12747
SVM	0.4080	0.2958	0.09345	-0.09064	-1.0309 (*)	-0.08410
XGBoost	0.9663 (**)	0.7339	-0.12733	0.17622	-1.3898 (***)	-0.09989

Real-world testing: Applying our analysis to a recording with pruritogen injection

Finally, we evaluated the performance of our pipeline under real-world experimental conditions by applying it to a recording obtained during an experiment involving BAM 8-22 pruritogen injection. We used the combination of the W_{raw} feature set and XGBoost, selected for its consistently stable performance across datasets and its highest F1-score for this specific recording on the background spikes (0.97). To a recording obtained during an experiment in which the pruritogen BAM 8-22 was injected. The recording characteristics are summarized in Table 7, and F1-scores for all tested combinations are provided in Supplementary Table S5.

Table 7. Recording overview with pruritogen injection. The table shows the key characteristics of dataset AC, including the recording duration, the number of tracks, and the total number of extracted spikes. For each track, individual spike counts are listed. As before, we specified the track of interest and the signal-to-noise ratio.

Dataset	Recording duration in sec	Number of tracks	Total number of spikes	Track labels and distribution	Track label of interest	SNR
AC	300	2	144	Track1: 73 Track3: 71	Track3	11.81

The spike templates and a corresponding segment of background noise, along with the template from the track of interest are shown in Figures 9a and 9b. We applied both Spike2 and our computational pipeline to the same recording. Since no ground truth was available, we assessed the quality of spike sorting indirectly by visualizing the instantaneous frequency (IF), computed from the inter-spike intervals (ISI) of two consecutive spikes (see Figure 9d).

