IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 February 2025, accepted 7 March 2025, date of publication 17 March 2025, date of current version 11 April 2025.

*Digital Object Identifier 10.1109/ACCESS.2025.3552008*

RESEARCH ARTICLE

# Binarized Neural Networks for Resource-Efficient Spike Sorting

**LUCA M. MEYER[1], MAJID ZAMANI[2], (Member, IEEE), AND ANDREAS DEMOSTHENOUS[3], (Fellow, IEEE)**

[1]65187 Wiesbaden, Germany
[2]School of Electronics and Computer Science, University of Southampton, SO17 1BJ Southampton, U.K.
[3]Department of Electronic and Electrical Engineering, University College London, WC1E 7JE London, U.K.

Corresponding author: Andreas Demosthenous (a.demosthenous@ucl.ac.uk)

**ABSTRACT** Deep learning is fastly gaining ground in neuroscience. In the field of implantable brain computer interfaces, a fundamental application of deep learning is to sort action potentials (known as spikes), measured with extracellular electrodes, according to their origin neurons. This enables the generation of precise modulatory patterns of neuronal circuits. Deep learning-based spike sorting algorithms are based on power-intensive dot products, which poses challenges for on-chip processing with resource-constrained devices. In contrast, binarized neural networks offer great potential for on-chip sorting, mainly relying on bitwise operations and accumulations. However, recently published binarized models perform significantly worse than deep full-precision networks and fail on challenging neural data. This work presents a binarized neural network for spike sorting that narrows the performance gap between recently developed binarized models and more accurate full-precision models. The novelty of this work resides in the developed network architecture. In comparison to previous research, this work presents a deep binarized neural network featuring two hidden layers, each containing 256 units to effectively capture the spike characteristics of complex neural data. Before training, spikes were pre-sorted in an unsupervised way to generate pseudo-labels. Subsequently, the deep binarized model and an equally sized full-precision model were trained and evaluated using experimentally obtained and synthetic spike waveforms. The proposed binarized model could achieve results close to more advanced network types, such as convolutional and long short-term memory networks, which is remarkable considering that the binarized model was primarily designed to maintain a balance between resource consumption and accuracy. The equally sized full-precision model could even outperform the aforementioned models, despite its much lighter architecture.

**INDEX TERMS** Binarized neural network, deep learning, implantable brain computer interfaces, neural spike classification, signal processing, spike sorting.

## I. INTRODUCTION

Neuronal signals can be captured using extracellular recordings, which typically show the activity of a small number of neurons that are communicating in the vicinity of the recording electrode by transmitting electrical impulses. The analysis of these signals on a cellular level offers insights into the functional behaviors of neuronal circuits. Clinically, it aids in gaining an understanding of neurological conditions like paralysis [1] or cognitive loss [2]. By using implantable brain computer interfaces (iBCIs), the neuronal signals can be translated into commands to control external devices [3]. iBCIs not only offer the ability to read neural signals but also enable the alteration of neural circuits through electrical stimulation [4]. In recent years, wireless iBCIs have gained lots of interest in the field of neurotechnology, which has been boosted by the emergence of Neuralink's wireless iBCI [5]. Wireless signal transmission significantly increases the practicality of these devices for daily use. However, the wireless transmission of neural signals captured with multiple electrodes usually results in high power consumption and delay which limits real-time processing. In [6], the authors illustrate this problem with the following example: using a multi-electrode array (MEA) with 100 channels, sampled

The associate editor coordinating the review of this manuscript and approving it for publication was Cesar Vargas-Rosales.

at 20 kHz with 10-bit resolution, results in a data stream of 20 Mbps. Nevertheless, the data stream can be reduced to 2-3 kbps if only spiking activity is transmitted. To reduce the data rate, there are two approaches to choose from: i) spikes are detected and compressed on-chip before transmission [8]; ii) spikes are detected and sorted on-chip [9], resulting in binary spike trains, (i.e., raster activity) that can be transmitted for external signal analysis [10], [11]. Sorting the spikes according to their neuron of origin has proven to be helpful in many applications such as visual stimuli [12] and memory decoding [2] and therefore adds value to iBCIs.

Different features of spike waveforms are mainly influenced by the type of cell, distribution of ion channels, structure of dendritic trees, and orientation and distance to the recording electrode and lead to characteristic waveforms that enable sorting [13], [14]. Conventionally spike sorting is a multi-step procedure with the following steps: a) preprocessing, which usually includes band-pass filtering [15]; b) spike detection: this is often done with voltage thresholding [15] or the non-linear energy operator method [16]; c) spike alignment: typically, spikes are aligned to their amplitude peaks as this facilitates the following step [15]; d) feature extraction: methods like Principle Component Analysis (PCA) [17] or Independent Component Analysis (ICA) [18] are used to reduce dimensions; e) clustering: the reduced feature space enables clustering with standard methods like k-means or hierarchical clustering [19]. Alongside the pipeline for spike sorting described above, there are also classic approaches such as template matching [20], as well as more modern solutions based on the use of deep neural networks (DNNs) to detect spikes [21] extract features [22], and classify spikes concerning their underlying neuron [23]. In 2024, the authors of this paper published a comprehensive survey of spike sorting models based on deep learning, providing more details in this area [24].

Most spike sorting solutions that are based on neural networks are not specifically designed for in vivo operation but for use on external processing units. The majority of studies in this area deal with the processing of single-channel data and use both synthetic and experimentally acquired recordings for the training and evaluation of their models [25], [26], [27], [28]. These studies involve supervised learning models that are provided with the corresponding label for each example during training. In addition to the models mentioned so far, networks with fully binarized weights and activations (BNNs) have also been presented in recent years, which are detailed below.

In 2021, Valencia and Mohammad [29] presented a super low-complexity solution for neural spike classification. Using the synthetic data in [15], their model takes a spike waveform as input (64 samples) and processes it through one hidden layer with only five hidden units. Trained and tested on the data in [15] they achieved an average accuracy of 90%, which is remarkable considering the shallow network architecture. In 2023, Valencia and Mohammad presented another solution based on partially binarized neural networks [30]. In contrast to their previous work, they applied a resource-efficient feature extraction method based on discrete derivates (DD-2Ex [31]) to the raw spike waveforms to reduce the network input from 64 to 4 samples. The model in [30] also has one hidden layer which, like the output layer, consists of only three units. Valencia and Mohammad carried out experiments on different quantization schemes. The model without binarization achieved the best results with an average accuracy of 88.1%, while the performance drop was already severe for the model with binarized output layer weights, which achieved an average accuracy of 69.69%. Nevertheless, the authors in [30] demonstrated that spike classification with super low complexity models can yield acceptable performance. Notably, the performance gap to other full-precision models (FPMs) is quite high. The much heavier deep learning networks mentioned at the beginning of this section managed to reach accuracies of more than 99% using the same data, clearly showcasing the trade-off between model accuracy and efficiency. However, due to their computational simplicity, BNNs are much better candidates for on-chip processing than the named FPMs but quickly reach their limits with more complex neural data. Attempts have been made in the past to reduce the resource consumption of deep learning models in spike sorting, such as Seong et al [9], who developed neural networks with quantized weights to save memory. The primary limitations of these models lie in the required chip area and power per processed channel. Minimizing resource consumption is therefore essential to develop scalable next-generation neurotechnology.

This work addresses this challenge by designing a model that maximizes performance while operating with minimal resource usage. The proposed binarized neural network with two hidden layers was trained and tested on both synthetic and experimentally obtained data, aimed at outperforming the classification accuracy of the existing BNN-based spike sorting models [29], [30] and reaching comparable performance to the DNN-based models proposed in [25], [26], [27], [28]. Thereby, the main focus was on maintaining low resource consumption to enable on-chip spike sorting. In contrast to [9], where weights were quantized to 4 or more bits, the weights and activations of the neural network in this work were quantized to one bit, which not only reduces memory but also simplifies the computation since no power-intensive dot-products are required during inference, creating optimal conditions for on-chip processing. To the best of our knowledge, this work represents the first ´deep´ BNN used for spike sorting. In addition to the proposed BNN, an equally sized FPM was developed to investigate the influence of binarization on the results. In contrast to the models just mentioned, this work did not use the true labels of the synthetic data. Instead, a pseudo-labeling technique was developed to simulate real experimental conditions. By using more challenging experimental data, this work demonstrates that the low-complexity model proposed in [29] is not capable

