A 192-Channel 1D CNN-Based Neural Feature Extractor in 65nm CMOS for Brain Machine Interfaces

Steven Bulfer[1], Jorge Gámez[1], Albert Yan-Huang[1], Benyamin Haghi[1], Volnei Pedroni[1], Richard A. Andersen[1], and Azita Emami[1]

[1]Affiliation not available

September 03, 2025

A 192-Channel 1D CNN-Based Neural Feature Extractor in 65nm CMOS for Brain Machine Interfaces

Steven Bulfer *Member, IEEE*, Jorge Gámez, Albert Yan-Huang, Benyamin Haghi, Volnei Pedroni, Richard A. Andersen, Azita Emami *Senior Member, IEEE*

***Abstract*— We present a 192-channel 1D convolutional neural network (1D CNN) based neural feature extractor for Brain-Machine Interfaces (BMI) that achieves state-of-the-art decoding stability at $1.8\,\mu\mathrm{W}$ and 12801 $\mu\mathrm{m}^2$ per channel in 65nm CMOS technology. Our device is a fully configurable, scalable, area and power efficient solution that supports models with 2-8 feature layers and a total kernel length of up to 256. This architecture reduces caching requirements by $5\times$ over conventional computation schemes. Channels and layers are individually power-switchable to further optimize power efficiency for a given neural application. We introduce an on-chip model, FENet-66, that achieves the highest cross-validated decoding performance compared to all previously reported feature sets. We show that this model maintains superior stability over time using recorded data from tetraplegic human participants with spinal cord injury. Our features have 18% higher overall average cross-validated R2 decoding performance compared to Spiking Band Power (SBP), with 28% better performance during the 4th year. Our proposed architecture can also extract mean wavelet power features at low power and latency. We show that custom 1D-CNN kernels achieve 10% better performance compared to wavelet features while compressing the neural data stream by $38\times$. The models and hardware were validated in real time with a human subject in online closed-loop center-out cursor control experiments with micro-electrode arrays that were implanted for 6 years. Decoders using features generated with this work substantially improve the viability of long-term neural implants compared to other feature extraction methods currently present in low-power BMI hardware.**

***Index Terms*— BMI, BCI, CNN, Feature Extraction, Streaming Processor**

Manuscript received May 22, 2025; revised July, 28, 2025; date of current version August 21, 2025. This work was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2039655, the Center for Sensing to Intelligence (S2I), Tianqiao and Chrissy Chen Brain–machine Interface Center, and Heritage Medical Research Institute at Caltech. Funding for this Institutional Review Board- and FDA-approved work has been provided by the National Institute of Health grant R01EY015545 and UG1EY032039 (J.G., R.A.A.), Tianqiao and Chrissy Chen Brain–machine Interface Center at Caltech (J.G., R.A.A.), Swartz Foundation (J.G.) and Boswell Foundation (R.A.A.).

S. Bulfer, A. Yan-Huang, B. Haghi, V. Pedroni, and A. Emami are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91106 USA (e-mail: sbulfer@caltech.edu).

J. Gámez, and R. A. Andersen, are with the Division of Biology and Biological Engineering and the T&C Chen BMI Center at California Institute of Technology, Pasadena, CA 91106 USA.

I. Introduction

KINEMATIC decoding with brain-machine interfaces (BMIs) enables restoration of mobility and independence for individuals with spinal cord injuries [1]. BMI technologies are rapidly evolving to require higher channel counts to perform increasingly complex tasks with higher precision, while fully implantable realizations are constrained by wireless bandwidth and power budgets [2], [3]. Feature extraction aims to reduce the data rate and distill information before data transmission or decoding by transforming neural electrical recordings into information-rich features. Long-term exposure to the neural environment on Multi-Electrode Array (MEA) implants can degrade spike Signal to Noise Ratio (SNR) due to gliation and electrode degradation [4]–[7]. This signal degradation undermines the accuracy and stability of many state-of-the-art feature extraction methods [8]. Implementing robust feature extraction methods that can operate reliably on degraded neural signals with low area and power cost is critical for the long-term viability of implanted BMI systems.

The current state of implantable feature extraction from intracortical neural recordings can be categorized into three main methods: Spike Detection (SD), Spike Sorting (SS), and compression through Broadband Feature Extraction (BFE) [3]. SD identifies occurrences of neuronal spikes in broadband data often through identifying a threshold crossing (TC) of some quality of the electrical recording (e.g. amplitude, teager energy, variance etc.). SS uses the output of SD algorithms by first isolating a spike waveform using SD, then sorting and clustering the spike based on various characteristics of the waveform with the aim of discriminating real spikes in TC from noise, or labeling each spike to a single neuron source. SS generates firing rates as features in the form of single unit activity when referring to well isolated spikes, and multi unit activity (MUA) when spikes are delineated from noise, but not each other. SS methods rely on high SNR to sort spikes and struggle to extract meaningful information from noisy signals, even with state-of-the-art algorithms [9].

BFE is distinct from SD and SS as it mitigates the SNR dependence of the spike event detection operation by directly transforming the entire sampled bandwidth of neural data into lower-dimensional features without SD to avoid losing aggregate neural information not associated with high-SNR spiking events [10], [11]. BFE techniques are able to aggregate spiking activity from low-amplitude spikes that would otherwise be lost to SD and SS. Spiking Band Power [12], [13] (SBP) is an example of a BFE technique with very low power and complexity, as it simply takes the mean of the magnitude of neural recordings filtered at 1 kHz within a given time bin. Software-based Wavelet Transform (WT) BFE demonstrated in [14], which extracts the mean wavelet transform power of neural data, also shows significant robustness in decoder performance. Building on WT BFE, our previous work uses a 1D CNN-based Feature Extraction Network (FENet) to process broadband neural data into neural features that outperform SBP and WT BFE in decoder performance and long-term stability [11].

Hardware realizations of SD [15], [16], SS [17]–[20], and neural data compression using deterministic transforms [21]–[24] are rich in literature. With low complexity, SBP has also been implemented in hardware with high area and power efficiency [25]. WT BFE has yet to be implemented on a low power and area system suitable for implantable devices. Conversely, tailored transforms such as FENet, Principle Component Analysis (PCA) [26], and Autoencoders (AE) [27] are less common as they generally suffer from high memory requirements and complexity [3]. These memory requirements arise from the many independent channels of continuous data and large sample bins used to generate each feature. We mitigate this cost through the design of a streaming architecture for 1D CNN computation which reduces neural caching requirements by 5× over CNN processors that only begin processing when a full bin of data is received. Furthermore, through efficient hardware reuse, we reduced the hardware requirements for each channel to obtain an area of 12801 μm. We achieve a power cost of 1.8 μW per channel which is comparable to state-of-the-art hardware realizations of neural data transformers. Our power and stability was achieved through architectural and algorithmic optimizations of FENet.

In this work, we re-trained and tuned the hyper-parameters of the FENet algorithm to generate a low-complexity FENet model (FENet-66), which is better suited for hardware implementation, with similar stability and decoding capabilities to the software-bound algorithms described in [11]. Further analysis into the sampling frequency sensitivity of this model shows model robustness to slower sampling-rates which reduce the power costs while maintaining similar decoder performance. We validated the model and system architecture in closed-loop with a human subject implanted with two Utah MEAs performing a center-out cursor control task.

To the best of our knowledge, this paper presents the first hardware-validated 1D CNN ASIC for neural feature extraction from broadband neural signals, achieving robust kinematic decoding six years after being implanted when single neuron activity is no longer separable. FENet-66 is introduced as a low-power hardware-compatible model which provides the best balance of feature quality and power cost. We explore two other variants of the FENet model that define a tunable solution space. The proposed 1D CNN architecture is neural stream oriented, scalable, and validated with closed-loop patient testing. This paper is organized in the following manner: We first introduce the feature extraction algorithm targeted in this work in Section II. This is followed by Section III where we present our solution to optimize this algorithm for hardware implementation. In Section IV we give an overview of our hardware implementation, and in Section V we present our methods for evaluating our solution. Section VI presents the results of our solution, and finally Section VII concludes the article with a brief discussion of potential extended applications of our techniques and future improvements to our methods.

II. FENet Algorithm

FENet [11] adopts a seven-layer architecture inspired by the Daubechies-20 (db20) wavelet transform, with additional nonlinearity and accumulation components. The model weights are initialized using db20 coefficients and are trained to optimally extract neural information, producing 8 features per data bin. Tailoring this transform to neural data allows the capture of aggregate neural activity that is nominally lost in low SNR signals. The resulting 8 features have demonstrated state-of-the-art performance in high-noise recordings from chronically implanted MEAs.

The full data processing pipeline is illustrated in Fig. 1. The data conditioning steps preceding feature extraction follow standard practices in BMIs and are already demonstrated in existing hardware systems [25], [28], [29]. Common average referencing (CAR) and neural data normalization are implemented for offline analysis, while the online system only implements CAR to simplify the system. Our ASIC implementation encompasses the feature extraction step in the decoding pipelining. Partial Least Squares Regression (PLSR) is used to reduce the number of output feature dimensions, preventing decoder overfitting. PLSR is trained per channel on data from a single day, then model parameters are averaged across all channels to generate a single transform used for all subsequent days, and is general to all channels.

The FENet algorithm is shown in Fig. 2. Each layer performs two separate 1D convolutions on its input stream with the traversal and feature-generating kernels. The traversal path (left) generates an intermediate output passed to the next layer. The feature-generating path (right) applies a Leaky Rectified Linear Unit (LReLU) non-linearity followed by global average pooling through accumulation and subsequent normalization through division.