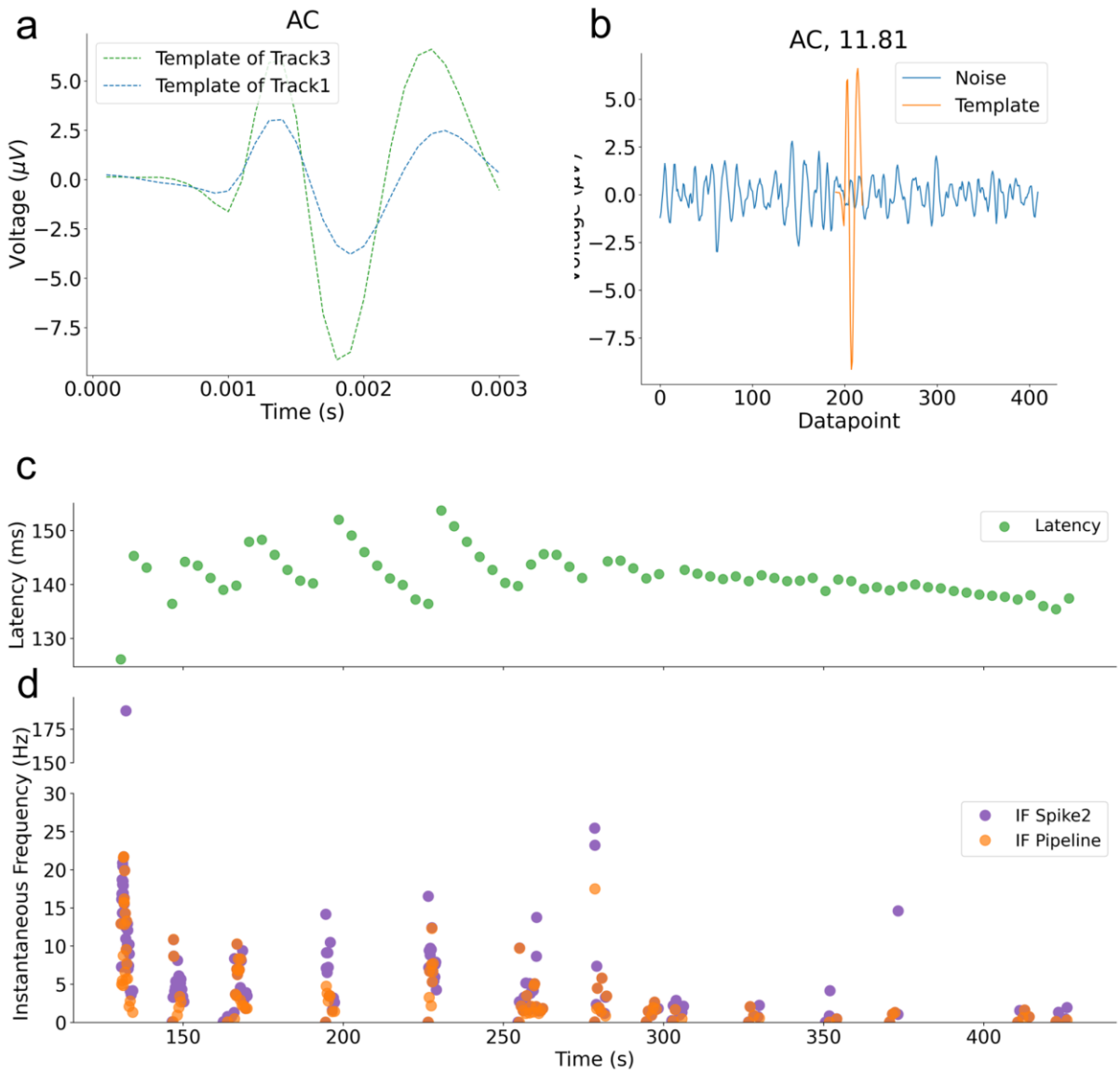


Figure 9. (a) Spike template for AC. Average spike template for both tracks, generated by aligning all extracted spikes. This visualization helped to identify the track of interest. **(b)** SNR visualization for AC. We calculated the SNR for the same track template, based on the shown noise segment. For AC, the SNR is 11.81, indicating a very high SNR signal suitable for reliable spike sorting. **(c)** Latency over time for track of interest. The latency provides critical validation for instantaneous frequency estimates. Sudden shifts in latency reflect the severity of fiber activity. In particular, very high instantaneous frequencies should be accompanied by pronounced latency

shifts. (d) Instantaneous firing frequency comparison between Spike2 and our pipeline for dataset AC. The instantaneous frequency (IF) was calculated from inter-spike intervals (ISIs) for two consecutive spikes, plotted over time. The purple dots show the IF from spikes sorted with Spike2, which includes at least two high-frequency outliers (200 Hz at 100 seconds, 17 Hz ca. 375 seconds), which could indicate false positives. The orange dots show the frequencies from our pipeline, which yield a more plausible firing pattern (0-20 Hz).

This analysis revealed that Spike2 produces several outlier spikes with instantaneous frequency near 200 Hz, an extremely high firing rate for a single C-fiber, indicating most probably false positive spikes. Notably, these false classifications occurred even in high SNR conditions. In contrast, our pipeline had firing rates that fell within a physiologically plausible range (0-20 Hz). While the absence of ground truths hinders validation, these findings suggest that our pipeline provides a more consistent spike sorting performance under high-frequency firing, which is beneficial when analyzing discharge patterns. All instantaneous firing frequencies obtained using Spike2 are reported in Supplementary Table S6, while those generated by our pipeline are presented in Supplementary Table S7.

To further assess the reliability of these frequency estimates, we examined latency shifts as an independent validation measure (see Figure 9c). Latency jumps are expected to reflect the intensity and timing of spikes, sharp increases in instantaneous frequency should coincide with latency jumps. For example, around 100 seconds, Spike2 shows a spike burst reaching nearly 200 Hz. However, no corresponding latency jump is observed, suggesting this may be a false prediction. Similarly, near 375 seconds, Spike2 again shows elevated frequencies, but the latency remains stable, reinforcing doubts about the validity of those spikes. In contrast, our pipeline has IF estimates that align more closely with the temporal evolution of latency, providing an additional indication of its robustness. Moreover, in regions such as just before 200 seconds, where Spike2 detects a strong burst that the pipeline does not, the presence or absence of latency shifts could serve as a deciding factor in evaluating which detection is accurate.

Discussion

In this work, we present a semi-automated spike sorting pipeline for noisy single-needle recordings with a focus on improving the classification of spike activity in nociceptors. Our approach leverages partial ground truth data derived from controlled electrical stimulation and knowledge about latency shifts resulting from activity-dependent slowing to constrain the search space for spike detection. By restricting the analysis to signal segments where spikes are expected, we reduce the likelihood of false positives and improve the classifier's precision.

Unlike many existing spike sorting approaches that apply unsupervised or fully data-driven algorithms, our method adopts a supervised framework. This allows us to train different classification models on verified spikes and to validate performance against spikes evoked by additional stimuli. Importantly, the pipeline is capable of extracting spiking patterns not only from evoked responses but also from chemically induced activity.

This pipeline opens new opportunities for the quantitative study of nociceptor behavior in response to sensory i.e., mechanical, chemical, or thermal stimuli as well as spontaneous activity patterns. Spontaneous activity is particularly important because it is linked to neuropathic pain patients and may help to better understand the underlying mechanisms [7]. By integrating the physiological characteristics of C-fibers directly into the spike sorting process, we aim to provide a more accurate and reproducible toolset for the study of peripheral sensory neurons.

The performance of our spike sorting pipeline is strongly influenced by the quality of the underlying microneurography recordings. In high-SNR data sets, such as recording A1, the model achieved high classification performance with only 4 false positives and one missed spike of in total 65 spikes. This demonstrates the feasibility of applying semi-automated spike sorting to peripheral nerve data when SNR and recording conditions are sufficient.