**TABLE 1. Spike counts per neuron (N) across channels in dataset1.**

| Channel ID | N1 | N2 | N3 | N4 | N5 |
|---|---|---|---|---|---|
| 125 | 1297 | 9315 | | | |
| 66 | 10076 | 6919 | 3455 | | |
| 69 | 11719 | 11183 | 9140 | | |
| 77 | 2217 | 16577 | 6261 | | |
| 79 | 2441 | 13441 | 3845 | | |
| 84 | 3682 | 6900 | 1485 | | |
| 87 | 423 | 40703 | 1819 | | |
| 91 | 4022 | 12711 | 6865 | | |
| 92 | 1630 | 15481 | 6344 | | |
| 94 | 308 | 1685 | 2367 | | |
| 98 | 103 | 9998 | 1067 | | |
| 99 | 810 | 19872 | 3938 | | |
| 100 | 4525 | 29771 | 6801 | | |
| 101 | 2209 | 22729 | 6213 | | |
| 108 | 2368 | 9954 | 3901 | | |
| 109 | 4764 | 12568 | 10724 | | |
| 112 | 187 | 1551 | 1165 | | |
| 114 | 4207 | 16073 | 6219 | | |
| 115 | 679 | 15541 | 12832 | | |
| 122 | 794 | 13904 | 8368 | | |
| 124 | 972 | 15163 | 5266 | | |
| 67 | 621 | 2507 | 2207 | 2361 | |
| 68 | 889 | 7017 | 2068 | 2164 | |
| 70 | 1672 | 8910 | 5448 | 1979 | |
| 83 | 2233 | 7289 | 9073 | 3063 | |
| 95 | 522 | 37769 | 13478 | 2980 | |
| 107 | 6261 | 9166 | 1040 | 795 | |
| 116 | 6402 | 12898 | 2955 | 1678 | |
| 82 | 76 | 9441 | 1216 | 478 | 494 |
| 90 | 1626 | 6924 | 2965 | 2946 | 2039 |
| 126 | 11853 | 14038 | 10046 | 4460 | 4011 |

of providing sufficient sorting results. Moreover, it is shown that the previously mentioned DNNs are fundamentally oversized, requiring extreme implementation costs which render them unsuitable for iBCI applications. Comparable results can be achieved with much simpler models, as long as the data is prepared appropriately and machine learning best practices are considered.

Section II describes the methods used in this paper including the datasets, the data preparation technique, and the neural network selection and architecture. Section III presents the results achieved. In Section IV, the general performance of the method is discussed and the models are compared with the aforementioned deep learning-based spike sorting algorithms. Furthermore, this section discusses the hardware considerations, addresses the limitations of this work and outlines future directions.

## II. METHODS

### A. DATASETS

This study used experimentally obtained recordings and synthetic neural data to train and evaluate the proposed models. In the following, the experimental recordings are often referred to as Dataset1 and the synthetic data as Dataset2.

#### 1) EXPERIMENTAL DATA

The proposed models in this work were optimized, trained and evaluated on extracellular recordings obtained from the primary cortex (V1) of macaque monkeys [33], [34]. Dataset1

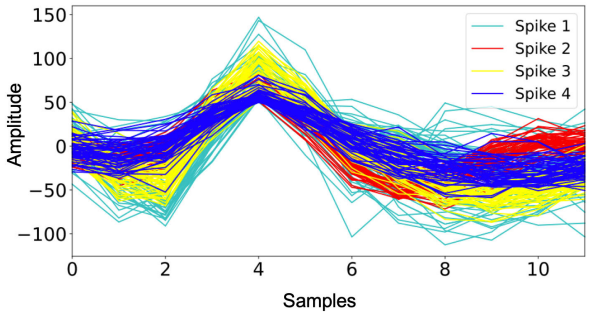


**FIGURE 1. Four distinct color-coded neurons captured from channel #70 inDataset1. These spikes are aligned and sorted using SPC and resortedmanually for accurate performance evaluation purpose.**

has been used in many spike sorting studies [25], [26], [27], [35] and is publicly available on the data-sharing platform of Collaborative Research in Computational Neuroscience [36]. The recordings in Dataset1 show the activity of two, three, four and five neurons that are influenced by recording noise (12 samples per spike; sampling rate: 24.4 kHz). Dataset1 has already been sorted using SPC [15]. Moreover, the spikes were re-sorted manually to minimize potential errors. Ground truth is not available for these recordings, as usual for real recordings. Table 1 indicates the data distribution of Dataset1. As can be seen, the number of firings per neuron is imbalanced, and some neurons only show a small number of spikes (cf. Table 1; Channel #82, Neuron 1). Fig. 1 illustrates the peak-aligned spike waveforms of channel #70 in Dataset1.

#### 2) SYNTHETIC DATA

Dataset2, introduced by Quiroga et al. [15], consists of the four subsets C_Easy1, C_Easy2, C_Difficult1 and C_Difficult2 that have also been widely used by the deep learning-based spike sorting community [23], [25], [26], [27], [35]. Each subset contains four one-minute single-channel recordings that show the activity of three distinct neurons at different noise levels (standard deviations $\sigma_N$ of 5%, 10%, 15% and 20% with respect to the normalized spike amplitudes). Dataset2 was synthesized in the following way: i) based on a pool of 594 spike waveforms derived from the basal ganglia and neocortex from monkey, randomly shifted waveforms were superimposed to mimic a local field potential (LFP); ii) three normalized distinct spike-waveforms of the above-mentioned pool were superimposed with this LFP to imitate neuron activity close the recording probe. The sampling rate of Dataset2 is 24 kHz. In contrast to Dataset1, Dataset2 contains spike overlaps. Due to the synthetic nature of this data, ground truth is available and could be used for model evaluation.

### B. DATA PREPARATION

#### 1) EXPERIMENTAL DATA

As outlined in Table 1, Dataset1 shows strong class imbalance. Using this data, most research groups in the field of deep learning-based spike sorting trained their models with a specific fraction, for example, 50% of all spikes from each

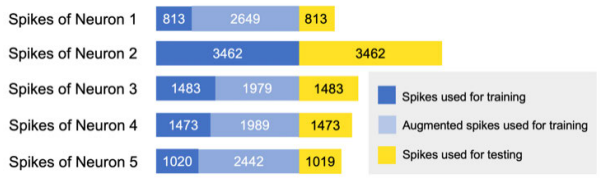


**FIGURE 2.** Data augmentation for dataset1. The presented example corresponds to Channel #90. First, 50% of all spike classes from the active neurons (Neuron 1 to 5) were included in the training set. Then all classes were augmented so that each class contains the most same number of samples. 50% of all spikes from each class were used for testing.

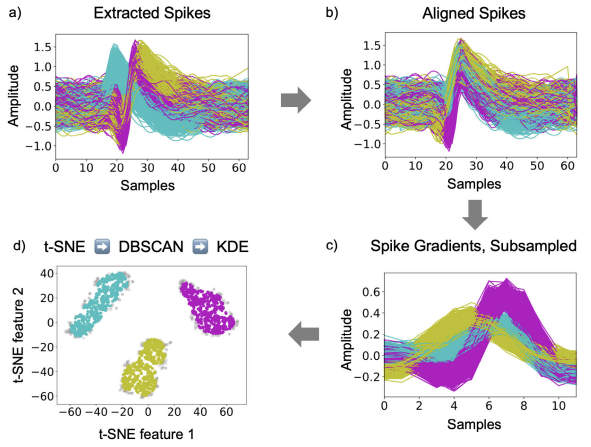


**FIGURE 3.** Fully automatic pre-processing and labelling scheme for dataset2. (a) Spikes from three neurons (cyan, magenta and yellow) are extracted from the signal and (b) aligned to their peaks. Note that overlapping spikes were excluded of this figure for clarity. (c) Gradients are determined and waveforms are reduced from 64 to 12 samples. (d) Features are extracted by using t-SNE and clustered by utilising DBSCAN. 80% of the spikes are then selected based on their density scores estimated with KDE. The neglected spikes are highlighted in grey. Spikes with a length of 12 samples that correspond to the colored examples are included into the training set.

class. This may result in a biased model as it overestimates the importance of classes with many examples. Therefore, this work employs data augmentation, a common best practice in machine learning [37], to re-balance the training sets. The proposed augmentation technique is illustrated in Fig. 2 and works as follows. First, a random example of the respective class was selected and copied. Subsequently, random scaling was applied to the copied spike by using noise from a Gaussian distribution, similar to the data augmentation technique in [35]. This procedure was repeated until the desired number of spikes was reached. The pseudo-labels obtained through the SPC and manual re-sorting were used to train the proposed models.