The output feature computation is defined in (1):

$$f_l = \frac{\sum_{i=0}^{\lceil \frac{B}{S_l} \rceil} LReLU\left[\sum_{j=0}^{K_l} x_{S_l*i-j} \cdot wf_j\right]}{D} \tag{1}$$

where f_l is the output feature from the l^{th} layer. Each layer receives a bin of B samples, which is convolved with kernel weights wf_j of width K_l and stride S_l. The LReLU output is accumulated and quasi-normalized using a division factor D. The traversal path computation, defined in (2):

$$X_{L+1} = \sum_{j=0}^{K_l} x_{S_l*i-j} \cdot wt_j \tag{2}$$

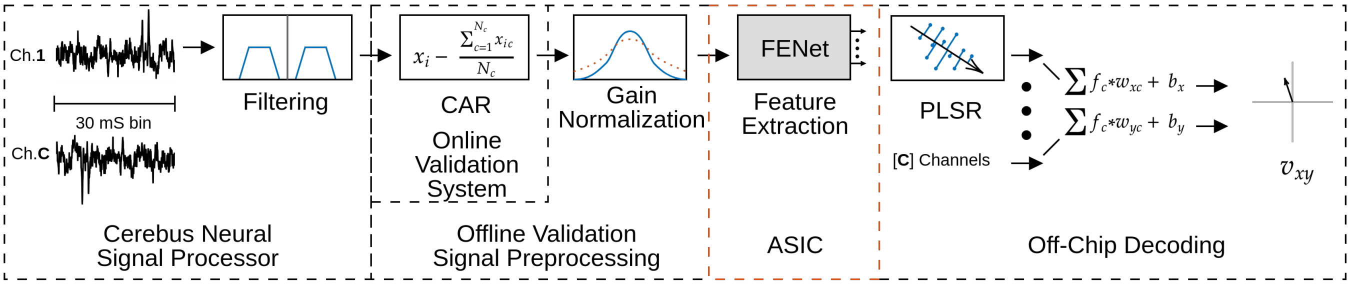

Ch.1
30 mS bin
Ch.C
Filtering
$x_i - \frac{\sum_{c=1}^{N_c} x_{ic}}{N_c}$
CAR
Online Validation System
Gain Normalization
FENet
Feature Extraction
PLSR
[C] Channels
$\sum f_c * w_{xc} + b_x$
$\sum f_c * w_{yc} + b_y$
v_{xy}
Cerebus Neural Signal Processor
Offline Validation Signal Preprocessing
ASIC
Off-Chip Decoding

Fig. 1: Feature extraction and decoding pipeline used with FENet models. ASIC targets the feature extraction step of the decoding pipeline.

produces the intermediate activation X_{L+1} for the next layer, using traversal weights wt_j. Kernel sizes are matched across both computation paths to reduce complexity.

III. Algorithm-Hardware Co-Design

Reducing model complexity while maintaining feature quality is essential for efficient hardware implementation. FENet performance remains robust under variations in kernel size (K_l) and stride (S_l) parameters [11]. A parameter sweep was performed across all FENet hyperparameters using the *wandb* [30] training framework with Bayesian optimization.

Models were chosen based on R^2 performance and complexity. Each model was trained on a 10-day dataset with 7-fold cross-validation. The selected models (Table I) span a range of models suited to different power-performance requirements. Three optimized models were evaluated:

FENet-240 -highest performance; highest complexity.

FENet-15 -highest efficiency with acceptable performance.

FENet-66 -balance between performance and complexity.

These three models demonstrate the architectural flexibility to support a range of power-accuracy trade-offs depending on system requirements. **FENet-66** was chosen for its power efficiency while maintaining the majority of the feature extraction capabilities of larger models.

The number of multiply-accumulate operations (MACs) is a rough measure of the complexity necessary for each feature extraction. The total number of MACs, Non-Padding MACs (NP-MACs), and pooling operations (which include the LReLU activation, rounding, quantization, and accumulation into the pooling register) for a given model is listed in Table I and assumes a bin size of 150 samples, corresponding to a 30 ms window at 5 kSps. Convolution padding mitigates aliasing from fixed-length binning but adds extra multiply operations and latency. For FENet-66 operating on 150-sample bins, padding accounts for 12.6% of total MACs. These computations were eliminated by trimming the convolution at the edges of data bins.

FENet software was previously tested at 30 kSps [11], consistent with other techniques in the literature [16], [17], [31], [32]. This rate was originally used because it is the maximum sampling rate provided by the FDA-approved Blackrock Cerebus system. To test robustness to lower sampling rates, data sampled at 30 kSps were reduced by integer factors while adjusting bin sizes to maintain an output feature rate of 33 Features-Per-Second (FPS). Downsampling was performed by first filtering the data by $\frac{1}{2}$ the downsampled rate with a low pass anti-aliasing filter followed by decimation.

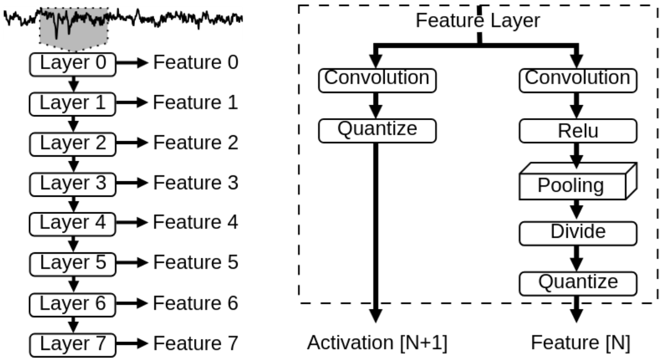


Fig. 2: (Left) Multi-layer data flow for FENet on a single neural channel. (Right) Internal computation within each layer.

In Section VI-B, we explore the effect of sampling rate on decoding performance. Even with aggressive down-sampling, decoding performance remains stable. Based on this, 5 kSps was selected for hardware comparison, reducing data volume by 6× with minimal impact on feature quality.

IV. Hardware Architecture

The proposed 1D-CNN stream-oriented processor is designed to extract features from high-bandwidth neural data streams in a scalable and configurable manner. The system is a broadband feature extractor which outputs 2-8 values per channel that represent the aggregate presence of learned feature kernels in the neural data. Like other BFE features, these features are able to represent the presence of low amplitude neural activity from highly noisy signals that would otherwise be lost to SD and SS feature extractors. The architecture is optimized for stream processing with minimal activation caching since it schedules operations based on the availability of input data and completes each convolution in a piecewise manner between slowly arriving neural samples. This allows scaling from very few, to hundreds of neural channels to be processed in real-time with minimal control and memory overhead.

The ASIC hardware components can be classified into two main groups: the CNN solver hardware under test, and the

TABLE I: Selected hardware-optimized models.

Model	[11]	FENet-240	FENet-66	FENet-15
K_0	40	40	36	10
S_0	2	2	2	3
LReLU Leak Slope	-1	-1	-1	$-1/2$
K_1	40	40	14	5
S_1	2	2	2	3
LReLU Leak Slope	-1	-1	-1	$-1/64$
K_2	40	40	16	-
S_2	2	2	2	
LReLU Leak Slope	-1	-1	-1	
K_3	40	40	-	-
S_3	2	2		
LReLU Leak Slope	-1	-1		
K_4	40	40	-	-
S_4	2	2		
LReLU Leak Slope	-1	-1		
K_5	40	40	-	-
S_5	2	2		
LReLU Leak Slope	-1	-1		
K_6	40	-	-	-
S_6	2			
LReLU Leak Slope	-1			
Total Weights	560	480	132	30
Total MACs	30240	27120	9136	1250
NP-MACs	19560	17960	7520	1176
Pooling OPs	417	379	210	91
SRAM Writes	528	489	327	222
PE Clock Multiplier	-	21	21	26
Cycle Count/Feature	-	11908	5449	1636

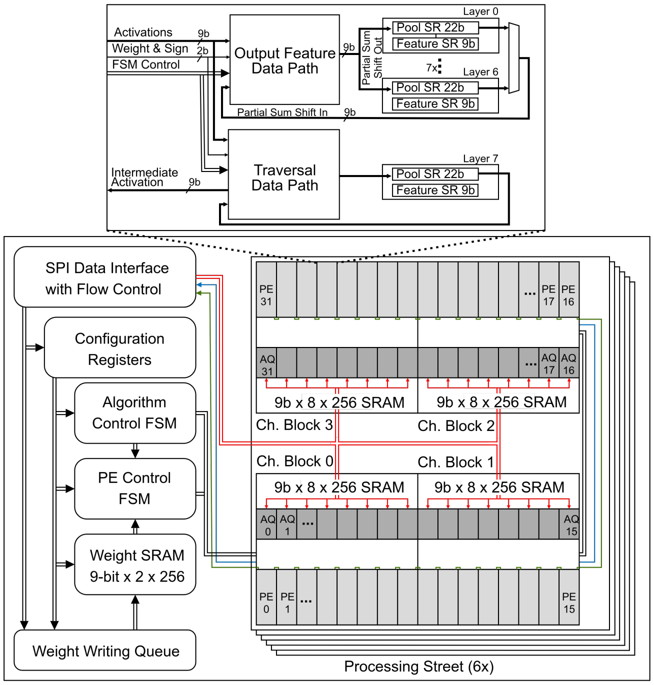


Fig. 3: (Top) Single channel Architecture. Two arithmetic units simultaneously process the traversal and feature generating data paths. Intermediate values are passed to pooling accumulation register blocks, selected by a multiplexer. (Bottom) System architecture with channels arranged in blocks, and blocks sequenced into streets. Various busses are color coded: Interface data bus and channel enable (red), algorithm and mac control (black), data available (blue), and feature out (green).

validation system. The former consists of the channel block macro, algorithm control finite state machine (FSM), processor control FSM, and configuration registers. The latter is the custom serial data interface, which exchanges data with the external validation system.

The system diagram is shown in Fig. 3, which illustrates the data flow of the system and channel architectures. Processors that share a common SRAM memory are grouped into channel blocks, and channel blocks that share a buffer chain and activation bus are further organized into streets. The data interface serves as the primary access point for configuring registers, loading weight memory, and streaming neural activations into the system. Multiple clock and voltage domains, as well as channel and layer level power gating are used to minimize power.

We enable a high degree of parallelism without excessive area overhead by utilizing word-serial processing between each system clock cycle. The use of separate clock networks allows the high speed processing element clock (MAC clk in Fig. 4) to be constrained to a lower VDD domain, reducing switching power. The PE control FSM synchronizes control signals to the rising edge of each system cycle, and does not begin a new control sequence until the next rising edge of the system clock, preventing control misalignment. The frequency ratio is adjusted for model parameters that require more MAC cycles.

The system utilizes a third asynchronous interface clock for IO to emulate data from independently timed sources. Neural data is distributed to each channel via the data interface by a shared bus. Although the data interface is utilized to compensate for the limited off-chip IO of the validation ASIC, the interface data bus is intended to be replaced by parallel data sources within a full chip BMI decoding system. For this reason, each channel is equipped with a 4 element Asynchronous Queue (AQ in Fig. 3), to allow for neural data caching while the SRAM is in use for computation.

The presence of data in all enabled channels signals the central FSM to trigger the load operation of the first layer. This operation transfers neural data from the AQ into the first layer's SRAM space. Once a stride of data is loaded within a layer's memory space, higher-order dependencies are satisfied, and the processing element becomes available, data within a layer's SRAM space is read back and presented to the processing element for computation.

At the completion of feature generation, the pooling register value is reduced to 9 bits, rounded, and added to the feature shifting register (Feature SR). At this point, the processor begins a new computation, while the feature is shifted out of the processing element. Each channel is equipped with an always-on skip multiplexer that allows its position in the feature scan chain to be skipped in the event the channel is powered down. The features are shifted out, and returned to the data interface for exporting off the chip.