However, no single combination of classification models and feature sets emerged as a universally best-performing configuration across all datasets when considering F1-scores. This variability in model performance reflects the broader challenge of working with noisy single-electrode recordings, for example, microneurography data, which are highly heterogeneous due to differences in electrode placements, active fibers, and noise levels. Consequently, recording conditions remain a limiting factor for the reliability of spike sorting. When analyzing single-electrode recordings in the future, a step of identifying the best combination of feature sets and analyzing algorithms using ground truth data should be introduced.

In lower SNR recordings, such as datasets A4 and A5, we observed a substantial increase in both false positive and false negative rates. These findings suggest that the models can oscillate between overly conservative and overly permissive behavior, depending on the characteristics of the input data. This highlights a central challenge: balancing sensitivity and specificity in highly variable recordings. Additional constraints, such as physiological veto rules like ensuring C-fiber frequency cannot exceed 200 Hz might be beneficial and including information from other sources such as the observation that the magnitude of latency shift correlates to the number of previously conducted spikes might increase precision and validate the results [17].

In comparison to unsupervised methods, our results reinforce the conclusions of Hussein et al., who demonstrated that supervised machine learning outperforms template matching in noisy synthetic recordings [26]. Additionally, Meyer et al. also highlight in their comprehensive review of deep learning-based spike sorting that supervised approaches generally outperform unsupervised methods in terms of accuracy and robustness, especially when ground truth data is available for training [27].

Interestingly, the W_{raw} feature set aligns with the approach taken by Vogelstein et al. [28], who also applied SVMs directly to raw waveform data to evaluate the maximal expected

spike sorting performance, when applying unsupervised approaches. Similarly, our study aims to explore this theoretical performance ceiling by training and optimizing models on ground truth background spikes and subsequently evaluating their generalization on spikes lacking ground truth annotations.

Our application of XGBoost is motivated by its success in other domains. For instance, Sharifrazi et al. used XGBoost to classify spike patterns in mosquito-borne disease detection, highlighting the algorithm's robustness and flexibility in handling complex, high-dimensional biological data [29]. This is particularly relevant in spike sorting, where many discriminative features are not linearly separable in low-dimensional space, making ensemble methods like XGBoost well-suited for the task.

Additionally, our findings align with observations from the SpikeForest benchmarking initiative [14]. Although we have not evaluated our data within that framework, the observed variability in algorithmic performance depending on dataset characteristics matches our own experience. In particular, we found that pipelines such as MountainSort5 [30] and SpykingCircus 2 [31] were unable to resolve spike classes even in a high-SNR example recording [16]. This supports our impression that fully automatic spike sorting pipelines may impose limitations in heterogeneous recording conditions, such as those examined in microneurography.

When benchmarked against Spike2, one of the current tools used for spike sorting in noisy single-electrode recordings, our pipeline demonstrated superior performance, which aligns with the idea that templates may struggle in real-world settings with high signal variability**.** In particular, our pipeline significantly reduced the number of false positive detections while maintaining comparable sensitivity to verified spikes, although the template matching approach should be more restrictive than simple thresholding. Moreover, Spike2 relies heavily on manual curation, which limits reproducibility cross analysis, which is another disadvantage.

In contrast, our pipeline is designed to be open, reproducible, and extensible. Built using the Snakemake workflow system [20], it enables full transparency of each processing and allows users to easily replace or reconfigure detection methods, feature set extraction strategies, or classification algorithms. This module design promotes benchmarking and allows adaptation to different experimental setups. Moreover, as an open-source solution, it is freely available and encourages community-driven improvements, addressing a critical need for accessible and standardized tools in peripheral electrophysiology.

Our pipeline builds on prior work in spike sorting and feature extraction, particularly the SPDF feature sets, which were presented by Caro-Martín et al. [22] in a template-matching approach. In this work, spikes from the rostral-medial prefrontal cortex (rmPFC) of rabbits during classical eyeblink conditioning were recorded. These recordings were obtained using single-electrode techniques, capturing neural activity from the central nervous system. The differences in recording techniques and the inherent distinctions between CNS and peripheral

nerve fiber recordings may contribute to the higher performance observed in CNS studies. We observed that raw waveform features alone have a similar or even superior performance compared to more elaborate feature-engineered alternatives.

While broader frameworks like SpikerInterface [13] and SpikeForest [14] are valuable for benchmarking and comparing spike sorting algorithms, they are not optimized for the specific challenges of single-electrode recordings [16]. Notably, these frameworks typically operate in an unsupervised manner and do not leverage experiment-specific limitations, such as considering only intervals with latency shifts. In contrast, our use case is uniquely suited to a semi-supervised approach, as we always have partial ground truth available through background stimulation. This allows us to train and validate models in a way that most spike sorting frameworks for central recordings cannot support.