### 2) SYNTHETIC DATA

As the classes of Dataset2 are almost balanced, data augmentation was not necessary. However, spikes needed to be extracted and prepared for training. Many research groups [25], [26], [27], [28] used the available ground-truth labels to train their supervised deep learning models on Dataset2. Since ground truth is not available in real data, an unsupervised pre-sorting stage is proposed in this work to provide pseudo-labels that can be used for model training. The data preparation and the employed pseudo-labelling technique for Dataset2 are illustrated in Fig. 3. The extracted spikes were aligned to their peaks, as shown in Fig. 3b, to ensure that the proposed model learns the spike waveforms instead of their cluster-specific peak positions. This means that each spike was shifted so that each peak amplitude was situated at sample #26. Subsequently, spike gradients were calculated as shown in Fig. 3c. Spike derivatives have already proven as an effective tool to extract useful features in class separation and diminishing the noise effect [38]. This technique was also used on Dataset2 in [22] to use more substantial spike features. The 64 sample long spikes were then reduced to twelve samples each, as the network input size impacts the processing speed and the required computation. Moreover, this reduced the number of actual overlaps of two spikes with peak-to-peak distances greater than six samples. Each subset was split into a training set (50%) and a testing set (50%). Subsequently, the training data was pre-sorted using t-distributed stochastic neighbor embedding (t-SNE) [39] to extract features and density-based spatial clustering of applications with noise (DBSCAN) [40] was utilized for clustering. T-SNE is a relatively modern dimensionality reduction technique which is increasingly being used in the field of spike sorting [32], [41]. DBSCAN is a common clustering method in spike sorting [22], [42, [43] and often outperforms standard methods like k-means as it does not require specifying the number of clusters, handles noise and outliers effectively, identifies small and arbitrarily shaped clusters, and is robust to variations in initialization and cluster sizes, making it particularly suited for complex neural data. Both t-SNE and DBSCAN rely on hyperparameters that were tuned using a grid-search algorithm. Subset C_Difficult2 with the highest noise level of 20% was used for optimization, resulting in the following hyperparameters: [perplexity = 30; learning rate = 500; epsilon = 8], where perplexity controls the balance between local and global structure preservation, and epsilon defining the maximum distance of two points to be considered as neighbors (same cluster). Finally, Kernel Density Estimation (KDE) [43] was applied to each data point, which reflects its local point density based on the surrounding distribution of spikes in the cluster. By ranking all points in a cluster according to these density scores, the top 80% of spikes—representing the densest regions—were retained. This ensures that the most representative and well-clustered spikes, which are less likely to be noise or artefacts, are used for further processing. This method effectively captures the core structure of irregularly shaped clusters, which improves the quality of the dataset used for training the neural network by focusing on high-density regions while mitigating the impact of outliers due to noise and artefacts. In contrast to this approach, other studies have relied on selecting a fraction of samples closest to the centroid of each cluster [23]. This is not effective for clusters with non-convex shapes, leading to many mislabeled examples that doomed the neural network

used in [23]. The proposed method of this work results in significantly fewer mislabeled spikes which is reflected in better results with an even simpler model. It should be noted that the proposed labeling technique for Dataset2 represents a fully automatic training method without requiring any human intervention.

### C. MODEL SELECTION AND ARCHITECTURE

While DNNs typically require a lot of space for memory, quantization techniques can help to reduce the required memory as already shown with DNN-based spike sorting models [9], [32]. Binarization is the most extreme form of quantization. In 2015, Courbariaux et al. presented BinaryConnect [44] where they trained DNNs with 1-bit weights instead of 32-bit floating point values. This converts multiply-accumulate operations (MACs) to simple accumulations and reduces the overall required system memory. In another work, Courbariaux et al. presented the first BNN model with binarized weights and activations [45]. In this network [45], most calculations are based on bitwise operations including XNOR gates and popcounts. This drastically reduces the computational complexity and makes BNNs good candidates for running on resource-constrained devices. Courbariaux et al. recommended to keep the input and output layers of BNNs on full precision to maintain high accuracy [45]. The forward propagation in BNNs is much more efficient than in full-precision models (FPMs) as it uses the *sign* function (1) for binarization and thus reduces overall complexity. However, the fine-tuning of the model weights (training) implies an increased complexity compared to FPMs. This is due to the fact that the derivative of the *sign* function is zero which makes it impossible to use standard methods like gradient descent for optimization. Therefore, Courbariaux et al. proposed to use the Straight-Through-Estimator (STE), which introduces gradient approximations as expressed in (2) and (3) [45]. For further details of BNNs and the current state of research in this area, reference is made to recently published reviews [46], [47].

$$Sign\,(x) = \begin{cases} 1, & if\ x \geq 0, \\ -1, & otherwise, \end{cases} \tag{1}$$

$$Approx\,(x) = \begin{cases} x, & if\ x \geq -1\ and\ x \leq 1, \\ -1, & if\ x < -1 \\ 1, & otherwise, \end{cases} \tag{2}$$

$$\frac{\delta Approx(x)}{\delta(x)} = \begin{cases} 1, & if\ x \geq -1\ and\ x \leq 1, \\ 0, & otherwise. \end{cases} \tag{3}$$

In this paper, the Python Larq framework [14] in TensorFlow was used to implement the BNN. In addition to the proposed BNN, an equally sized FPM was designed to investigate the performance gap caused by the binarization of the model's weights and activations. Note that the input and output layer units were not binarized in this work and the models were developed without bias terms. Given that the BNNs usually require longer training times due to the nature of binary weight updates and the ineffectiveness of many gradient changes, a higher than usual learning rate of 0.01 was chosen to accelerate convergence and compensate for the slow binary weight changes. The models were trained over 250 epochs using categorical cross-entropy loss and the Adam optimizer [48]. A batch size of 128 was used to speed up training. The BNN was optimised using Channel #125 of Dataset1 as this recording consists of highly correlated spikes with a high level of class imbalance. A grid search was performed to find the optimal network architecture with regard to the number of hidden layers [1], [2], [3] and units per layer [8, 16, 32, 64, 128, 256, 512, 1024]. The model input consists of 12 units which is equal to the number of samples per spike, while the output of the model has $n$ units depending on the number of active neurons in the recording. The Softmax function was used for classification.

Classification models can be evaluated based on several metrics. Classification accuracy, defined in (4), is commonly utilized to quantify the performance. With strongly imbalanced data, like in Dataset1, Accuracy often does not correctly reflect the actual performance of the classification model for minor classes. The $F_1$-Score (5), a more appropriate metric for imbalanced data, determines the harmonic mean of the precision and recall and usually ranges from zero to one. In the following, $F_1$ is always denoted in percent. Precision indicates the ratio of positive classifications that were actually correct, while recall represents the proportion of actual positives that were correctly identified. For multi-class classification problems with class imbalance, Macro-$F_1$ (6) is commonly used as it treats every class with the same importance. Due to the strong imbalance in Dataset1, it was used as the evaluation metric in the described grid search. In (4), (5) and (6), $TP$ denotes true positive predictions, $FP$ represents false positives, $FN$ are false negatives and $C$ denotes the number of classes in the data:

$$Accuracy = \frac{\sum_{j=1}^{C} TP_j}{\sum_{j=1}^{C} TP_j + FN_j}, \tag{4}$$

$$F_1 = \frac{TP}{TP + 0.5(FP + FN)}, \tag{5}$$

$$Macro{-}F_1 = \frac{1}{C} \sum_{j=1}^{C} F_{1j}. \tag{6}$$

The results of the grid search are shown in Fig. 4. As can be seen, models with a single hidden layer achieve neither satisfactory nor stable results. Similar results were observed when evaluating the models with a single hidden layer across alternative channels, hence, the authors of this paper refrained from investigating these models further in this study, as the complexity of the neural data requires more sophisticated and extensive architectures to adequately capture relevant spike features. This is also true for multi-hidden layer models with a small number of hidden units per layer. However, the BNN performs better when the number of units per hidden layer increases. This is due to the fact that more units help to identify precise decision boundaries in the feature space. Since

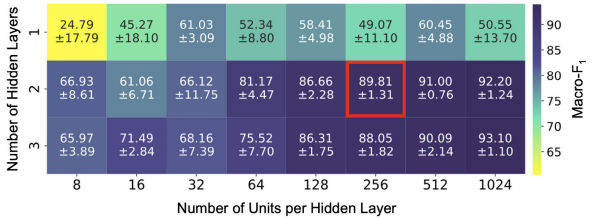


**FIGURE 4.** Evaluation of the neural architecture search. Results of the grid search using a BNN with an equal number of hidden units per hidden layer. The red highlighted parameter combination was used for the experiments in this work.

the performance of the model with three hidden layers is not superior to the model with two hidden layers, the following experiments were carried with a BNN with two hidden layers. The required memory capacity was compared for a varying number of hidden units as shown in Fig. 5. The turning point occurs when the hidden layer consists of 256 neurons, where an acceptable trade-off between memory requirement and Macro-$F_1$ is achieved (cf. Fig. 5). The final model is illustrated in Fig. 6. The proposed BNN requires only 12.44 kB (FPM: 274 kB) of memory. As pointed out by Comon [18], the binarization of model parameters itself can be seen as a form of regularisation. Moreover, dropout [19] and batch normalization (BN) [20] were utilized to stabilize training and minimize overfitting For the sake of simplicity, both of these methods are not illustrated in Fig. 6. Note that the full precision parameters were used for BN. Shift-based BN, which approximates BN nearly without any multiplications, while it maintains the same level of accuracy [45] can be used to keep resources low. Note that 32.16% of the required memory in the proposed model is used to store the parameters for BN. Early stopping was also used for regularisation, meaning that if there was no improvement in validation loss over 25 consecutive epochs, training ceased, and the best model parameters were restored. The primary distinction from prior works, such as [29], lies in the enhanced deep network architecture proposed. The observed improvement in accuracy is a result of this tailored architectural design, rather than simply increasing the network size. In addition, this work introduces specific modifications, e.g., BN or advanced data preprocessing, aimed at improving the network's capacity, further distinguishing it from prior research.