The proposed architecture is designed to efficiently map a wide range of FENet models. Kernel size, stride, LReLU leak slope, and pooling parameters are all configurable through the CNN control FSM sequence. Up to 8 features can be generated from 7 feature-generating layers and one terminal traversal layer. Kernel size is limited only by the depth of the weight SRAM (256 elements) such that the sum total of traversal kernel weights must be less than 256 (the total number of kernel weights is 2× traversal weights). Bin sizes are defined by multiplying the first layer stride with a programmable cycle counter, allowing maximum bin lengths up to 2048 strides. These options provide a wide hyperparameter space for optimizing power, performance, and decoding stability.

To support system scalability, the processing hardware and activation/feature caching are integrated into a modular channel block macro. Each channel block contains 8 processing elements (PEs), each with their own asynchronous data queue, gated control-signal buffers, and power-domain level shifters. All PEs share a customized TSMC 72-bit × 256-element low-leakage single-port SRAM macro. To accommodate the higher voltage headroom requirements of the proprietary SRAM, we separate the memory and processing element VDD domains. Single-port SRAM macros are used for compactness and power efficiency, with write access multiplexed between asynchronous activation queues and processing elements.

A. Channel Architecture

Each channel is designed to minimize hardware complexity while maintaining full configurability of the FENet algorithm and enabling fine-grained power optimization. Control signals and weight data are broadcast globally by a centralized FSM, orchestrating synchronized computation across all processing elements. We introduce a multi-modal data path design to enable the simultaneous computation of output features and intermediate activations unique to the FENet algorithmic architecture. Each channel contains two fused data paths operating on the same input activations; the feature path, which writes partial sums to one of the first seven pooling blocks, and the traversal path, which handles intermediate feature computation between layers. For lower-order layers, traversal partial sums are written back to SRAM to serve as inputs for higher-order layers. For the highest-order layer, the traversal output is instead accumulated into the final pooling block, producing the last feature output.

A block diagram of the MAC architecture is shown in Fig. 4. To minimize the hardware footprint, the 8-bit adder is reused across multiplication, leaky ReLU application, rounding, overflow clamping, and pooling accumulation. Pooling division and leaky ReLU are efficiently implemented through bit-shift operations. This design choice reduces the area cost and enables scalable integration of hundreds of channels, at the cost of switching power.

Analysis of bit resolution on the broadband neural data determined that 9-bit sign-magnitude fixed-point format with a 6-bit fractional component for activations and weights minimizes hardware cost while maximizing decoder performance by preserving sufficient dynamic range for neural feature extraction. The sign-magnitude data format reduces SRAM switching activity [33], [34] compared to two's complement during write operations. We determined that the pooling register accumulator required a size of 22-bits to avoid loss of decoder accuracy. This register is rounded at the end of computation to reduce each feature to 9-bits to minimize output bandwidth while maintaining decoder performance. During convolution, intermediate results are accumulated in a 16-bit two's complement register to simplify arithmetic operations, then converted back to sign-magnitude format for SRAM storage. To ensure robustness during long accumulation sequences, overflow clamping is implemented.

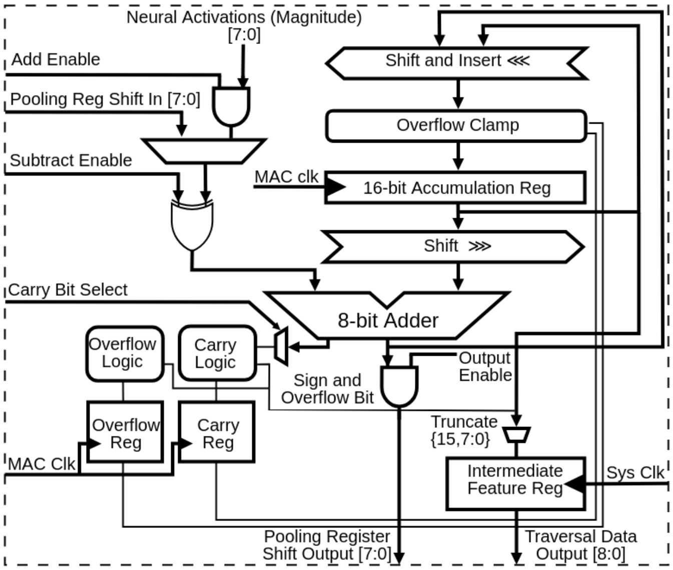


Fig. 4: Architecture of single data path processor.

Following the convolution, the accumulation register is rounded and reduced to 9 bits and accumulated in one of 7 pooling registers. The traversal path rounds and latches its value to the intermediate feature register. This value is written in the SRAM space of a higher-order layer. If the final layer is active, the traversal path instead accumulates its value in its own pooling register.

B. Control FSMs

For streaming neural interfaces, caching full neural data bins is impractical due to memory costs, necessitating real-time data processing. We implemented a streaming-oriented CNN control FSM to generate SRAM addresses and sequence layer operations so that higher-order layers only compute once sufficient data (one stride) is available and processing element resources are free. The algorithm also manages efficient zero-padding by dynamically adjusting kernel width: growing at startup, maintaining a constant size during steady-state processing, and shrinking at completion as the active window exits the kernel. Fig. 5 illustrates this control across three layers during startup, steady-state, and completion phases. It also depicts the interaction between states of each layer. These interactions include the triggering of the loading state of higher-order layers by the completion of a lower-order layer's convolution and higher-order layer's priority over MAC resources to free up their memory spaces for new intermediate activations from lower-order layers.

The stall data signal in Fig. 5 indicates when the AQ is full and to apply back-pressure on the data sources. During the conclusional padding state (Fig. 5 C), memory resources become busy, preventing the loading of new data. By optimizing the system frequency to consume data at the optimal rate for a given model, back pressure is minimized or eliminated altogether.

The CNN sequence is directed by a group of centralized FSMs (one for each layer), which ensure that the states of each layer are compatible with each other to prevent resource contention. Each layer will wait in the `IDLE` state while higher order layers complete their computations, which frees up memory within the higher order layer's partition. The `LOAD DATA` state writes activations to a layer's partition, and is triggered either by partition space becoming available (layer 0), or a lower order layer finishing a convolution (layers 1-6). On the completion of loading a full stride of activations into a layer's partition, the layer enters the `CONVOLVE` state, on the condition that memory space is available in the higher-order layer and PE resources are available. If neither one of those two criteria are met, this layer is stalled by waiting in the `WAIT MULT` state. Following the convolution, the `UPDATE` state adjusts the convolution pointers for the next convolution. In the `WAIT FOR RESULT` state, LReLU is applied and partial sums are either added to the pooling register (when generating an output feature), or to the higher order layer's SRAM partition (traversal path for layers lower than the last layer). Finally, the `WAIT FOR FORMAT` state normalizes and rounds the feature in the pooling register to 9 bits. The `FINISH` state waits for higher order layers to finish their padding computations before commencing a new sequence.

During each system clock cycle, the processing element modifies its objective depending on the current state of the CNN control hardware as shown in Fig. 6. During the `CONVOLVE` CNN state, the blue path is taken, where the PE word-serially multiplies and accumulates activations in the accumulation register within the `ADD` MAC state. During the `Update` CNN state, the partial sum is added to the pooling register of the current active layer. The pooling accumulation and LReLU occur simultaneously by first rectifying the accumulation register in the `LReLU` MAC state, then shifting and adding this value to the pooling register. If the LReLU parameter is greater than 0, the pooling register of channels with negative accumulation values are stalled during shifting, such that the accumulator value is effectively divided by the parameter's set number of bits within the `LATCH POOL STALL` and `FINISH POOL` states. During the `Wait For Format` CNN state, the pooling register is normalized by bit shifting (`DIVIDE POOL`, `SHIFT POOL`), rounded (`START ROUND & QUANTIZE`, `FINISH ROUND & QUANTIZE`), and shifted back into place (`RESTORE POOL`). During the `WAIT FOR FORMAT` CNN state, a 9 bit partition of the pooling register is written to the feature shift register for export.

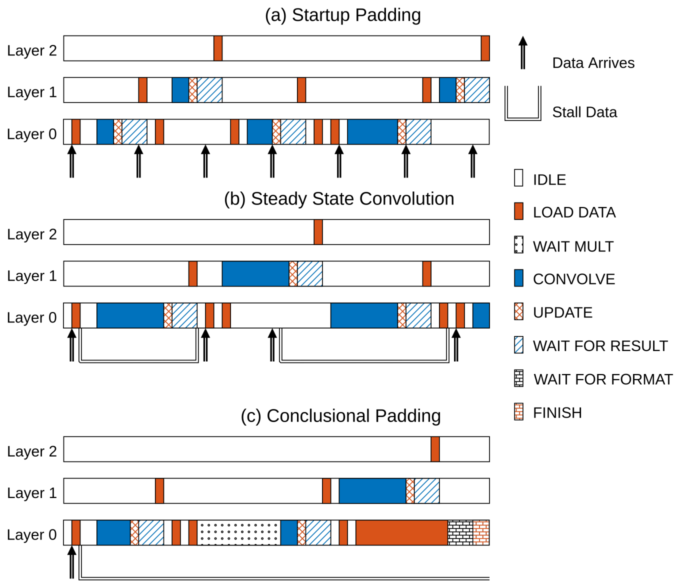


Fig. 5: Control behavior for 3 layers of the CNN depicting the 3 padding state behaviors: (a) Startup Padding (b) Steady State Convolution (c) Conclusional Padding.

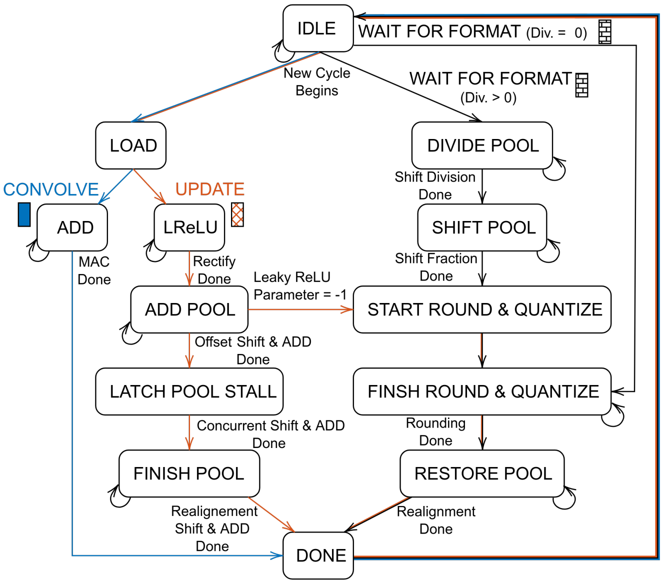


Fig. 6: FSM control of the PE. Each colored path corresponds to a different control path depending on the state of the CNN control FSM.