As such, our contribution is not only a new pipeline, but a task-specific framework applicable to any recordings in which spikes can be evoked by stimuli and used as ground truth training data, including, but not limited to peripheral nerve recordings. By integrating known biological constraints, we proved a more focused and interpretable solution in a domain where general-purpose methods fail or produce misleading results.

While our findings demonstrate the potential for a supervised spike sorting pipeline, several limitations must be acknowledged. The number of recordings analyzed was relatively small but reflected the full range of variability encountered in noisy electrode recordings. This variability is intrinsic to the technique and remains a central challenge for automated analysis promoting not limited to a single feature set and model combination.

A key ambiguity arises from false positive detection, which may in fact represent genuine spikes from other fibers. However, without fiber-specific ground truth, spike classes cannot be validated. Incorporating additional physiological characteristics, such as the expected maximum firing rate of C-fibers (typically < 200 Hz). This biological filter may improve the precision and interpretability of spike classification.

Importantly, traditional performance metrics like the F1-scores can be misleading in a physiological context. For instance, A2 achieved a maximal F1-score of 0.86 but included 21 false positives and 4 false negatives for 80 true positives, an error rate that may be unacceptable for analysis requiring precise spike quantification. This highlights the need for caution when interpreting machine learning metrics, and shows the need for new metrics to reflect, which error rate is tolerable depending on the research question.

Finally, spontaneous or chemically evoked activity lacks an external ground truth, which inherently limits validation. While our method performs well on electrically evoked responses, some degree of uncertainty will remain. As such, future advances will likely depend on integrating probabilistic models that can incorporate biological prior knowledge to improve the sorting of temporal spike patterns.

A key contribution of our work is the introduction of a novel strategy for generating experimental ground truth data using evoked responses. While this was developed for peripheral nerve recordings, it is not limited to this context. It holds potential for use in other sensory systems, including studies involving sensory prostheses, and may even be extendable to central nervous system recordings where stimulation-based validation is possible.
Our findings point to several directions for enhancing both the accuracy and generalizability of spike sorting of noisy single-electrode recordings. One promising approach involves incorporating additional physiological constraints, such as expected spike numbers from latency shifts.

In terms of spike detection, although the current version of our pipeline employs simple thresholding as a baseline, there is room for improvement through more advanced techniques. Future work could explore alternatives such as wavelet transforms [30, 31], which have proven effective in denoising and feature extraction, or autoencoders that enable unsupervised dimensionality reduction [34]. Additionally, deep learning-based tools like Variable Projection Networks (VPNets) offer end-to-end solutions that may further improve performance under noisy or complex recording conditions [33, 34].

Finally, the pipeline could benefit from incorporating deep-learning architectures specifically designed for extracellular data. Models capable of learning temporal and spatial dependencies directly from raw input without relying on hand-crafted features could significantly enhance robustness and adaptability across diverse experimental setups.

References

[1] Dayan P, Abbott LF. Theoretical Neuroscience: Computational And Mathematical Modeling of Neural Systems. Cambridge, Mass.: MIT Press; 2005.

[2] Grothe B, Klump GM. Temporal processing in sensory systems. Current Opinion in Neurobiology. 2000 Aug 1; **10(4)**: 467–73.

[3] Farkhooi F, Froese A, Muller E, Menzel R, Nawrot MP. Cellular Adaptation Facilitates Sparse and Reliable Coding in Sensory Pathways. PLOS Computational Biology. 2013 Oct 3; **9(10)**: e1003251.

[4] Stampanoni Bassi M, Iezzi E, Gilio L, Centonze D, Buttari F. Synaptic Plasticity Shapes Brain Connectivity: Implications for Network Topology. Int J Mol Sci. 2019 Dec 8; **20(24)**: 6193.

[5] Shouman K, Benarroch EE. Peripheral neuroimmune interactions: selected review and some clinical implications. Clin Auton Res. 2021; **31(4)**: 477–89.

[6] Vallbo ÅB. Microneurography: how it started and how it works. J Neurophysiol. 2018 Sep 1; **120(3)**: 1415–27.

[7] Kleggetveit IP, Namer B, Schmidt R, Helås T, Rückel M, Ørstavik K, et al. High spontaneous activity of C-nociceptors in painful polyneuropathy. Pain. 2012 Oct; **153(10)**: 2040–7.

[8] Fiebig A, Leibl V, Oostendorf D, Lukaschek S, Frömbgen J, Masoudi M, et al. Peripheral signaling pathways contributing to non-histaminergic itch in humans. Journal of Translational Medicine. 2023 Dec 12; **21(1)**: 908.

[9] Schmelz M. How Do Neurons Signal Itch? Front Med (Lausanne). 2021 Mar 15; **8**: 643006.