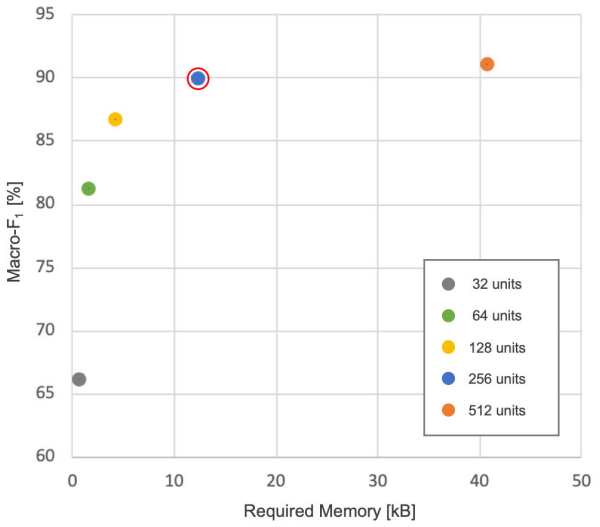


**FIGURE 5.** Memory usage of the BNNs with two hidden layers in comparison to their Macro-F1. Macro-F1 increases only slightly from a value of 256 hidden units per layer whereas the required memory increases exponentially. This is considered as an optimal design point (highlighted in red).

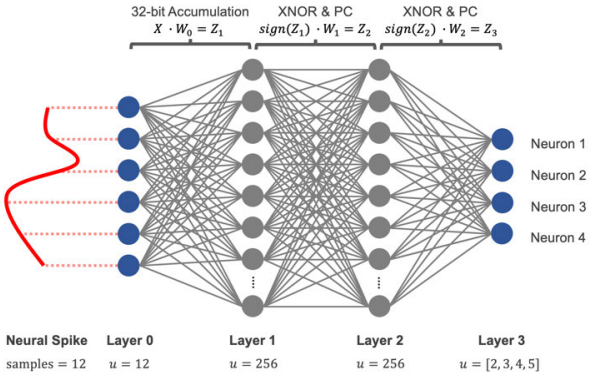


**FIGURE 6.** Proposed architecture. A spike waveform is fed to BNN with 12 full precision units (u). This waveform is then processed by two hidden layers with 256 units. The dot product of the full-precision input X and the binarized weights $W_0$ are simplified to a 32-bit accumulation, resulting in $Z_1$, which is then binarized using the sign function. Subsequently, the XNOR operator is applied to sign($Z_1$) and $W_1$. Popcounting (PC) the result of this operation leads to $Z_2$. The previous step is repeated in the next layer. A Softmax function is applied in the last layer to classify the input into one of two, three, four or five classes that correspond to the underlying neurons in the neural recording channel. The non-binarized elements of the BNN are highlighted in blue.

## III. RESULTS

The derived results from the BNN architecture shown in Fig. 6 using the presented datasets in Section II are presented and discussed in the following sections. Ten experiments were conducted for each recording and the standard deviations were determined to provide statistical results.

### A. PERFORMANCE ON DATASET1

Table 2 shows the classification accuracy and Macro-$F_1$ of the BNN and the equally-sized FPM using Dataset1. The results indicate that the FPM could reach an average accuracy of 97.30% and Macro-$F_1$ of 94.63%. The lightweight BNN reached values of 94.61% and 90.02%, respectively. This is a remarkable result considering that the model compression rate is 5%, with 95.6% of the MACs being binarized. The trade-off between performance and efficiency is shown in Fig. 7 and is clearly in favor of the BNN. While the FPM reaches accuracy of 99.15% and Macro-$F_1$ of 98.87% on certain recordings, including Channel #126 with five active neurons, the BNN could also reach high results of 97.48% and 96.57% on this data. Nevertheless, there are two recordings on which the proposed models were not able to achieve satisfying results. For example, the BNN reached an accuracy of 86.31% and a Macro-$F_1$of 75.34% on channel #68.

**TABLE 2. Accuracy and Macro-F1 achieved with the proposed BNN and FPM using dataset1.**

| Channel ID | Accuracy BNN | Accuracy FPM | Macro-$F_1$ BNN | Macro-$F_1$ FPM |
|---|---|---|---|---|
| 125 | 94.45 ± 0.86 | 97.24 ± 0.45 | 89.51 ± 1.31 | 93.93 ± 0.55 |
| 66 | 95.73 ± 0.55 | 98.54 ± 0.13 | 95.04 ± 0.58 | 98.40 ± 0.27 |
| 69 | 97.65 ± 0.19 | 98.66 ± 0.15 | 97.62 ± 0.19 | 98.64 ± 0.22 |
| 77 | 95.49 ± 0.67 | 97.65 ± 0.12 | 92.00 ± 0.90 | 95.66 ± 0.23 |
| 79 | 92.66 ± 0.95 | 96.86 ± 0.32 | 89.57 ± 1.10 | 95.39 ± 0.54 |
| 84 | 97.25 ± 1.17 | 98.91 ± 0.56 | 96.14 ± 1.30 | 98.42 ± 0.76 |
| 87 | 97.31 ± 0.32 | 98.66 ± 0.10 | 80.55 ± 1.52 | 88.12 ± 0.17 |
| 91 | 93.63 ± 0.69 | 96.60 ± 0.30 | 92.20 ± 0.86 | 95.91 ± 0.36 |
| 92 | 95.58 ± 0.61 | 98.13 ± 0.23 | 91.68 ± 1.02 | 95.97 ± 0.43 |
| 94 | 94.36 ± 0.83 | 96.21 ± 0.33 | 90.07 ± 1.49 | 93.44 ± 0.45 |
| 98 | 95.88 ± 0.62 | 97.21 ± 0.55 | 83.45 ± 1.37 | 86.12 ± 0.87 |
| 99 | 95.03 ± 0.49 | 97.54 ± 0.45 | 85.31 ± 1.43 | 91.83 ± 0.60 |
| 100 | 92.68 ± 0.97 | 97.58 ± 0.25 | 89.16 ± 1.38 | 96.19 ± 0.40 |
| 101 | 94.03 ± 1.47 | 97.59 ± 0.19 | 89.23 ± 1.88 | 94.79 ± 0.22 |
| 108 | 94.51 ± 0.77 | 96.86 ± 0.13 | 93.14 ± 0.80 | 95.95 ± 0.13 |
| 109 | 92.70 ± 1.13 | 96.31 ± 0.32 | 91.44 ± 1.15 | 95.59 ± 0.43 |
| 112 | 93.80 ± 1.15 | 94.77 ± 0.53 | 90.16 ± 1.20 | 92.67 ± 0.78 |
| 114 | 94.52 ± 1.56 | 97.91 ± 0.13 | 92.91 ± 1.58 | 97.09 ± 0.18 |
| 115 | 97.22 ± 0.18 | 98.62 ± 0.10 | 90.50 ± 0.98 | 94.38 ± 0.14 |
| 122 | 96.28 ± 0.51 | 98.04 ± 0.11 | 90.36 ± 0.90 | 94.37 ± 0.15 |
| 124 | 94.38 ± 0.50 | 97.04 ± 0.20 | 88.24 ± 1.07 | 92.65 ± 0.23 |
| 67 | 96.07 ± 0.65 | 97.22 ± 0.12 | 94.26 ± 0.91 | 96.20 ± 0.33 |
| 68 | 86.31 ± 2.53 | 97.18 ± 0.33 | 75.34 ± 3.52 | 93.86 ± 0.59 |
| 70 | 94.56 ± 0.43 | 97.02 ± 0.12 | 91.46 ± 0.73 | 95.11 ± 0.17 |
| 83 | 93.47 ± 0.50 | 97.12 ± 0.21 | 92.06 ± 0.45 | 95.97 ± 0.26 |
| 95 | 98.76 ± 0.11 | 99.50 ± 0.04 | 91.16 ± 0.41 | 95.78 ± 0.29 |
| 107 | 92.61 ± 1.06 | 96.60 ± 0.35 | 89.33 ± 1.20 | 95.32 ± 0.36 |
| 116 | 97.56 ± 0.41 | 98.54 ± 0.13 | 96.45 ± 0.54 | 97.99 ± 0.15 |
| 82 | 92.65 ± 0.65 | 92.65 ± 0.23 | 79.12 ± 1.78 | 79.12 ± 0.47 |
| 90 | 88.34 ± 0.75 | 94.38 ± 0.45 | 86.66 ± 0.96 | 86.66 ± 0.51 |
| 126 | 97.48 ± 0.13 | 99.15 ± 0.08 | 96.57 ± 0.15 | 98.87 ± 0.11 |
| **Average** | **94.61 ± 0.76** | **97.30 ± 0.25** | **90.02 ± 1.12** | **94.21 ± 0.37** |