V. System Evaluation

A. Validation Setup and Metrics

The hardware setup for offline validation is shown in Fig. 7 and consists of the fabricated FENet ASIC, a Xilinx ZCU106 FPGA validation server, and laboratory power measurement equipment. Pre-recorded or real-time neural data are streamed to the FPGA, which applies simple array-wide CAR and transmits the data to the ASIC for feature extraction.

An emulated replica of the ASIC logic was implemented on the FPGA to verify ASIC outputs in real time. The

ASIC was powered with separate voltage domains for the validation interface, processing elements, and memory. Power measurements were collected after a 30-second stabilization period to ensure steady-state conditions. IO and validation system power were excluded from all reported values.

It is important to note that unlike SD and SS methods, BFE does not detect and classify spikes, but captures aggregate neural activity within broadband data, which makes direct comparison with these methods inapplicable. Therefore, to evaluate performance, we followed the cross-validated linear decoding methodology established in [11]. Features were generated from neural recordings and used to train and test a linear decoder. Decoder performance is measured by the coefficient of determination (R^2), described in (3):

$$R^2_{vx|vy} = \left(\frac{\sum (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum (y_i - \bar{y})^2 \sum (\hat{y}_i - \bar{\hat{y}})^2}} \right)^2 \qquad (3)$$

which quantifies the correlation between the decoded and intended target velocities. Given the 2 degrees of freedom in center-out tasks (X and Y velocity), R^2 values for each dimension are combined into a single score via the root mean square, as shown in (4).

$$R^2 = \frac{1}{\sqrt{2}} \sqrt{(R^2_{vx})^2 + (R^2_{vy})^2} \qquad (4)$$

Offline open-loop decoding analysis validates the ASIC feature performance across a large set of prerecorded data. Data from 48 center-out sessions [11] were used for benchmarking. Raw neural signals were first preprocessed by removing the first 2 PCA components of each array (for PCA-based CAR), followed by an 8th-order elliptical high-pass filter (80 Hz cutoff, 0.01 dB passband ripple, 40 dB stopband attenuation) and batch normalization.

Hardware-generated features were reduced from N to one feature per channel using a PLSR model. To mitigate overfitting, a single averaged PLSR model was trained offline on one session, then applied across all 48 sessions. A linear least-mean-squares regression decoder was trained with 10-fold cross-validation for each session to compensate for non-stationary effects of the implant due to micro-movements.

Feature performance was compared against established feature extraction methods in the literature including WT BFE, SBP [25], [35], TC, and MUA. WT features were generated by loading our ASIC with the hardware-friendly Haar WT which has 3 layers with kernel size 2, totaling 4 output features. We use the same number of layers as FENet-66 to keep the decoding dimensions the same. We also used the same sampling rate as the target rate of FENet-66 (5 kSps) such that only the effects of using trained kernels are compared. SBP features were generated by first filtering the neural data at 1 kHz, and downsampling to 2 kSps, then averaging the magnitude of neural recordings within the 30 mS time bin. TC features were generated by counting crossings over an adaptive threshold set at $-3.5\times$ the root mean square of the neural signal in 30 mS bins. MUA features were generated by SS the threshold crossing events utilizing the sorting and clustering algorithms used in [36].

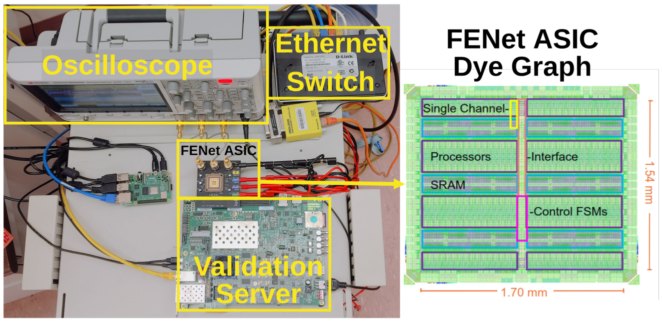


Fig. 7: Offline validation setup (left). The FENet ASIC is connected to the validation server over the built in mezzanine connector. The validation server is attached to an Ethernet local area network which relays neural recordings from either an experimental test computer, or real-time neural data from a Cerebus neural signal processor. The Ethernet network also relays features generated from the ASIC to an external computer for decoding. The dye graph (right) of the FENet ASIC.

B. Online Validation

We validated the FENet ASIC's performance with online kinematic decoding trials. Our participant (JJ) was implanted with two 96-channel MEA Utah devices in their motor and peripheral parietal cortices six years before this online evaluation. All procedures were approved by Caltech's Institutional Review Board (IR20-0983).

We validated closed-loop decoding using hardware-generated features during a center-out cursor control task. An initial decoder was trained in an open-loop trial where the participant (JJ) imagined tracking an on-screen cursor with his thumb without feedback. ASIC-extracted features from this trial were used to train a linear decoder. In subsequent closed-loop trials, the cursor position was updated in real time based on decoded kinematics. The decoder was fine-tuned through successive trials, gradually reducing assistance, ultimately achieving fully autonomous control.

The online setup (Fig. 8) streamed neural data at 30 kSps using a Cerebus neural signal processor, which digitized and bandpass filtered the signals. Data were sent to the FPGA-based validation server where CAR was applied before being forwarded to the ASIC for feature extraction. Features were transmitted back to the trial computer for real-time decoding and cursor control.

The decoding pipeline was implemented in Matlab on the experiment computer. Features were dimensionally reduced using PLSR, then decoded to kinematics with a linear decoder, exponentially smoothed [37], and displayed as cursor position to the patient.

Post-hoc SS analysis was performed on recorded neural data to assess electrode quality and evaluate the viability of SS feature extraction under the implant's noise conditions, following the method of [36] using threshold detection and k-means clustering on 2-4 PCA components of the spike waveform.

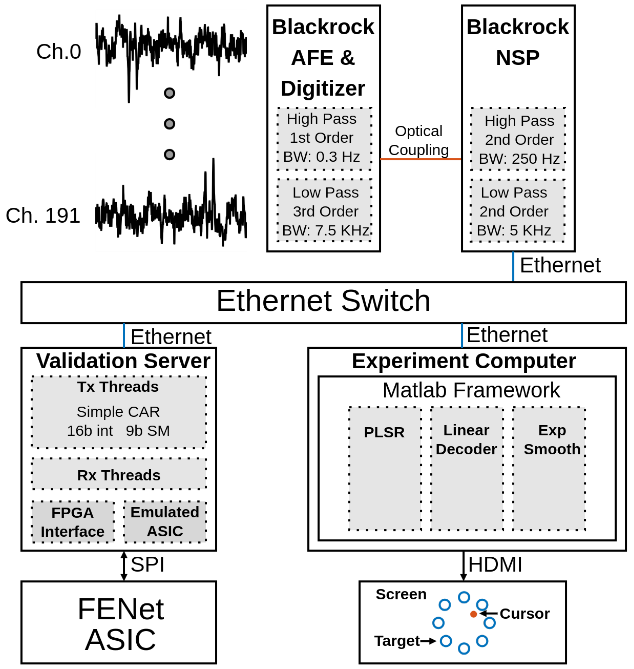


Fig. 8: Data flow for neural data retrieved during an online session.

VI. Results

A. Hardware Analysis

Conventional CNN processing systems must cache the entire bin of neural data before computing convolutions. They must also write back their intermediate results to memory for the next layer. This architecture employs a scheme that processes data as it become available in large batches. This minimized the memory requirements on the system to only the width of the kernels utilized in the CNN model. For instance, FENet-66 only requires 66 elements of memory per channel, whereas a conventional system would need enough memory for a full bin of raw neural data and intermediate activations (total of 327 elements for FENet-66 operating on sample bins of size 150). Our architecture reduces the algorithmic memory requirements by 5× for FENet-66 operating on 5 kSps data and 26× operating on 30 kSps.

The FENet ASIC, shown in the die graph in Fig. 7, was implemented in 65 nm LP CMOS technology. The ASIC supports up to 192 neural streaming channels and occupies a total core area of 2.62 mm^2. Each channel occupies 12801 μm of area with the processing element consuming 1447 standard cells (7156 $\mu\mathrm{m}^2$). This chip demonstrates chaining of up to 4 channel blocks in series with 6 parallel chains, showing the channel block MARCO's scaling potential since additional channels only require linear increase in area and power without adjusting clock frequency.

Two core voltage levels are used to minimize processing power (MAC VDD) while maintaining the voltage headroom necessary for low leakage SRAM (MEM VDD). The breakdown of power for the components of each channel is shown in Fig. 9 B and is estimated by post-place and route simulation at 1 V and 1.2 V and was observed to be relatively consistent across voltages within each respective domain. The total power ratio was measured directly with a MEM and MAC voltage of 0.9 V and 0.63 V, respectively. The voltage sensitivity of the ASIC is depicted in Fig. 9 A. The measured sensitivity is consistent for system clock frequencies less than 700 kHz, which is fast enough to operate FENet-66 with a padding latency of only 1.62 ms. No errors were observed for MAC VDDs greater than 0.65V within this operating range. The mean squared error (MSE) between the ASIC and FPGA-reference generated features are plotted alongside the cross-validated decoder R^2 using data from the first session of the offline data also used in Fig. 10. The decoder features show robustness to total MSE values less than 100 such that the R^2 performance maintains 98.8% of its value. As such, the minimum voltage values for the MAC and MEM VDDs are chosen to be 0.63 V and 0.90 V, respectively. The decoder was trained using the FPGA features and validated with those generated by the undervolted ASIC. For this performance, the feature extraction ASIC consumed 346.2 μW to generate features for all 192 channels at an efficiency of 335 GOPS/W where a MAC operation is two OPS with a padding latency of 6 mS. The processing element consumes 179 μW (52%) of the total power. These power results exclude the power contributions of the validation interface and IO.

The power consumed in various device configurations is shown in Table II. Power was not measured directly during the online test, but the power draw for the same number of channels and sampling rate was measured to be 586 uW (3.05 μW per channel). In a highly optimized case, where only top-64 most informative channels are used streaming neural data at a 5 kSps sampling rate, the feature extraction power was reduced to 140 μW (2.2 μW per channel), while maintaining an average R^2 of 0.354 over all 48 sessions. Alternatively, all 192 channels can be operated at 2 kSps with 178 uW (0.93 μW per channel), with an average R^2 of 0.41. Reducing the device to 1 channel shows a minimal operating power of 25.2 μW, demonstrating the operation floor of this device.

The chip was constructed with a custom serial interface for validation that has a maximum bandwidth of 95 Mb/s operating at a clock frequency of 42 MHz. The power consumption of this data interface was measured at 0.9 V to be 6.14 μW/MHz. The chip was also fitted with a JTAG interface to support debugging.