[10] Ackerley R, Watkins RH. Microneurography as a tool to study the function of individual C-fiber afferents in humans: responses from nociceptors, thermoreceptors, and mechanoreceptors. J Neurophysiol. 2018 Dec 1; **120(6)**: 2834–46.

[11] Namer B, Lampert A. Functional signatures of human somatosensory C fibers by microneurography. Pain. 2025 Apr 28.

[12] Turnquist B, RichardWebster B, Namer B. Automated detection of latency tracks in microneurography recordings using track correlation. Journal of Neuroscience Methods. 2016 Mar 15; **262**: 133–41.

[13] Buccino AP, Hurwitz CL, Garcia S, Magland J, Siegle JH, Hurwitz R, et al. SpikeInterface, a unified framework for spike sorting. Colgin LL, Grün S, Kloosterman F, editors. eLife. 2020 Nov 10; **9**: e61834.

[14] Magland J, Jun JJ, Lovero E, Morley AJ, Hurwitz CL, Buccino AP, et al. SpikeForest, reproducible web-facing ground-truth validation of automated neural spike sorters. Meister M, Calabrese RL, Meister M, editors. eLife. 2020 May 19; **9**: e55167.

[15] Buccino AP, Garcia S, Yger P. Spike sorting: new trends and challenges of the era of high-density probes. Prog Biomed Eng. 2022 May; **4(2)**: 022005.

[16] Troglio A, Konradi P, Fiebig A, Garriga AP, Rohrig R, Dunham J, et al. Supervised Spike Sorting Feasibility of Noisy Single-Electrode Extracellular Recordings: Systematic Study of Human C-Nociceptors recorded via Microneurography. bioRxiv; 2025. p. 2024.12.31.630860.

[17] Schmelz M, Forster C, Schmidt R, Ringkamp M, Handwerker HO, Torebjörk HE. Delayed responses to electrical stimuli reflect C-fiber responsiveness in human microneurography. Exp Brain Res. 1995; **104(2)**: 331–6.

[18] Serra J, Campero M, Ochoa J, Bostock H. Activity-dependent slowing of conduction differentiates functional subtypes of C fibres innervating human skin. J Physiol. 1999 Mar 15; **515(Pt 3)**: 799–811.

[19] Turnquist B. DAPSYS (Data Acquisition Processor System). Available from: http://dapsys.net/

[20] Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, et al. Sustainable data analysis with Snakemake. F1000Research; 2021.

[21] Konradi P, Troglio A, Pérez Garriga A, Pérez Martín A, Röhrig R, Namer B, et al. PyDapsys: an open-source library for accessing electrophysiology data recorded with DAPSYS. Frontiers in Neuroinformatics. 2023; **17**.

[22] Caro-Martín CR, Delgado-García JM, Gruart A, Sánchez-Campusano R. Spike sorting based on shape, phase, and distribution features, and K-TOPS clustering with validity and error indices. Sci Rep. 2018 Dec 12; **8(1)**: 17796.

[23] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. arXiv; 2018.

[24] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery; 2016. 785–94.

[25] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, NY, USA: Association for Computing Machinery; 2019. 2623–31.

[26] Ahmed Hussein H, Khorsheed Mohammed A. Improving the Accuracy of Neurons Spike Sorting by Using Supervised Machine Learning. ACAD J NAWROZ UNIV. 2024 Mar 31; **13(1)**: 451–64.

[27] Meyer LM, Zamani M, Rokai J, Demosthenous A. Deep learning-based spike sorting: a survey. J Neural Eng. 2024 Nov; **21(6)**: 061003.
[28] Jacob Vogelstein R, Murari K, Thakur PH, Diehl C, Chakrabartty S, Cauwenberghs G. Spike sorting with support vector machines. Conf Proc IEEE Eng Med Biol Soc. 2004;2006: 546–9.
[29] Sharifrazi D, Javed N, Alizadehsani R, Paradkar PN, Rajendra Acharya U, Bhatti A. Automated detection of Zika and dengue in *Aedes aegypti* using neural spiking analysis: A machine learning approach. Biomedical Signal Processing and Control. 2024 Oct 1; **96**: 106594.
[30] Chung JE, Magland JF, Barnett AH, Tolosa VM, Tooker AC, Lee KY, et al. A Fully Automated Approach to Spike Sorting. Neuron. 2017 Sep 13; **95(6)**: 1381-1394.e6.
[31] Yger P, Spampinato GL, Esposito E, Lefebvre B, Deny S, Gardella C, et al. A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. eLife. **7**: e34518.
[32] Prada J, Sasi M, Martin C, Jablonka S, Dandekar T, Blum R. An open source tool for automatic spatiotemporal assessment of calcium transients and local 'signal-close-to-noise' activity in calcium imaging data. PLOS Computational Biology. 2018 Mar 30; **14(3)**: e1006054.
[33] Lee GCF, Libedinsky C, Guan C, So R. Use of wavelet transform coefficients for spike detection for a Robust Intracortical Brain Machine Interface. In: 2017 8th International IEEE/EMBS Conference on Neural Engineering (NER). 2017. 540–3.
[34] Ardelean ER, Coporîie A, Ichim AM, Dînșoreanu M, Mureșan RC. A study of autoencoders as a feature extraction technique for spike sorting. PLOS ONE. 2023 Mar 9; **18(3)**: e0282810.
[35] Kovács P, Bognár G, Huber C, Huemer M. VPNet: Variable Projection Networks. Int J Neur Syst. 2022 Jan; **32(01)**: 2150054.
[36] Troglio A, Fiebig A, Kutafina E, Namer B. Automated Spike Sorting in Microneurography: A Proof-of-Concept through Classification on Ground Truth Data. 2024.