Fig. 8a shows the confusion matrix that was obtained in an experiment where the BNN was applied to Channel #68. The model performed well in classifying Neurons 2, 3 and 4 but it overpredicts spikes of Neuron 1. This resulted in an $F_1$ drop of Neuron 1 to 37.64%. The reason for this may be the high waveform similarity of spikes from Neurons 1, 2 and 4 in this recording. For such scenarios, it is beneficial to use the FPM (accuracy of 97.18% and Macro-$F_1$ of 93.84%), as the binarized parameters of the model are not sufficient to distinguish highly correlated spikes. Channel #82 is the other recording where the BNN did not yield satisfying results. The respective confusion matrix is displayed in Fig. 8b. As can be seen, there are only 38 examples of Neuron 1 in the test set (cf. Table 1: Channel #82, 50% of N1), implying that only a few examples could be used to augment the training set for this class. This was not sufficient to create a representative training set, resulting in a relatively large number of FPs and FNs for this class. The value of Macro-$F_1$ is low for this channel because $F_1$ is only 51.11% for class 1. In addition, many spikes of Neuron 2 are predicted as spikes of Neuron 5, as these waveforms appear to be very similar, resulting in an $F_1$ of 61.79% for class five.

### B. PERFORMANCE ON DATASET2

Table 3 shows the model performance on the synthetic data. The respective noise level of each recording is denoted in the last part of the Channel ID, e.g., C_Easy1_05 refers to a noise level of 5%. In addition to the results of the neural networks, the performance of the proposed labeling technique is also shown. Applied to the test data, an average accuracy of 98.97% was achieved using t-SNE and DBSCAN. The FPM slightly outperformed this performance with an accuracy of 99.07%, whereas the BNN still reached a remarkable accuracy of 98.39%. The performance gap between the two neural networks is only 0.68%, which is much smaller than for Dataset1 where the performance gap for accuracy and Macro-$F_1$ is 2.81% and 4.61%, respectively. Moreover, it can be observed that the model performance differs between the subsets C_Easy∗ and C_Difficult∗, with the latter showing slightly lower results. This is attributed to the higher prevalence of overlapping spikes and the greater similarity of spike waveforms among different neurons in C_Difficult∗. The proposed models of this work and the results are discussed in the following.

## IV. DISCUSSION

### A. PERFORMANCE ANALYSIS

The obtained results in Dataset2 are significantly higher than those achieved with Dataset1. In addition, the performance gap between the BNN and FPM is bigger for Dataset1. In this regard, it should be emphasized that it is possible to achieve higher results on Dataset1 when using a network with more

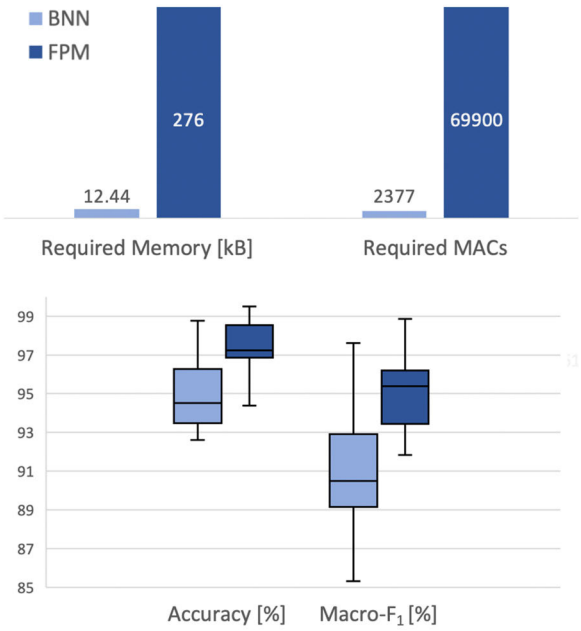


**FIGURE 7.** Comparison of the required memory and full-precision MACs ofthe BNN and FPM (top) and quantitative metrics (bottom). Compared to the FPM, the BNN takes a small fraction of the required memory and full precision MACs while maintaining high accuracy and Macro F1. All results.

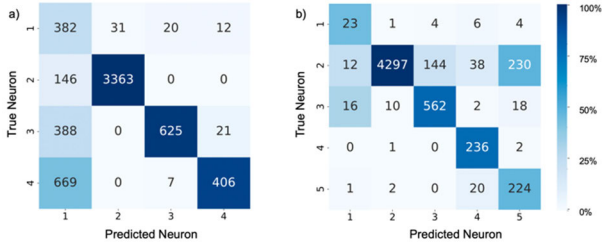


**FIGURE 8.** Confusion matrices: Results of the BNN applied on Channel #68 (a) and #82 (b). Accuracy is on an acceptable level as the TPs of the dominant classes are high in absolute numbers. However, Macro-F1 is significantly lower as it treats every class with the same weight, showcasing that the BNN struggles to identify spikes from certain neurons. The colormap considers each column individually instead of the matrix as a whole to emphasise precision.

hidden units (cf. Fig. 4 and 5). For Dataset2, the model performance is closer to convergence with the proposed architecture. As the dimensions of the model increase and the accuracy converges towards the maximum achievable value, the performance gap between FPM and BNN decreases. Moreover, the spike waveforms of the experimental dataset often show a higher inter-class correlation, making them more difficult to distinguish. Furthermore, the spikes of Dataset1 are contaminated with real noise, which makes signal processing more challenging compared to spikes from Dataset2, where noise is solely simulated by the superposition of attenuated spikes. For Dataset2, the selected spikes utilizing the approach explained in Section II-B demonstrate a minimal error rate, culminating in a robust training set containing a small number of misclassified instances. However, the accuracy of the labels for the experimental recordings relies on

**TABLE 3.** Classification accuracy achieved using dataset2.

| Channel | t-SNE + DBSCAN | BNN | FPM |
|---|---|---|---|
| C_Easy1_05 | 99.32 ± 0.06 | 98.63 ± 0.78 | 99.00 ± 0.12 |
| C_Easy1_10 | 99.49 ± 0.03 | 99.01 ± 0.25 | 99.50 ± 0.04 |
| C_Easy1_15 | 99.54 ± 0.02 | 98.90 ± 0.13 | 99.53 ± 0.06 |
| C_Easy1_20 | 99.48 ± 0.01 | 99.19 ± 0.39 | 99.29 ± 0.12 |
| C_Easy2_05 | 98.71 ± 0.06 | 98.06 ± 0.62 | 99.24 ± 0.13 |
| C_Easy2_10 | 99.43 ± 0.01 | 97.94 ± 0.21 | 99.41 ± 0.06 |
| C_Easy2_15 | 99.30 ± 0.02 | 98.38 ± 0.17 | 99.30 ± 0.04 |
| C_Easy2_20 | 99.26 ± 0.05 | 98.83 ± 0.50 | 99.16 ± 0.09 |
| C_Difficult1_05 | 98.93 ± 0.07 | 98.20 ± 0.77 | 98.80 ± 0.09 |
| C_Difficult1_10 | 98.78 ± 0.10 | 98.65 ± 0.43 | 99.07 ± 0.06 |
| C_Difficult1_15 | 98.67 ± 0.07 | 97.90 ± 0.20 | 98.55 ± 0.05 |
| C_Difficult1_20 | 98.36 ± 0.05 | 97.73 ± 0.33 | 98.69 ± 0.16 |
| C_Difficult2_05 | 98.63 ± 0.05 | 97.74 ± 0.75 | 98.94 ± 0.13 |
| C_Difficult2_10 | 98.67 ± 0.04 | 98.99 ± 0.30 | 99.30 ± 0.07 |
| C_Difficult2_15 | 98.89 ± 0.06 | 98.27 ± 0.35 | 98.74 ± 0.08 |
| C_Difficult2_20 | 97.99 ± 0.05 | 97.82 ± 0.22 | 98.64 ± 0.09 |
| **Average** | **98.97 ± 0.05** | **98.39 ± 0.40** | **99.07 ± 0.09** |

SPC and manual re-sorting and remains unverifiable, due to missing ground truth. The significant spike correlation indicates the potential for Dataset1 to possess a higher labelling error rate compared to Dataset2, thereby posing a greater challenge for the classification of spikes by the model as labelling errors may have a negative impact on the model performance. At this point, it must be mentioned that the performance on Dataset1 is relative to the labeling method mentioned above and the results are therefore not entirely conclusive. The ensuing discussion examines specific challenges related to spike sorting that the proposed BNN has been able to effectively address, as well as those that need to be given greater focus in the future.