The FENet ASIC running the FENet-66 model substantially cuts the data rate necessary for transmission by 37.5× when operating on 5 kSps neural streams and 225× when operating on 30 kSps neural streams. Features are successfully generated at a rate of 33 FPS, which is sufficient for rapid, fine motor control.

Intrinsic processing latency in the ASIC is dominated by the final padding phase, when no new input data is available for the current bin. Data that is not streamed during padding must either be externally cached or discarded, however, as discussed in Section VII, this issue would be mitigated through modification of the control FSM, to allow caching into the

TABLE II: Power of various configurations using FENet-66.

Channel Count	S. Rate (kSps)	Freq. (MHz) System//MAC	Voltage MEM//MAC	Total (μW)	Ch. (μW)
192	2	0.095//2.0	0.90//0.63	178	0.93
64	5	0.188//3.984	0.90//0.63	140.0	2.2
192	5	0.188//3.984	0.90//0.63	346.2	1.8
192	10	0.345//7.25	0.93//0.65	586.4	3.05
192	30	0.980//20.58	0.98//0.67	1584.0	8.25

Gray highlighting denotes sampling rate used in online testing.

SRAM of the first layer during the conclusional padding phase. Because new data cannot be streamed during padding, optimizing the number of MAC operations during this phase is critical. The number of clock cycles required to process the padding phase for each model is summarized in Table III along with the padding latency incurred while operating at the minimum frequency to achieve a feature rate of 33 FPS. Since all other computations are completed in-between the arrival of data-samples, the number of clock cycles of latency is constant and determined by the number of padding cycles for a given FENet model, regardless of the number of channels.

FENet-66 requires 6.2× fewer padding cycles than FENet-240, enabling feature generation at 33 FPS while operating the system clock at only 188 kHz with 5 kSps neural data and a padding latency of 6 ms. In contrast, FENet-240 requires a 2.9× higher system clock to meet the same feature rate due to its larger model complexity. These improvements in padding efficiency directly translate to lower operating frequencies and reduced dynamic power.

We explore the power requirements of different models at various system clock frequencies in Fig. 9 C. The MAC clock was maintained at a multiple of the system, consistent with the PE clock multiplier listed in Table I. The data interface frequency was maintained at $9 - 14\times$ the system frequency. We further limited the neural data packet rate to 5 kSps, regardless of the maximum throughput of the ASIC, so that our system generates 33 FPS over the entire range of operating frequencies to match realistic data rates. MAC VDD was scaled so that R^2 performance is maintained above 98%. Since the interface speed was limited, MEM VDD was maintained at 0.9V across all frequencies. We further discuss the effect of model complexity on the minimum operating frequency of the model in Appendix I.

We also measured the performance of the system using each model over the same system frequency range, without constraining the neural data rate. In this case, we optimized the interface clock frequency to the minimum frequency at which the processor can remain computationally limited, maximizing efficiency. FENet-15, FENet-66, and FENet-240 were measured to each require a minimum energy of 24 nJ, 42 nJ, and 82 nJ, respectively for each feature set per channel. Noting the values in Table I, we calculate the max efficiency to be 104 $GOPS/W$, 424 $GOPS/W$, and 661 $GOPS/W$, respectively (1 MAC = 2 OPs). This range in efficiencies correlates to the ratio of MAC operations, to the total number of cycles each feature set requires.

Each processing element, which has two independent data paths, consumes 7165 μm^2 of area per channel (1447 gates), and the 4 element asynchronous queue and level shifters further consumes 1144 μm^2 (121 gates), which is 56% and 9% the total area per channel, respectively. The 256-element SRAM consumes 3790 μm^2 which is 30% the channel area, the remainder being used for buffering and signal gating. The control logic and weight SRAM occupies an area of 31626 μm^2 (2423 gates) and 100082 μm^2, respectively.

With neural decoding systems rapidly incorporating hundreds to thousands of channels, scalability of power, area, and latency are of utmost importance. Our architecture completes all possible computations at the same rate as data arrival. Conclusional padding cycles, defined only by the model, delineates the latency and therefore remains constant when scaling the system. The scaling behavior of power and efficiency of our system is shown in Fig. 9 D. The power of the system scales with the equation: (5)

$$Power(\mu W) = \alpha + \kappa * N_{Ch.} + \beta * N_{Bl.} + \sigma * N_{St.} \tag{5}$$

where α is the baseline power of the system and κ, β, and σ are the scaling factors for the number of channels, blocks, and streets enabled, respectively. We measured the parameters running FENet-66 at 0.188 MHz with a MEM and MAC VDD of 0.9 and 0.63, respectively. The values for $\alpha,\ \kappa,\ \beta$, and σ are measured to be 15.102, 1.245, 0.985, and 7.833, respectively.

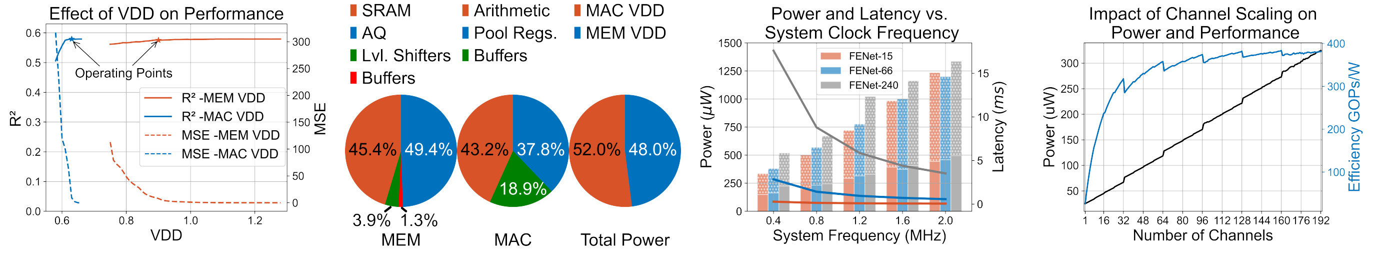
With this model, we can extrapolate the power scaling of the system. For a projected feature extraction system with 1024 channels, we would simply increase the number of streets (32 channels and 4 blocks per street) from 6 to 32. This predicts our system power to be around 1.67 mW.

All controls for the system are centralized and broadcast to each channel, with each channel block designed to be self-contained. Adding additional streets requires only buffering the control signals from the central FSM. As a result, the area of our system scales linearly, with a projected 1024 channel system requiring an additional 5.59 mm^2. Furthermore, each channel has an individual neural data port. With our validation system limited by IO, our channel count is ultimately constrained by the design specifications of the data interface. Integrating this FENet MACRO into an SOC with independent data sources would allow each channel to accept neural streams in parallel.

B. Decoding Performance

The cross-validated R^2 decoding performance of the proposed FENet models was evaluated across 48 neural recording sessions and compared against conventional feature extraction methods. The performance over four years post-implant is shown in Fig. 10 and the average performance across sessions for various sampling rates is shown in Fig. 11.

We observed high stability in performance for FENet 66 and 240 versus sampling rate down to 5 kSps. This is a result of the fact that a typical neural spike sampled at 30 kSps has a waveform that occupies approximately 40 samples, which is similar to the kernel width of the first layer of FENet 66 and 240. Their ability to accentuate real neural spikes from noise is related to the kernel's similarity to the average neural spiking shape. Furthermore, models with more layers are able



(A) (B) (C) (D)

Fig. 9: **Operating conditions** (unless otherwise specified): **Model**: FENet-66; **Sampling Rate**: 5 kSps; **Clock Frequency**: System 0.188 MHz, MAC 3.948 MHz, Interface 6 MHz; **VDD**: MEM 0.9 V, MAC 0.63 V
(A) VDD scaling effect on feature quality. Solid lines denote R^2, while dashed lines denote MSE of the features. First session of offline data is used for analysis on effect. **(B)** Power breakdown of a single channel separated by VDD domain. **(C)** System power and latency at different operating frequencies. MAC voltage is scaled so R^2 maintains 98.8% performance. Bar graph denotes the power with top and bottom bars denoting MAC an MEM domains, respectively. **(D)** Power and efficiency scaling relation to the number of channels enabled.

TABLE III: Latency and minimum system clock frequencies required to achieve 33 FPS feature generation.

Model		FENet-240	FENet-66	FENet-15
Padding Clock Cycles		7038	1135	111
30 kSps	System Clock [MHz] Latency (ms)	1.60 4.4	0.980 1.2	0.310 0.4
10 kSps	System Clock [MHz] Latency (ms)	0.664 10.6	0.345 3.3	0.105 1.1
5 kSps	System Clock [MHz] Latency (ms)	0.556 12.6	0.188 6.0	0.054 2.1

to maintain their feature extraction ability at lower sampling frequencies because the power of neural spiking shapes is redistributed to higher-order layers.

FENet 15, has first layer kernel size of 10, which at 30 kSps is unable to fit an entire waveform into a single convolution; this explains why it performs worse at 30, kSps, and better when the size of a typical neural spike after downsampling is similar in length to the first kernel. However, since it has the least number of layers of all the models, less power is able to be redistributed to higher order layers when the sampling rate is reduced further from its optimal value.

We compare the robustness of models FENet-240, FENet-66, and FENet-15 to lower sampling rates in Fig. 11. For comparison, the session-averaged R^2 performance of WT, SBP, MUA, and TC features are shown as stars for reference as shown in Fig. 11.

At 5 kSps, FENet-66 achieves an average R^2 of 0.46, maintaining 98.7% of the cross-validated offline performance of FENet-240 while requiring 2.6× fewer MAC operations. FENet-66 also outperforms WT (10%), SBP (18%), TC (38%) and MUA (41%), achieving a total average R^2 of 0.382, 0.282, and 0.275, respectively. To fairly compare SBP to FENet, we also measured the performance of FENet-66 on 2 kSps data and found the average R^2 to be 0.411, which is still 8% better than SBP. This performance is attained while only consuming 346 μW over all 192 channels (1.8μW per channel). FENet-15, although lower in decoding performance, still outperforms all other hardware implemented methods at 5 kSps, while also minimizing power consumption to 219 μW (1.14 μW per channel). The 4 layer Haar WT does remarkably well for its simplicity, achieving 90% the performance of our trained kernels, while requiring 177 μW (0.92 μW per channel).

The proposed architecture differs from conventional CNN accelerators by processing time-series neural data as it arrives, without requiring large activation caches. As a result, 3 distinct timing metrics impact system performance: the time to process incoming data streams, the latency incurred during the padding phases of convolution, and the time required for the validation system to deliver input data to the asynchronous queues.

In this work, those timing metrics combine to affect the feature rate defined as the speed at which the system can generate complete feature sets from streamed input data at a given set of clock frequencies. Feature rate depends on both data availability and the computational latency of the processor.