Acknowledgements

We acknowledge Danxia Bao for her technical support.

Funding

BN is supported by the German research Council (DFG) NA 970 6-2, 7-1, 9-2.

Author contributions

B.N. and E.K. conceptualized the work; B.N., A.F., and A.M. performed the experiments and acquired the data; A.F. curated the data; A.T., E.K., B.N., and A.M. analyzed the data; A.T. developed the computational pipeline; E.K. and B.N. interpreted the data; All authors critically discussed the results; A.T. wrote the main manuscript text and prepared the figures; All authors have revised the manuscript and approved the final version.

Data availability statement

The raw datasets generated and/or analyzed during the current work are available from the corresponding author on reasonable request. The source code and selected test data are available on https://github.com/Digital-C-Fiber/SpikeSortingForSpikingPatterns.

Additional Information

Competing Interests Statement

BN received consulting fees from Vertex. The other authors declare no competing interests.

Figure legends

Figure 1. (a) Example waterfall plot of two tracks**.** The signal is segmented every 4 seconds, beginning with the electrical background stimulus. Two tracks are visible, marked by the green and blue dots, indicating the activity of two C-fibers. The conduction velocity stays almost constant leading to vertical alignment of the elicited spikes. **(b)** An example spike at noise level. The spike in the blue box would not be detectable without the vertical alignment of repeated responses enabled by the marking method and waterfall representation. This temporal alignment facilitates the identification of low-amplitude spikes by leveraging consistent latencies evoked by background stimuli.

Figure 2. (a) Example of activity-dependent slowing. Each line begins with the periodic background pulse indicated by the blue rectangle, resulting in signal segments with a duration of 4 seconds. When no extra activity is observed (lines 1-3) the spikes (green dots) align vertically due to the almost constant conduction velocity. When additional (electrical) stimuli are applied, there is a decrease in conduction velocity (line 4). Responses to background stimuli are marked by green dots with a blue circle. **(b)** Example of activity-dependent slowing stimulated with the ground truth protocol. The signal is cut just before each stimulation pulse, whether it is a background pulse or an additional pulse. This ensures that all spikes are aligned on the green track. Responses to background stimuli remain marked by green dots with blue circles, while responses to additional stimuli are indicated by purple rectangles and circles, both for the stimuli and the evoked spikes.

Figure 3. Overview of a hybrid spike detection and evaluation pipeline combining knowledge- and data-driven steps. (1) Spike templates are extracted from background spikes (blue circle) using low-frequency stimulation to identify the track of interest for further analysis. (2) Search spaces containing activity are identified, and spikes are detected using threshold-based methods. (3) Feature sets are computed from background spikes and used as input to different classification models (One-Class SVM, SVM, XGBoost). Model performance is evaluated through 5-fold cross-validation to determine the optimal model-feature set combination for the background spikes. (4) Evaluate detection and sorting based on ground truth. First, spike detection is assessed by comparing detected spikes to known ground-truth spikes, classifying outcomes as true positives, false positives, false negatives, or true negatives. Next, sorting performance is evaluated by verifying whether detected spikes are correctly assigned to their originating fibers. (5) Benchmark detection and sorting against Spike2 with ground truth data. The pipeline's detection and sorting performance is directly compared to Spike2 by applying both methods independently to the same recording segments. (6) Assess chemical spikes performance. For recordings with the pruritogen injection, we compute the instantaneous frequency and latency of chemically evoked spikes

sorted by both the pipeline and Spike2. Since no ground truth is available for these spikes, we compare the resulting patterns to assess how plausible and physiologically consistent the detected activity is across methods.