### 1) NOISE RESISTANCE

The BNN exhibited no significant decline in performance despite the heightened noise levels in Dataset2, which is beneficial for the use on neural data that is often contaminated with high noise from various sources such as the recording device or the signals of more distant neurons.

### 2) OVERLAPPING SPIKES

It was noted that temporal spikes with stronger phase shifts do not disrupt the network. Nonetheless, an examination of the network's errors indicated that a considerable number of misclassified instances stemmed from overlapping spikes with minimal peak-to-peak intervals. The superposition of these waveforms generates an unfamiliar pattern for the network, leading to a somewhat arbitrary classification into one of the different classes.

### 3) SCALABILITY

Scaling is another important challenge in spike sorting, as the number of electrodes used in electrophysiology is growing continuously. At this point, the proposed model in this work is tailored for single-channel data. High-density multi-electrode

arrays (HD-MEAs) may require more efficient models for resource-adequate spike sorting. Extending the suggested framework to incorporate the processing of numerous channels, and the additional utilization of spatial spike data, possesses the capability to enhance not solely the rate of data processing, but also the accuracy and therefore the applicability in practical scenarios for iBCIs.

#### 4) TRANSFERABILITY AND REPRODUCIBILITY

The model transferability could be shown with high results across datasets. Reproducibility is ensured by making the proposed model open source. The model can be found using the following link: www.github.com/LucaMMeyer/BNN.

### B. COMPARISON TO OTHER WORK

A straightforward tabular comparison of model performance with other state-of-the-art models has been omitted owing to considerable variations in training set sizes, foundational assumptions, and evaluation parameters across different studies. Such discrepancies can result in erroneous conclusions if not adequately contextualized. Rather, comprehensive descriptions and metrics are presented in this section to enable readers to assess the models within their specific contexts, aiming to allow a fair comparison.

#### 1) DATSET1

In 2020, Li et al. [25] proposed a 1D-CNN for spike classification which was also evaluated on Dataset1. This model contains four convolutional layers, 2 max pooling layers and a fully connected network (FCN) with 300, 100 hidden units and *two to five* units. The FCN, which only represents the output of the model in [25], has a comparable size to the entire network proposed in this paper. The model of Li et al. [25], due to its heavy architecture, certainly relies on external processing, however, the models proposed in this paper can compete with the performance of the heavy model in [25]. While the FPM in this work takes 139,267 FLOPs to process a neural waveform, the CNN in [25] requires 5,788,250 FLOPs [24]. Using Dataset1, Li et al. [25] reached an average accuracy of 96.53% and Macro-$F_1$ of 95.68% using 50% of the available spikes, while not testing their model on Channel #87 and #90. Averaged over the same channels, the proposed FPM achieved values of 97.48% and 94.89% respectively, showcasing the proposed data augmentation technique can compensate for a much more complex network architecture in [25].

In 2023, Wang et al. [28] proposed a model for spike classification consisting of an LSTM layer, two 2D-convolutional layers and a fully connected network at the end of the network. Applied to channels #66, #69, #98, #115, #68, #70, #83, #95, #82 and #126 of Dataset1, Wang et al. [28] achieved an average accuracy of 94.77% using 50% of the spikes for training. Both models presented in this work (BNN: 94.97%; FPM: 97.93%) outperformed the model presented by Wang et al. [28].

While the authors of [28] did not try to come up with a solution regarding the class imbalance issue of Dataset1, in 2023, Li et al. [27] presented a deep reinforcement learning approach targeting this issue. They trained a digital agent called 'ImbSorter' to classify spikes using a dynamic reward function that pays higher reward to the model for classifying rare classes. The proposed model is a convolutional network with two 2D-convolutional layers, a max pooling layer and a fully connected network at the network output. Evaluated on Dataset1, Imbsorter achieved an average accuracy of 97.9% and an average Macro-$F_1$ of 95.8%. Despite its significantly higher complexity, it performed only slightly better than the proposed FPM.

#### 2) DATSET2

In 2019, Park et al. [23] proposed an approach based on PCA and the k-means to obtain pseudo labels which they used to train their deep learning model. After the clustering stage, they selected examples that were located closest to the centroids in the feature space in order to minimize labeling errors. They trained a fully connected network with four hidden layers (256 units each) with the labeled spikes and reached an overall accuracy of 94.09%. The proposed BNN and FPM reached accuracies of 98.39% and 99.07%, clearly outperforming the model from Park et al. However, the better performance achieved in this work is mainly associated to the use of spike gradients and the proposed pseudo-labeling technique (t-SNE + DBSCAN + KDE) which generates pseudo-labels with much lower error rates.

Li et al. [25], Wang et al. [28] and Li et al. [27] also evaluated their models using Dataset2, but trained their models with ground-truth labels. However, the extracted spikes from the synthetic recordings were not aligned in [27] before they were included in the training set, which may have led to falsely promising results [24]. Moreover, in [27], classes were handled slightly different as overlapping spikes were treated as separate classes, resulting in an average accuracy of 98.4%. Nevertheless, these studies were used for comparison in order to compare the performance of the models presented in this work with more complex deep learning methods. Using the sixteen recordings in Dataset2 and utilizing 50% of the available spikes for training, Li et al. [25] reached an average accuracy of 99.32%. By using 40% of the spikes for training the model in [28], Wang et al. achieved an average accuracy of 99.5%. With an accuracy of 99.07%, the performance of the FPM in this work is just slightly below the models mentioned, even though the FPM has a much smaller size and was trained with pseudo-labels instead of ground truth. It is mentioned in Section I that Valencia and Mohammad applied their BNN [29] and PBNN [30] to Dataset2 as well and achieved an average accuracy of up to 90%. By using an additional layer, more units per layer and BN, the BNN in this work reduces the performance gap of [29] and [30] to the more advanced deep learning models mentioned above on less than 1%.

Given the advantages of the proposed BNN over the proposed FPM, and thus all other deep learning models mentioned in Section IV-B, it becomes clear why the proposed BNN is preferable in the context of iBCIs:

1. Small chip area
2. Low implementation cost
3. Moderate energy use
4. Robust performance

Chip area, a critical constraint in iBCIs, is significantly reduced in the proposed BNN compared to FPMs. While this claim is currently theoretical—rooted in the BNN's lower computational complexity—ongoing work on an FPGA-based implementation, to be published soon, will substantiate it. Moreover, the BNN incurs lower implementation costs, requiring reduced memory and fewer, less complex mathematical operations, as illustrated in Fig. 7. Consequently, it also offers lower power consumption. Although BNNs from [29], [30] excel in these aspects, the deep BNN proposed here achieves significantly higher accuracy, enabling its application to more complex neural datasets.

### C. HARDWARE CONSIDERATIONS

The proposed BNN is specifically tailored for resource-constrained environments, such as iBCIs. Several key features of the BNN model directly address common hardware limitations and are discussed in the following.

#### 1) ADAPTIVE MODEL ARCHITECTURE

The relationship between model size and performance, as illustrated in Fig. 5, provides a clear guideline for hardware implementation, allowing for precise model tuning to make use of available resources while maximizing performance. In scenarios with limited memory, a smaller model can be selected with a predictable trade-off in accuracy. This flexibility ensures that the proposed models can be adapted to a wide range of hardware configurations, from highly constrained implantable devices to more capable external processors, where the FPM may be preferred over the BNN architecture.

#### 2) MEMORY AND COMPUTATIONAL EFFICIENCY

The proposed BNN requires only 12.44 kB of memory, which is a 95% reduction compared to the equally-sized FPM. This dramatic decrease in memory is crucial for iBCI applications. The compact memory requirement not only saves physical space in the implantable device but also reduces power consumption associated with memory access operations, which can be a significant factor in overall energy usage. The binarization of weights and activations in the BNN model directly translates to bitwise operations in hardware. This means that spike sorting can be performed using simple logic gates and accumulators rather than complex floating-point operations, drastically reducing both the silicon area required and the power consumed per operation. With 95.6% of MACs being binarized, the BNN achieves substantial computational efficiency. This allows faster processing times, enabling real-time spike sorting even in resource-constrained environments. As proposed in [49], during inference, the computationally expensive Softmax function can be replaced with Hardmax (argmax). Hardmax can be used because Softmax is a monotonic function, or in other words, it maintains the order of the input values. This further optimizes hardware implementation without sacrificing classification accuracy. The combination of reduced memory, simplified computations, and flexible architecture allows for efficient real-time processing of neural signals, enabling more sophisticated on-chip analysis in implantable neural interfaces. As shown in previous studies [9], models with milder quantization can also be used for spike sorting and achieve good efficiency. However, such mild quantization, to e.g. 4-bit, may be sufficient for smaller systems, but can become a significant bottleneck when scaling up to larger systems. iBCIs with a large number of channels require minimal resource consumption due to strict power and thermal constraints. In these scenarios, small binarized models offer significant advantages over medium-sized networks with mild quantization, as they allow a significant reduction in energy and area with little loss of accuracy.