Chronic performance stability is illustrated in Fig. 10. Notably, TC performance degrades sharply by year three after implant (session 35), coinciding with the loss of separable single-unit activity (SUA) on the 2 arrays. FENet-66 consistently maintains a higher average decoding performance after the loss of SUA in the fourth year (sessions 35-48) of 0.404 compared to WT (0.370), SBP (0.315), MUA (0.134), and TC (0.083). The Normalized Performance Retention (NPR) show in (6):

$$NPR = \frac{R^2_{ith\ \ year}}{R^2_{first\ \ year}} \tag{6}$$

provides a measure of the stability in performance for each feature extraction method. Comparing the first-year average R^2 (FENet-66: 0.605, WT:0.578, SBP:0.552, MUA: 0.552, TC: 0.509) to the fourth, the NPR for FNet-66 is 0.66, whereas the other methods have a NPR of 0.64, 0.57, 0.48, and 0.16, respectively. This highlights the benefit of the FENet hardware in maintaining decoder stability over long implant lifetimes. The power consumption of our chip is comparable to prior state-of-the-art neural data transformers implemented in hardware as shown in Table IV while maintaining long-term stability.

The validation system delivers input data serially at a limited effective bandwidth of approximately 2.25 $\mathrm{Mb/MHz}$, which can constrain the maximum achievable feature rate for models with high computational loads and high channel counts. However, this parasitic limitation is a function of the validation system interface, and not the processor architecture.

For each relevant sampling rate, the validation system interface clock frequency was chosen such that the interface bandwidth is sufficient to support the required bandwidth for each sampling rate. It is notable that the minimum operating frequency is slightly inflated due to the constraint imposed by the validation system, since the processor would not have to wait for the excess loading time imposed by the validation system in neural systems with multiple data sources.

C. Online Closed-Loop Decoding

Offline validation allows analysis of a broad set of system parameters, whereas real-time closed-loop analysis ensures generalization of offline results within a real-world setting that has a number of confounding variables such as latency between feature generation and kinematic prediction. We tested the feature extraction system by decoding kinematic intent using ASIC-generated features and returning visual feedback to the patient by updating the position of a cursor on screen within a center-out task.

The linear decoder was first trained using an open-loop trial and had a combined x-y R^2 performance of 0.71. The neural data for this open-loop trial was later processed using the software-FENet implementation of [11], which yielded a cross-validated R^2 of 0.70. This shows that the hardware implementation maintains open-loop decoding performance similar to software-bound implementations even six years after implant. Spike analysis of this neural data yielded no detectable single neurons and a total of 119 non-separable spike channels with a mean and median SNR of 1.12 and 0.94, respectively. There were only two channels with an SNR greater than 3 (maximum 5.25). The ability of our hardware to generate usable features for kinematic decoding from such noisy signals exemplifies the importance of stable decoding hardware for implantable devices. Utilizing the spike activity filtered from noise, we performed open-loop decoding which yielded a cross-validated MUA R^2 performance of 0.43.

We show the closed loop kinematic decoding trial utilizing FENet-66 processing 10 kSps neural data streams in Fig. 12. The FENet ASIC generated 33 FPS for all 192 channels. The mean time-to-target is measured at 1.00 seconds, with an R^2 of 0.66. All 192 channels were used in decoding without gain normalization on the validation server.

D. Comparison with Other Neural BMI ASICs

There is currently a surge in the development of processors targeting low-power edge applications [18], [40]–[42]. Our feature extraction chip is a domain-specific architecture optimized to implement FENet in the neural decoding environment. To the best of our knowledge, this is the first system which integrates multi-level global average pooling accumulators and dual mode convolutional data paths for each channel. Our dual mode processing elements generate intermediate activations for higher order layers, while simultaneously computing output features. With an 8 layer output stationary processing element, we entirely avoid re-fetching intermediate activations to generate the output features. Through intensive hardware reuse, channel-level-power scaling granularity, and unique data streaming structure, we optimized this architecture for the neural decoding environment, where other conventional CNN processors can be too bulky or memory intensive for the FENet workload. We optimized the data flow for FENet feature generation to achieve high kinematic decoding accuracy and stability with long implant lifetimes. Table IV compares the proposed FENet architecture with existing hardware implementations of neural feature extractors.

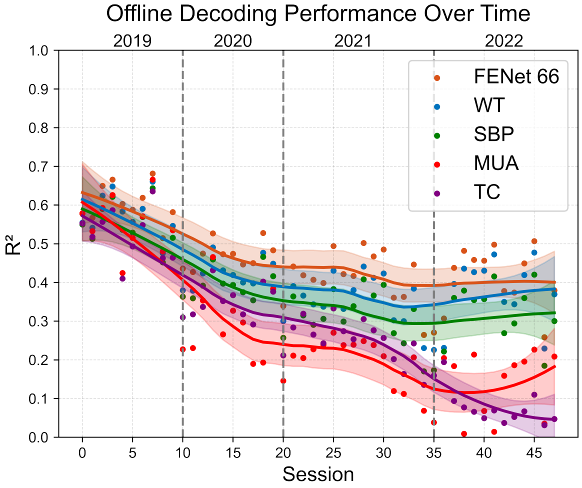


Fig. 10: Cross-validated decoder R^2 performance over four years post-implantation. Locally Estimated Scatterplot Smoothing (LOESS) fits and confidence intervals are shown for each feature type.

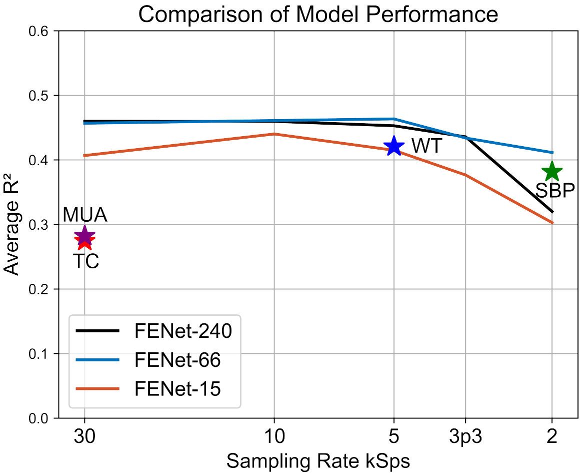
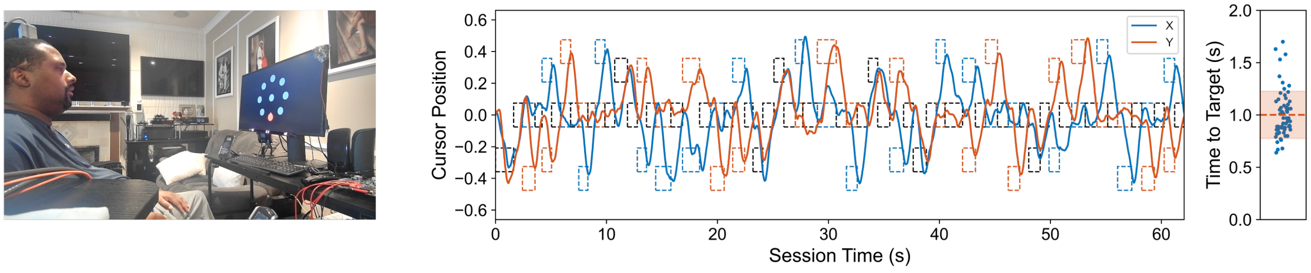


Fig. 11: Cross-validated decoder R^2 performance of FENet models versus sampling rate. The average R^2 performance of other features from Fig. 10 is shown as starred points for reference.



(A) (B) (C)

Fig. 12: (A) Research participant controlling a cursor utilizing ASIC for kinematic decoding in a center out task. (B) Online closed-loop decoding session using FENet ASIC in loop. Boxes represent the target where the height is the size of the target in its represented dimension, and the width represents the time it took to reach the target; color corresponds to the x and y dimension of the cursor control. (C) Time-to-target plot for all 63 targets with a mean time-to-target of 1.00 (seconds).

TABLE IV: Comparison with other state-of-the-art neural feature extraction ICs.

Metric	**This Work**	TBCAS22 [25]	BCAS19 [19]	TBCAS24 [38]	TBCAS22 [18]	TBCAS23 [15]	JNE25 [16]
Process	65	180	32	180	22	65	65
Implementation	Digital ASIC	Digital ASIC	Digital Sim.	Digital Sim.	Digital ASIC	Digital Sim.	Digital Sim.
Number of Channels	192	93	1	96	16	128	8
Channel Area μm^2	12801	28443$^\alpha$	2570000	20000	14000	6760	6450
Scaled Area$^\phi$ μm^2	12801	2370$^\alpha$	8481000	1667	92394	6760	6450
Channel Power μW	1.8	3.68	2.78	0.076	2.79	0.038	0.532
Resolution (bits)	9	16	6	10	8	1	1
Sampling rate (kHz)	5-30	2	24	24	20	7	24
Feature Type	FENet	SBP	MUA	LFP	MUA	TC	TC
Algorithm$^\gamma$	CNN	MAV	SS OSort	AE	SS	SD NEO	SD TEO & SWT
Avg. Feature R^2	0.446	0.382	0.275	-	0.275	0.282	0.282
Feature NPR	0.66	0.57	0.48	-	0.48	0.16	0.16
Feature Rate FPS	33	20	Async.	-	Async.	Async.	Async.
Bin Size (samples)	150$^\beta$	100	64	-	64	16	80
Supply Voltage (V)	0.63//0.9$^\beta$	0.625	1.16	1.8	0.63	1.8	1.2
Clock (MHz)	0.188$^\delta$	0.068$^\eta$	0.024	0.004	0.400	0.896	0.200
Latency (ms)	6.0	0.5	1.3	-	0.07	-	0.05
Validation Model	Human	Primate	Synthetic	Primate	Rat	Synthetic	Primate

$^\alpha$Calculated from feature extraction hardware only.
$^\beta$Configured for sampling rate of 5 kSps with FENet-66.
$^\delta$System clock frequency. Mac frequency for FENet-66 is 21x the system clock frequency.
$^\eta$Spiking band power feature extraction unit runnning at 2.9 MHz.
$^\phi$Scaled to 65 nm process using methods in [39]
$^\gamma$Mean Abolute Value (MAV); Auto Encoder (AE); Nonlinear Energy Operator (NEO); Teager Energy Operator (TEO); Stationary Wavelet Transform (SWT)

The FENet algorithm is trained on data from Utah arrays with large probe spacing (0.4 mm) and therefore does not see significant inter-channel correlations of spike signals from the same neuron [43], however, training of FENet with presence of inter-channel correlations is an interesting topic for future investigations. Spike detectors and spike sorters like those found in [15], [16], [18], [19] are able to distinctly identify neural sources, and firing patterns, which is well suited for systems with high inter-channel correlations, but require high SNR for accurate detection and sorting. The calibration free SD system in [15] has a power and area cost of 6760 μm^2 and 0.038 μW per channel, respectively. Their system employs adaptive thresholding techniques, which showed maintained detection accuracy for 200 days. However, chronically implanted neural probes used in our study have mean noise levels of 89% six years after implant, which is much higher than the 20% tested in [15]. Even with our most advanced adaptive thresholding techniques, FENet features outperform SD (TC features) by 487% after 4 years.