Figure 4. (a) Average spike templates for each track in all recordings**.** For each dataset, we extracted spike waveforms from each track and aligned them according to the time point corresponding to the largest downward slope (time point of the maximum negative peak in the first derivative of the signal as the temporal midpoint) [22]. We averaged all spikes to generate the templates. Each subplot shows the templates for all tracks within a single recording. These templates reflect the characteristic spike shapes, enable the visual comparison of waveform morphology, and support the selection for the track of interest. **(b)** Signal-to-noise (SNR) visualization for each recording and track of interest. For each recording, the track containing the spikes with the largest amplitudes was used to calculate the SNR. We calculated the SNR by dividing the peak amplitude of the template by the standard deviation of a noise segment extracted from the raw recording. The noise standard deviation was estimated using the median absolute deviation (MAD). The computed SNR for each dataset is indicated next to its name in the figure title.

Figure 5. Spike detection performance using thresholding and peak detection. TP, FP, and TN are visualized on a log scale for the detected spikes in the constrained search space. All recordings have higher false positive values compared to the TP values. A4 exhibits the most false positives and 22 spikes could not be detected with the set threshold.

Figure 6. (a) Heatmaps of spike sorting performance across feature sets, classification models, and evaluation metrics. Each cell is color-coded according to the F1-score with magenta indicating low performance and green indicating high performance. The heatmaps display results for all tested combinations of three feature sets and classification models, evaluated using multiple metrics (F1-score, TP, FP, FN, and TN). A rectangle highlights the best-performing configurations based on the F1-score for each recording. **(b)** Relationship between SNR and best F1-score for recordings with varying number of tracks. Each dot represents a recording colored by the number of simultaneously recorded tracks (2, 3, or 5). As expected, lower SNR values correspond to reduced F1-scores. Notably, the F1-score plateaus beyond an SNR of approximately 7.5, indicating no further improvement in performance.

Figure 7. (a) Boxplots of F1-scores are shown for three feature sets (SPDF, $\mathrm{SPDF}_{\mathrm{FV3}}$, and $\mathrm{W}_{\mathrm{raw}}$), evaluated across three classification models: One-Class-SVM, standard SVM, and XGBoost. Each boxplot summarizes the distribution of F1-scores for all six recordings with the horizontal line indicating the median. **(b)** Dataset-specific trends are illustrated, showing F1-scores for each dataset (A1-A6) across the combinations. The grey lines connect scores for each dataset to illustrate relative changes in performance. While some recordings, such as, A1 show good performance for all combinations, others show greater variability depending on the chosen feature set and model (for example, A5).

Figure 8. Spike detection and sorting performance comparison between Spike2 and our pipeline across datasets A1-A6. **(a)** Bar plots show absolute counts of TP, FP, and FN for each dataset on a log scale. The top panel shows results from Spike2 on the full recordings, the middle panel shows Spike2 results restricted to the same segments found via the pipeline, and the bottom panel shows results from our pipeline using the best-performing feature set-model combination (based on F1-score). The computed SNR for each dataset is shown in parentheses next to the dataset name. **(b)** Quantitative comparison of sorting performance between Spike2 and our pipeline. The top plot displays F1-scores, while the bottom plot shows the percentage of correctly classified spikes.

Figure 9. (a) Spike template for AC. Average spike template for both tracks, generated by aligning all extracted spikes. This visualization helped to identify the track of interest. **(b)** SNR visualization for AC. We calculated the SNR for the same track template, based on the shown noise segment. For AC, the SNR is 11.81, indicating a very high SNR signal suitable for reliable spike sorting. **(c)** Latency over time for track of interest. The latency provides critical validation for instantaneous frequency estimates. Sudden shifts in latency reflect the severity of

fiber activity. In particular, very high instantaneous frequencies should be accompanied by pronounced latency shifts. (d) Instantaneous firing frequency comparison between Spike2 and our pipeline for dataset AC. The instantaneous frequency (IF) was calculated from inter-spike intervals (ISIs) for two consecutive spikes, plotted over time. The purple dots show the IF from spikes sorted with Spike2, which includes at least two high-frequency outliers (200 Hz at 100 seconds, 17 Hz ca. 375 seconds), which could indicate false positives. The orange dots show the frequencies from our pipeline, which yield a more plausible firing pattern (0-20 Hz).

Tables

Table 1. Data overview. The table summarizes the key properties of each dataset, including the total recording duration, the number of tracks (i.e., distinct recorded nerve fibers), and the total number of extracted spikes. For each track, individual spike counts are listed. Additionally, we specify the track selected for detailed analysis, and the signal-to-noise ratio (SNR) computed for the track of interest.

Dataset	Recording duration in sec	Number of Tracks	Total number of spikes	Track labels and distribution	Track label of interest	SNR	Threshold
A1	375	2	293	Track1: 148 Track3: 145	Track1	9.51	3.60
A2	923	2	597	Track3: 299 Track4: 298	Track3	6.82	4.64
A3	581	2	346	Track1: 173 Track2: 173	Track2	5.54	5.12
A4	1774	3	1780	Track1: 585 Track2: 570 Track4: 625	Track4	3.06	3.50
A5	1390	5	2167	Track1: 436 Track5: 437 Track20: 437 Track11: 436 Track23: 421	Track11	6.77	5.00
A6	740	3	796	Track1: 288 Track9: 220 Track13: 288	Track1	5.54	3.00

Table 2. Spike detection performance. For each recording, the detection threshold for the Dapsys recordings, F1-scores, precision, and recall are reported, along with the number of FP/TP. We set the thresholds based on the amplitudes of the spike templates and optimized the balance between reducing the number of false negatives and false positives.