### D. LIMITATIONS AND OUTLOOK

The proposed BNN, while effective, presents several limitations. First, it is designed for single-channel processing, restricting its scalability. Additionally, the model struggles with overlapping spikes, particularly those with minimal peak-to-peak intervals, as the resulting superimposed waveforms lead to higher misclassification rates. The BNN's performance is also influenced by the accuracy of pseudo-labels, which can introduce errors and affect overall classification accuracy. Finally, although the model shows robustness to both simulated and real noise, its generalization to more advanced signal complexities, such as biological artifacts or electromagnetic interference, requires further investigation. In general, it must be emphasized at this point that supervised models must always be trained on the respective recording to achieve optimal performance. Even if this is the case, certain signal complexities can pose problems for the model, such as infrequently firing neurons whose spikes were not part of the training set.

To address these challenges and elevate BNN-based spike sorting to a more advanced status, prospective research should concentrate on three principal domains: model adaptivity, hardware implementation and multi-channel processing for large-scale HD-MEAs. For hardware, the BNN will be optimized for FPGAs or ASICs, emphasizing memory efficiency, reduced power consumption, and low-latency operation. Techniques like weight pruning will further minimize memory usage, while replacing Softmax with Hardmax will streamline computations and reduce inference latency. In parallel, the framework should be extended to process multi-channel data from HD-MEAs using a binarized convolutional neural network (BCNN) designed for

spatiotemporal feature extraction. This BCNN will leverage spatial correlations across electrodes and temporal dynamics to enhance spike classification. Additionally, adaptive mechanisms could be explored to dynamically allocate computational resources based on channel activity, improving accuracy and efficiency. These advancements aim to create a scalable, energy-efficient spike sorting solution suitable for next-generation neural interfaces.

## REFERENCES

[1] M. A. L. Nicolelis, "Actions from thoughts," *Nature*, vol. 409, no. 6818, pp. 403–407, Jan. 2001, doi: 10.1038/35053191.
[2] T. W. Berger, A. Ahuja, S. H. Courellis, S. A. Deadwyler, G. Erinjippurath, G. A. Gerhardt, G. Gholmieh, J. J. Granacki, R. Hampson, M. C. Hsaio, J. Lacoss, V. Z. Marmarelis, P. Nasiatka, V. Srinivasan, D. Song, A. R. Tanguay, and J. Wills, "Restoring lost cognitive function," *IEEE Eng. Med. Biol. Mag.*, vol. 24, no. 5, pp. 30–44, Sep. 2005, doi: 10.1109/MEMB.2005.1511498.
[3] C. Clément, *Brain-Computer Interface Technologies: Accelerating Neuro-Technology for Human Benefit*. Cham, Switzerland: Springer, 2019.
[4] Z.-P. Zhao, C. Nie, C.-T. Jiang, S.-H. Cao, K.-X. Tian, S. Yu, and J.-W. Gu, "Modulating brain activity with invasive brain–computer interface: A narrative review," *Brain Sci.*, vol. 13, no. 1, p. 134, Jan. 2023, doi: 10.3390/brainsci13010134.
[5] E. Musk and Neuralink, "An integrated brain-machine interface platform with thousands of channels," *J. Med. Internet Res.*, vol. 21, no. 10, Oct. 2019, Art. no. e16194, doi: 10.2196/16194.
[6] D. Valencia and A. Alimohammad, "Towards in vivo neural decoding," *Biomed. Eng. Lett.*, vol. 12, no. 2, pp. 185–195, Feb. 2022, doi: 10.1007/s13534-022-00217-z.
[7] D. Valencia, J. Thies, and A. Alimohammad, "Frameworks for efficient brain-computer interfacing," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1714–1722, Dec. 2019, doi: 10.1109/TBCAS.2019.2947130.
[8] L. Kong, Z. Zhang, S. Yu, and J. Mao, "An intracortical wireless bidirectional brain-computer interface with high data density," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2024, pp. 1–5, doi: 10.1109/iscas58744.2024.10558134.
[9] C. Seong, W. Lee, and D. Jeon, "A multi-channel spike sorting processor with accurate clustering algorithm using convolutional autoencoder," *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 6, pp. 1441–1453, Dec. 2021, doi: 10.1109/TBCAS.2021.3134660.
[10] M. Zamani, D. Jiang, and A. Demosthenous, "An adaptive neural spike processor with embedded active learning for improved unsupervised sorting accuracy," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 3, pp. 665–676, Jun. 2018, doi: 10.1109/TBCAS.2018.2825421.
[11] M. Zamani, J. Sokolic, D. Jiang, F. Renna, M. R. D. Rodrigues, and A. Demosthenous, "Accurate, very low computational complexity spike sorting using unsupervised matched subspace learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 221–231, Apr. 2020, doi: 10.1109/TBCAS.2020.2969910.
[12] R. Q. Quiroga, L. Reddy, C. Koch, and I. Fried, "Decoding visual inputs from multiple neurons in the human temporal lobe," *J. Neurophysiol.*, vol. 98, no. 4, pp. 1997–2007, Oct. 2007, doi: 10.1152/jn.00125.2007.
[13] S. Gibson, J. W. Judy, and D. Markovic, "Spike sorting: The first step in decoding the brain: The first step in decoding the brain," *IEEE Signal Process. Mag.*, vol. 29, no. 1, pp. 124–143, Jan. 2012, doi: 10.1109/MSP.2011.941880.
[14] C. Gold, D. A. Henze, C. Koch, and G. Buzsáki, "On the origin of the extracellular action potential waveform: A modeling study," *J. Neurophysiol.*, vol. 95, no. 5, pp. 3113–3128, May 2006, doi: 10.1152/jn.00979.2005.
[15] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004, doi: 10.1162/089976604774201631.
[16] K. Hwan Kim and S. June Kim, "Neural spike sorting under nearly 0-dB signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 10, pp. 1406–1411, Oct. 2000, doi: 10.1109/10.871415.
[17] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Phil. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, Nov. 1901, doi: 10.1080/14786440109462720.
[18] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, Apr. 1994, doi: 10.1016/0165-1684(94)90029-9.
[19] X. Geng, G. Hu, and X. Tian, "Neural spike sorting using mathematical morphology, multiwavelets transform and hierarchical clustering," *Neurocomputing*, vol. 73, nos. 4–6, pp. 707–715, Jan. 2010, doi: 10.1016/j.neucom.2008.11.034.
[20] F. Franke, R. Quian Quiroga, A. Hierlemann, and K. Obermayer, "Bayes optimal template matching for spike sorting—Combining Fisher discriminant analysis with optimal filtering," *J. Comput. Neurosci.*, vol. 38, no. 3, pp. 439–459, Jun. 2015, doi: 10.1007/s10827-015-0547-7.
[21] C. O. Okreghe, M. Zamani, and A. Demosthenous, "A deep neural network-based spike sorting with improved channel selection and artefact removal," *IEEE Access*, vol. 11, pp. 15131–15143, 2023, doi: 10.1109/ACCESS.2023.3242643.
[22] J. Eom, I. Y. Park, S. Kim, H. Jang, S. Park, Y. Huh, and D. Hwang, "Deep-learned spike representations and sorting via an ensemble of auto-encoders," *Neural Netw.*, vol. 134, pp. 131–142, Feb. 2021, doi: 10.1016/j.neunet.2020.11.009.
[23] I. Y. Park, J. Eom, H. Jang, S. Kim, S. Park, Y. Huh, and D. Hwang, "Deep learning-based template matching spike classification for extracellular recordings," *Appl. Sci.*, vol. 10, no. 1, p. 301, Dec. 2019, doi: 10.3390/app10010301.
[24] L. M. Meyer, M. Zamani, J. Rokai, and A. Demosthenous, "Deep learning-based spike sorting: A survey," *J. Neural Eng.*, vol. 21, no. 6, Nov. 2024, Art. no. 061003, doi: 10.1088/1741-2552/ad8b6c.
[25] Z. Li, Y. Wang, N. Zhang, and X. Li, "An accurate and robust method for spike sorting based on convolutional neural networks," *Brain Sci.*, vol. 10, no. 11, p. 835, Nov. 2020, doi: 10.3390/brainsci10110835.
[26] M. Liu, J. Feng, Y. Wang, and Z. Li, "Classification of overlapping spikes using convolutional neural networks and long short term memory," *Comput. Biol. Med.*, vol. 148, Sep. 2022, Art. no. 105888, doi: 10.1016/j.compbiomed.2022.105888.
[27] S. Li, Z. Tang, L. Yang, M. Li, and Z. Shang, "Application of deep reinforcement learning for spike sorting under multi-class imbalance," *Comput. Biol. Med.*, vol. 164, Sep. 2023, Art. no. 107253, doi: 10.1016/j.compbiomed.2023.107253.
[28] M. Wang, L. Zhang, H. Yu, S. Chen, X. Zhang, Y. Zhang, and D. Gao, "A deep learning network based on CNN and sliding window LSTM for spike sorting," *Comput. Biol. Med.*, vol. 159, Jun. 2023, Art. no. 106879, doi: 10.1016/j.compbiomed.2023.106879.
[29] D. Valencia and A. Alimohammad, "Neural spike sorting using binarized neural networks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 206–214, 2021, doi: 10.1109/TNSRE.2020.3043403.
[30] D. Valencia and A. Alimohammad, "Partially binarized neural networks for efficient spike sorting," *Biomed. Eng. Lett.*, vol. 13, no. 1, pp. 73–83, Feb. 2023, doi: 10.1007/s13534-022-00255-7.
[31] M. Zamani and A. Demosthenous, "Feature extraction using extrema sampling of discrete derivatives for spike sorting in implantable upper-limb neural prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 716–726, Jul. 2014, doi: 10.1109/TNSRE.2014.2309678.
[32] J. Yi, J. Xu, E. Chen, M. Chamanzar, and V. Chen, "Multichannel many-class real-time neural spike sorting with convolutional neural networks," *IEEE Open J. Circuits Syst.*, vol. 3, pp. 168–179, 2022, doi: 10.1109/OJCAS.2022.3184302.
[33] C. C. J. Chu, P. F. Chien, and C. P. Hung, "Multi-electrode recordings of ongoing activity and responses to parametric stimuli in macaque V1," CRCNS, Tech. Rep., 2014, doi: 10.6080/K0J1012K. [Online]. Available: https://crcns.org/data-sets/vc/pvc-5/about
[34] C. C. J. Chu, P. F. Chien, and C. P. Hung, "Tuning dissimilarity explains short distance decline of spontaneous spike correlation in macaque V1," *Vis. Res.*, vol. 96, pp. 113–132, Mar. 2014, doi: 10.1016/j.visres.2014.01.008.
[35] L. M. Meyer, F. Samann, and T. Schanze, "DualSort: Online spike sorting with a running neural network," *J. Neural Eng.*, vol. 20, no. 5, Oct. 2023, Art. no. 056031, doi: 10.1088/1741-2552/acfb3a.
[36] J. L. Teeters, K. D. Harris, K. J. Millman, B. A. Olshausen, and F. T. Sommer, "Data sharing for computational neuroscience," *Neuroinformatics*, vol. 6, no. 1, pp. 47–55, Mar. 2008, doi: 10.1007/s12021-008-9009-y.
[37] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, Dec. 2022, Art. no. 100258, doi: 10.1016/j.array.2022.100258.