SPB calculated in [25] has remarkably low complexity in relation to its performance, achieving 3.68 μW and a scaled area of 2370 μm per channel. SBP features further reduce each bin of neural data down to a single feature. For a 30 ms time bin sampled at 2 kSps, this reduces the dimensionality of the neural data 60×. The simplicity of this algorithm accentuates its utility when power and area constraints are high and decoding precision is less pertinent.

Other neural interface modalities employ signal compression such that low data rate representations of the neural data can be transmitted then reconstructed with little or no loss. The system in [38], utilizes an Autoencoder (AE) to compress local field potentials (LFP) at 0.076 μW and 0.02 μm^2 per channel with a compression ratio of 19.2. The compression system is designed specifically for LFP as it relies on the spatially correlated nature of these signals. The system allows for lossy reconstruction of the original signal with a signal-to-noise distortion ratio of 15-19 dB.

While our system primarily focuses on feature extraction, the systems in [18], [25], [44], incorporate on-chip decoders to fully integrate the decoding pipeline. The system in [25] achieved an average cross validated correlation of 0.29-0.49 with 1D, and 2D kinematic decoding trials, respectively using a steady state Kalman filter. In [44], a decoder using distinctive neural codes and a linear discriminant analysis classifier were able to achieve 31-class handwriting classifications at 1 classification a second with 91.3% accuracy. This system was able to achieve this while only consuming $0.44\ \mu W$ and $1500\ \mu m^2$ per channel. Notably, their system employs gaussian smoothing on firing rates derived either from raw threshold crossings, or spike sorted features. Feature smoothing can help assuage high frequency fluctuations, and improve decoder performance as shown in our explorations in Appendix I.

VII. Conclusion

In this work we have developed a scalable, low power 1D CNN-based feature extraction ASIC optimized to process broadband neural data streams with low power and area overhead. Our architecture is optimized to implement the FENet algorithm on hundreds of continuous streams of data, which reduced memory requirements by 5× over conventional architectures. The algorithm implemented by this hardware generates neural features with state-of-the-art stability even six years post-implantation, where the maximum SNR of the MEAs over all 196 channels was 4.18. We validated the hardware-optimized FENet models and the ASIC that implemented them online through closed-loop cursor control with a human subject.

The power consumption of the ASIC was $341.2\ \mu W$ (5kSps) to generate neural features at 33 FPS for all 192 channels at a latency of 6 ms, fast enough for accurate and responsive kinematic decoding. The feature extraction hardware is highly power-scalable, providing flexibility to the decoding system. While the system does not dynamically reconfigure itself, the models, and channel counts could be updated by a central control system of a more complex SOC that integrates the FENet hardware into its neural decoding pipeline. Early in an implant's life-cycle, when only few informative channels are necessary, power can be optimized to achieve quality decoding performance with low drain on system power. Later in an implant's life-cycle, when noise is high, more channels can be enabled to maintain performance at a linear expense in power.

Scalable processors for 1D CNNs may prove useful in other many-channel signal processing applications, such as ultrasonic sensing. It is worth noting that data-specific choices made to suit the neural decoding environment may need optimization for these application.

Further improvement of the FSM control flow would alleviate the necessity for caching input data during the conclusional padding phase. Simply allowing input data to be written into the SRAM space that becomes unused during padding would build-in caching into the already available hardware. Additionally, unused SRAM kernel space could be used for storing PLSR and linear decoder weights, further expanding the hardware's capabilities by minimally adjusting the data-flow architecture and CNN control flow.

Brain machine interfaces have evolved expeditiously over the past few years. This work provides one important block of the neural decoding pipeline that significantly reduces the bandwidth of downstream decoding components, while maintaining much of the important information found in broadband neural data.

Appendix I
Model Parameter Exploration

In Fig. 13 A, we show the tradeoffs between the number of features, the total number of cycles required for each feature (solid line) and the number of those cycles that are necessary for padding (dashed lines). The minimum operating frequency of the system is related to the cycle count by equation (7):

$$f_{sys} = N_{features} * N_{cycles} \tag{7}$$

where f_{sys} is the minimum operating frequency, $N_{features}$ is the desired number of features per second, and N_{cycles} is the minimum number of cycles required of the model. Using this frequency, we can roughly determine the power and latency tradeoffs for a given feature rate based on Fig. 9 C.

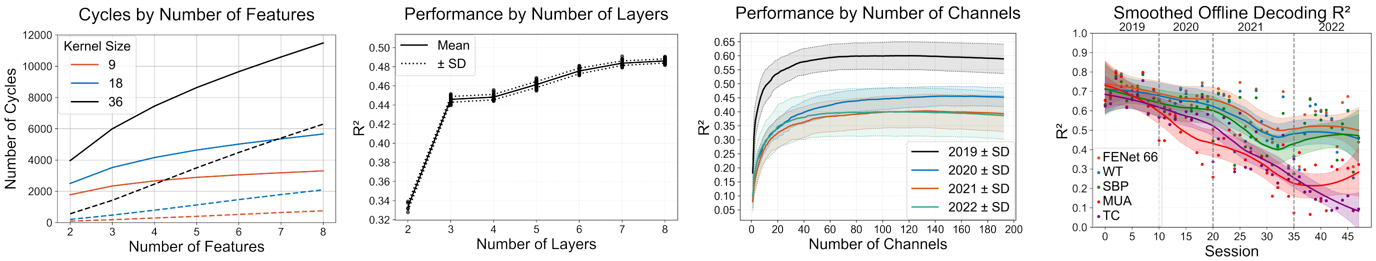
While the effect of kernel size on accuracy is highly non-linear, the number of layers can in some degree be related to decoder accuracy. We explore this effect in Fig. 13 B. We trained 51 models and held the kernel width and stride constant at 40 and 2, respectively, to tease out only the effect of the number of layers. We notice that there is a quasi-logarithmic effect on the number of layers to decoding accuracy, which reflects that the majority of neural information is captured in the lowest feature layers, with diminishing, but extant returns on performance as the number of layers is increased.

We further explore the effect of channel gating on this particular center-out decoding task in Fig. 13 C. We notice that early years require very few channels for high decoding accuracy, allowing for significant reduction in system resources. Later years often require more channels to achieve better performance.

Smoothing features prior to decoding alleviates effects of high frequency noise on decoders, which improves performance. In Fig. 13 D, we apply the gaussian kernel used in [44]. We scaled the window size and standard deviation by a factor of $\frac{1}{3}$ to accommodate the fact that our neural data bins were 3× larger.

Acknowledgment

The authors would like to thank JJ for their participation in online trials. This work was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2039655, the Center for Sensing to Intelligence (S2I), Tianqiao and Chrissy Chen Brain–machine Interface Center, and Heritage Medical Research Institute at Caltech. Funding for this Institutional Review Board- and FDA-approved work has been provided by the National Institute of Health grant R01EY015545 and UG1EY032039 (J.G., R.A.A.), Tianqiao and Chrissy Chen Brain–machine Interface Center at Caltech (J.G., R.A.A.), Swartz Foundation (J.G.) and Boswell Foundation (R.A.A.).



(A) (B) (C) (D)

Fig. 13: **(A)** Cycle count for models with various hyperparameters. Solid lines denote the total cycle count, dashed lines indicate padding cycles. Kernel sizes are constant for all layers. Bin size: 150. **(B)** Affect of the number of feature layers on decoding performance with a constant kernel size and stride of 40 and 2, respectively. R^2 from 10 days of training data only. **(C)** Effect of the number of channels on decoding performance. Shaded regions indicate the standard deviation of performance within the year. Colors indicate the year. **(D)** Cross-validated decoder R^2 performance over four years post-implantation with a gaussian kernel applied to features prior to decoding. Locally Estimated Scatterplot Smoothing (LOESS) fits and confidence intervals are shown for each feature type.

References

[1] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive Control Signals for Neural Prosthetics," Tech. Rep. 9, 2002.

[2] I. H. Stevenson and K. P. Kording, "How advances in neural recording affect data analysis," in *Nature Neuroscience*, vol. 14, pp. 139–142, 2 2011.

[3] M. A. Shaeri and A. M. Sodagar, "Data Transformation in the Processing of Neuronal Signals: A Powerful Tool to Illuminate Informative Contents," *IEEE Reviews in Biomedical Engineering*, vol. 16, pp. 611–626, 2023.

[4] C. Sponheim, V. Papadourakis, J. L. Collinger, J. Downey, J. Weiss, L. Pentousi, K. Elliott, and N. G. Hatsopoulos, "Longevity and reliability of chronic unit recordings using the Utah, intracortical multi-electrode arrays," *Journal of Neural Engineering*, vol. 18, 12 2021.

[5] D. A. Bjånes, S. Kellis, R. Nickl, B. Baker, T. Aflalo, L. Bashford, S. Chivukula, M. S. Fifer, L. E. Osborn, B. Christie, B. A. Wester, P. A. Celnik, D. Kramer, K. Pejsa, N. E. Crone, W. S. Anderson, N. Pouratian, B. Lee, C. Y. Liu, F. Tenore, L. Rieth, and R. A. Andersen, "Quantifying physical degradation alongside recording and stimulation performance of 980 intracortical microelectrodes chronically implanted in three humans for 956-2246 days.," *medRxiv : the preprint server for health sciences*, 9 2024.

[6] J. W. Salatino, K. A. Ludwig, T. D. Kozai, and E. K. Purcell, "Glial responses to implanted electrodes in the brain," *Nature Biomedical Engineering*, vol. 1, pp. 862–877, 11 2017.

[7] L. Iannucci, G. L. Barbruni, D. Ghezzi, M. Parvis, S. Grassini, and S. Carrara, "Changes Over Time in the Electrode/Brain Interface Impedance: An Ex-Vivo Study," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 17, pp. 495–506, 6 2023.

[8] M. Ferguson, D. Sharma, D. Ross, and F. Zhao, "A Critical Review of Microelectrode Arrays and Strategies for Improving Neural Interfaces," 10 2019.

[9] A. P. Buccino, C. L. Hurwitz, S. Garcia, J. Magland, J. H. Siegle, R. Hurwitz, and M. H. Hennig, "Spikeinterface, a unified framework for spike sorting," *eLife*, vol. 9, pp. 1–24, 10 2020.

[10] N. Ahmadi, T. G. Constandinou, and C. S. Bouganis, "Inferring entire spiking activity from local field potentials," *Scientific Reports*, vol. 11, 12 2021.