Dataset	Detection Threshold	F1-Score	Precision	Recall	Number of FP per TP
A1	3.60	0.19	0.11	0.98	8
A2	4.64	0.37	0.22	1.00	3
A3	5.12	0.49	0.33	0.97	2
A4	3.50	0.08	0.04	0.90	23

A5	5.00	0.11	0.06	0.95	16
A6	3.00	0.17	0.10	0.90	9

Table 3. Statistically significant parameter estimates ($p < 0.05$) from the Beta and negative Binomial mixed models across the eight-outcome metrics (F1-score, precision, recall, FN, FP, TN, and TP). Estimates for F1-score, precision, and recall are presented on the logit scale. Estimates for FN, FP, TN, and TP are presented on the log scale. Non-significant terms are omitted for clarity. Significance levels are denoted by asterisks (*, **, ***).

Parameter	Outcome Measurement	Parameter Estimate	p-value
W_{raw}	Precision	1.63	0.00639 (**)
SVM	Recall	1.44	0.01922 (*)
SVM * $SPDF_{FV3}$	Recall	-2.21	0.00751 (**)
SVM	FN	-1.75	0.000243 (***)
XGBoost	FN	-0.96	0.027265 (*)
$SPDF_{FV3}$	FN	-1.47	0.001504 (**)
SVM * $SPDF_{FV3}$	FN	2.35	0.000405 (***)
XGBoost * $SPDF_{FV3}$	FN	1.70	0.007833 (**)
SVM * W_{raw}	FN	1.59	0.016725 (*)
SVM * $SPDF_{FV3}$	TP	-0.48	0.0416 (*)

Table 4. Spike sorting performance using Spike2. For each recording, the detection threshold used in Spike2 is reported along with the resulting F1-score, precision, and recall values. The last column indicates the number of FP/TP, providing a measure of spike sorting specificity. In well-sorted recordings, this value should ideally be below 0.5, indicating fewer than one FP for every two TPs.

Dataset	Detection Threshold	F1-Score	Precision	Recall	Number of FP per TP
A1	-1700	0.91	0.88	0.95	0.14
A2	-68	0.40	0.26	0.91	3
A3	-54	0.33	0.20	0.95	4
A4	-640	0.04	0.02	0.76	44
A5	-248	0.03	0.02	0.91	62
A6	-8130	0.26	0.15	0.91	6

Table 5. Impact of template selection strategy on performance in recording A2. The comparison of performances for two approaches is shown: using only the initial spike templates vs. retaining all templates generated during the analysis. We summarized TP, FP, FN, F1-scores, precision, and recall values for both strategies. Due to waveform drift over time, retaining all templates improves the spike detection and sorting that would otherwise be missed (recall improves to 0.91 from 0.55).

Template selection method	TP	FP	FN	F1-Score	Precision	Recall	Number of FP per TP
Only initial template used	164	181	135	0.51	0.48	0.55	1
All templates retained	271	782	28	0.40	0.26	0.91	3

Table 6. Parameter estimates (with significance levels) for each model (One-Class-SVM, Standard SVM, and XGBoost) compared to the Spike2 baseline across six outcome metrics (F1-score, Precision, Recall, FN, FP, and TP). Estimates for F1-score, precision, and recall are presented on the logit scale, while those for FN, FP, and TP are on the log scale. In both scales, a positive estimate indicates a higher mean outcome compared to Spike2, and a negative estimate indicates a lower mean outcome. Significance levels are denoted by asterisks (*, **, ***).

Model/Outcome Metric	F1-Score	Precision	Recall	FN	FP	TP
One-Class-SVM	0.6834 (*)	0.9705 (*)	-0.20010	0.42670	-1.0024 (*)	-0.12747
SVM	0.4080	0.2958	0.09345	-0.09064	-1.0309 (*)	-0.08410
XGBoost	0.9663 (**)	0.7339	-0.12733	0.17622	-1.3898 (***)	-0.09989

Table 7. Recording overview with pruritogen injection. The table shows the key characteristics of dataset AC, including the recording duration, the number of tracks, and the total number of extracted spikes. For each track, individual spike counts are listed. As before, we specified the track of interest and the signal-to-noise ratio.

Dataset	Recording duration in sec	Number of tracks	Total number of spikes	Track labels and distribution	Track label of interest	SNR
AC	300	2	144	Track1: 73 Track3: 71	Track3	11.81