[38] Z. Yang, Q. Zhao, and W. Liu, "Improving spike separation using waveform derivatives," *J. Neural Eng.*, vol. 6, no. 4, Aug. 2009, Art. no. 046006, doi: 10.1088/1741-2560/6/4/046006.
[39] L. V. D. Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, Jan. 2008.
[40] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Portland, Oregon, Aug. 1996, pp. 226–231.
[41] M. H. Nadian, S. Karimimehr, J. Doostmohammadi, A. Ghazizadeh, and R. Lashgari, "A fully automated spike sorting algorithm using t-distributed neighbor embedding and density based clustering," *bioRxiv*, Sep. 2018, doi: 10.1101/418913.
[42] A. Markanday, J. Bellet, M. E. Bellet, J. Inoue, Z. M. Hafed, and P. Thier, "Using deep neural networks to detect complex spikes of cerebellar Purkinje cells," *J. Neurophysiol.*, vol. 123, no. 6, pp. 2217–2234, Jun. 2020, doi: 10.1152/jn.00754.2019.
[43] S. Węglarczyk, "Kernel density estimation and its application," in *Proc. ITM Web Conf.*, vol. 23, 2018, p. 37.
[44] M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Proc. 29th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Curran Associates, 2015, pp. 3123–3131. Accessed: Jul. 10, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/hash/3e15cc11f979ed25912dff5b0669f2cd-Abstract.html
[45] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1," 2016, *arXiv:1602.02830*.
[46] T. Simons and D.-J. Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, p. 661, Jun. 2019, doi: 10.3390/electronics8060661.
[47] R. Sayed, H. Azmi, H. Shawkey, A. H. Khalil, and M. Refky, "A systematic literature review on binary neural networks," *IEEE Access*, vol. 11, pp. 27546–27578, 2023.
[48] D. P. Kingma and L. J. Ba. (2015). *Adam: A Method for Stochastic Optimization*. Accessed: May 15, 2024. [Online]. Available: https://dare.uva.nl/search?identifier=a20791d3-1aff-464a-8544-268383c33a75
[49] E. Zacharelos, C. Scognamillo, E. Napoli, and D. Gragnaniello, "On-chip spike detection and classification using neural networks and approximate computing," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2023, pp. 1–5, doi: 10.1109/biocas58349.2023.10388854.

**LUCA M. MEYER** received the B.Eng. degree in mechanical engineering from the RheinMain University of Applied Sciences, Wiesbaden, Germany, in 2020, and the M.Sc. degree in biomedical engineering from Technische Hochschule Mittelhessen, Giessen, Germany, in 2023.

He currently works as an Independent Researcher. He is the author of four peer-reviewed research articles in the field of spike sorting, with a focus on machine learning-based techniques. His latest article "Deep learning-based spike sorting: A survey" was accepted by the *Journal of Neural Engineering* recently.

**MAJID ZAMANI** (Member, IEEE) received the Ph.D. degree from University College London (UCL), London, U.K., in 2017, with outstanding contribution in implantable brain decoding.

He was a Postdoctoral Research Fellow with the Analog and Biomedical Electronics Group, UCL, from 2017 to 2022. He is currently a Lecturer (Assistant Professor) with the School of Electronics and Computer Science, University of Southampton, U.K. He has extensive experience in design and fabrication of novel computational platforms for biomedical applications. His research interests include advanced biomedical signal and image processing algorithms using variety of AI techniques, augmented navigation in body corridors and canals with extremely limited access, computing frameworks for implantable applications, and miniaturization of low-power and real-time processors for implantable brain machine interfacing (iBMI) in 180/90/45 nm CMOS technologies. He has published 42 articles in journals (IEEE TRANSACTIONS ON MEDICAL IMAGING, IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, and IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS) and international conference proceedings (ESSCIRC). He was a recipient of the Overseas Research Scholarship and the UCL Graduate Research Scholarship to pursue the Ph.D. degree. He was also a recipient of the Best Researcher M.Sc. Student Award.

**ANDREAS DEMOSTHENOUS** (Fellow, IEEE) received the B.Eng. degree in electrical and electronic engineering from the University of Leicester, Leicester, U.K., in 1992, the M.Sc. degree in telecommunications technology from Aston University, Birmingham, U.K., in 1994, and the Ph.D. degree in electronic and electrical engineering from University College London (UCL), London, U.K., in 1998.

He is currently a Professor with the Department of Electronic and Electrical Engineering, UCL, where he leads the Bioelectronics Group. He has made outstanding contributions to improving safety and performance in integrated circuit design for active medical devices, such as spinal cord and brain stimulators. He has numerous collaborations for cross-disciplinary research, both within the U.K. and internationally. He has authored over 350 articles in journals and international conference proceedings, several book chapters, and holds several patents. His research interests include analog and mixed-signal integrated circuits for biomedical, sensor, and signal processing applications. He is a fellow of the Institution of Engineering and Technology (IET), a fellow of the European Alliance for Medical and Biological Engineering Sciences (EAMBES), and a Chartered Engineer (C.Eng.). He was a co-recipient of a number of best paper awards and has graduated many Ph.D. students. He was an Associate Editor, from 2006 to 2007, and the Deputy Editor-in-Chief, from 2014 to 2015, of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS; and an Associate Editor, from 2008 to 2009, and the Editor-in-Chief, from 2016 to 2019, of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS. He was an Associate Editor of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, from 2013 to 2023, and currently serves on its steering committee. He serves on the editorial board for *Physiological Measurement*. He has served on the technical programme committee of numerous conferences, including ISCAS, BIOCAS, ICECS, ESSCIRC, and NER. He was the Chair of the IEEE Circuits and Systems Society (CASS) Fellows Evaluation Committee (2022–2023) and has served on many CASS committees including the Board of Editors, John Choma Education Award Evaluation Committee, and Mac Van Valkenburg Award Evaluation Committee. He is the Chair of the U.K. and Ireland IEEE CASS Chapter and the General Co-Chair of the 2025 IEEE International Symposium on Circuits and Systems (ISCAS 2025).