[11] B. Haghi, T. Aflalo, S. Kellis, C. Guan, J. A. Gamez de Leon, A. Y. Huang, N. Pouratian, R. A. Andersen, and A. Emami, "Enhanced control of a brain–computer interface by tetraplegic participants via neural-network-mediated feature extraction," *Nature Biomedical Engineering*, 2024.

[12] E. Stark and M. Abeles, "Predicting movement from multiunit activity," *Journal of Neuroscience*, vol. 27, pp. 8387–8394, 8 2007.

[13] S. R. Nason, A. K. Vaskov, M. S. Willsey, E. J. Welle, H. An, P. P. Vu, A. J. Bullard, C. S. Nu, J. C. Kao, K. V. Shenoy, T. Jang, H. S. Kim, D. Blaauw, P. G. Patil, and C. A. Chestek, "A low-power band of neuronal spiking activity dominated by local single units improves the performance of brain–machine interfaces," *Nature Biomedical Engineering*, vol. 4, pp. 973–983, 10 2020.

[14] M. Zhang, M. A. Schwemmer, J. E. Ting, C. E. Majstorovic, D. A. Friedenberg, M. A. Bockbrader, W. Jerry Mysiw, A. R. Rezai, N. V. Annetta, C. E. Bouton, H. S. Bresler, and G. Sharma, "Extracting wavelet based neural features from human intracortical recordings for neuroprosthetics applications," *Bioelectronic Medicine*, vol. 4, 12 2018.

[15] Z. Zhang, P. Feng, A. Oprea, and T. G. Constandinou, "Calibration-Free and Hardware-Efficient Neural Spike Detection for Brain Machine Interfaces," *IEEE Transactions on Biomedical Circuits and Systems*, 2023.

[16] Z. Zhou, Z. Hu, and H. Lyu, "A 0.53-μW/channel calibration-free spike detection IC with 98.8-%-accuracy based on stationary wavelet transforms and Teager energy operators," *Journal of Neural Engineering*, vol. 22, 4 2025.

[17] Z. Hu, Z. Zhou, and H. Lyu, "A Microwatt/Channel Neural Signal Processor for High-Channel-Count Spike Detection and Sorting," in *Proceedings - IEEE International Symposium on Circuits and Systems*, Institute of Electrical and Electronics Engineers Inc., 2024.

[18] S. M. A. Zeinolabedin, F. M. Schuffny, R. George, F. Kelber, H. Bauer, S. Scholze, S. Hanzsche, M. Stolba, A. Dixius, G. Ellguth, D. Walter, S. Hoppner, and C. Mayr, "A 16-Channel Fully Configurable Neural SoC With 1.52 μW/Ch Signal Acquisition, 2.79 μW/Ch Real-Time Spike Classifier, and 1.79 TOPS/W Deep Neural Network Accelerator in 22 nm FDSOI," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, pp. 94–107, 2 2022.

[19] D. Valencia and A. Alimohammad, "A Real-Time Spike Sorting System Using Parallel OSort Clustering," *IEEE transactions on biomedical circuits and systems*, vol. 13, pp. 1700–1713, 12 2019.

[20] M. A. Shaeri and A. M. Sodagar, "A method for compression of intra-cortically-recorded neural signals dedicated to implantable brain-machine interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, pp. 485–497, 5 2015.

[21] K. G. Oweiss, A. Mason, Y. Suhail, A. M. Kamboh, and K. E. Thomson, "A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 6, pp. 1266–1278, 2007.

[22] L. A. Koyrakh, S. J. Medical, and S. Paul, "Data Compression for Implantable Medical Devices," tech. rep.

[23] W. Biederman, D. J. Yeager, N. Narevsky, J. Leverett, R. Neely, J. M. Carmena, E. Alon, and J. M. Rabaey, "A 4.78 mm2 Fully-Integrated Neuromodulation SoC Combining 64 Acquisition Channels with Digital Compression and Simultaneous Dual Stimulation," *IEEE Journal of Solid-State Circuits*, vol. 50, pp. 1038–1047, 4 2015.

[24] Y. Ding, Y. Liu, and X. Liu, "An Iterative Spike Compression Method Based on Wavelet Transform and Heuristic Algorithm," in *2024 IEEE Biomedical Circuits and Systems Conference, BioCAS 2024*, Institute of Electrical and Electronics Engineers Inc., 2024.

[25] H. An, S. R. Nason-Tomaszewski, J. Lim, K. Kwon, M. S. Willsey, P. G. Patil, H. S. Kim, D. Sylvester, C. A. Chestek, and D. Blaauw, "A Power-Efficient Brain-Machine Interface System with a Sub-mw Feature Extraction and Decoding ASIC Demonstrated in Nonhuman

Primates," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, pp. 395–408, 6 2022.

[26] T. Wu, W. Zhao, H. Guo, H. H. Lim, and Z. Yang, "A Streaming PCA VLSI Chip for Neural Data Compression," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, pp. 1290–1302, 12 2017.

[27] T. Wu, W. Zhao, E. Keefer, and Z. Yang, "Deep compressive autoencoder for action potential compression in large-scale neural recording," *Journal of Neural Engineering*, vol. 15, 10 2018.

[28] K. A. Ludwig, R. M. Miriani, N. B. Langhals, M. D. Joseph, D. J. Anderson, and D. R. Kipke, "Using a common average reference to improve cortical neuron recordings from microelectrode arrays," *Journal of Neurophysiology*, vol. 101, pp. 1679–1689, 3 2009.

[29] H. S. Lee, K. Eom, M. Park, S. B. Ku, K. Lee, T. Kim, T. Kim, Y. M. Shon, H. Park, and H. M. Lee, "A Multi-Channel Neural Recording System With Neural Spike Scan and Adaptive Electrode Selection for High-Density Neural Interface," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, pp. 2844–2857, 7 2023.

[30] L. Biewald, "Experiment tracking with weights and biases," 2020. Software available from wandb.com.

[31] M. Nekoui and A. M. Sodagar, "Spike Compression through Selective Downsampling and Piecewise Curve Fitting Dedicated to Neural Recording Brain Implants," in *BioCAS 2022 - IEEE Biomedical Circuits and Systems Conference: Intelligent Biomedical Systems for a Better Future, Proceedings*, pp. 50–54, Institute of Electrical and Electronics Engineers Inc., 2022.

[32] S. Gibson, J. W. Judy, and D. Marković, "Technology-aware algorithm design for neural spike detection, feature extraction, and dimensionality reduction," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, pp. 469–478, 10 2010.

[33] A. P. Chandrakasan, "Ultra Low Power Digital Signal Processing," tech. rep., Bangalore, 1 1996.

[34] B. H. Gwee, J. S. Chang, Y. Shi, C. C. Chua, and K. S. Chong, "A low-voltage micropower asynchronous multiplier with shift-add multiplication approach," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 7, pp. 1349–1359, 2009.

[35] A. J. Bullard, S. R. Nason, Z. T. Irwin, C. S. Nu, B. Smith, A. Campean, P. H. Peckham, K. L. Kilgore, M. S. Willsey, P. G. Patil, and C. A. Chestek, "Design and testing of a 96-channel neural interface module for the Networked Neuroprosthesis system," *Bioelectronic Medicine*, vol. 5, 12 2019.

[36] C. Y. Zhang, T. Aflalo, B. Revechkis, E. R. Rosario, D. Ouellette, N. Pouratian, and R. A. Andersen, "Partially Mixed Selectivity in Human Posterior Parietal Association Cortex," *Neuron*, vol. 95, pp. 697–708, 8 2017.

[37] N. Y. Masse, B. Jarosiewicz, J. D. Simeral, D. Bacher, S. D. Stavisky, S. S. Cash, E. M. Oakley, E. Berhanu, E. Eskandar, G. Friehs, L. R. Hochberg, and J. P. Donoghue, "Non-causal spike filtering improves decoding of movement intention for intracortical BCIs," *Journal of Neuroscience Methods*, vol. 236, pp. 58–67, 10 2014.

[38] D. Valencia, P. P. Mercier, and A. Alimohammad, "Efficient in Vivo Neural Signal Compression Using an Autoencoder-Based Neural Network," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 18, pp. 691–701, 6 2024.

[39] A. Stillmaker, Z. Xiao, and B. Baas, "Toward More Accurate Scaling Estimates of CMOS Circuits from 180 nm to 22 nm," tech. rep., 2011.

[40] J. Loh and T. Gemmeke, "Dataflow Optimizations in a Sub-uW Data-Driven TCN Accelerator for Continuous ECG Monitoring," in *2022 IEEE Nordic Circuits and Systems Conference, NORCAS 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022.

[41] V. Jain, S. Giraldo, J. D. Roose, L. Mei, B. Boons, and M. Verhelst, "TinyVers: A Tiny Versatile System-on-Chip with State-Retentive eM-RAM for ML Inference at the Extreme Edge," *IEEE Journal of Solid-State Circuits*, vol. 58, pp. 2360–2371, 8 2023.

[42] C. Zhang, Z. Huang, C. Zhou, A. Qie, and X. Wang, "An Energy-Efficient Configurable 1-D CNN-Based Multi-Lead ECG Classification Coprocessor for Wearable Cardiac Monitoring Devices," *IEEE Transactions on Biomedical Circuits and Systems*, 2025.

[43] G. Buzsáki, "Large-scale recording of neuronal ensembles," *Nature Neuroscience*, vol. 7, pp. 446–451, 5 2004.

[44] M. A. Shaeri, U. Shin, A. Yadav, R. Caramellino, G. Rainer, and M. Shoaran, "A 2.46-mm2 Miniaturized Brain-Machine Interface (MiBMI) Enabling 31-Class Brain-to-Text Decoding," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 11, pp. 3566–3579, 2024.

Steven P. Bulfer (M'18) was born in Fairmont, Minnesota, USA in 1998. He received a B.Sc. in electrical engineering from the University of Minnesota, Twin Cities, in 2020 and an M.Sc. in electrical engineering from California Institute of Technology (Caltech) in 2022 where he is currently pursuing his PhD. in electrical engineering; with special interests in digital/mixed-signal IC design for brain machine interfaces.

He was awarded an NSF GRFP fellowship in the spring of 2021.

Jorge Gámez received the B.Sc. degree in Electronics System Engineering from the Instituto Tecnológico y de Estudios Superiores de Monterrey in 2000, and the M.Sc. and PhD degrees from Universidad Nacional Autónoma de México in 2011 and 2019, respectively. Since 2019, he has been a postdoctoral fellow at Caltech. His research interests include the neural mechanisms underlying complex behaviors, sensorimotor neurophysiology, and brain–machine interface technology.

Albert Yan-Huang is pursuing the B.S. degree in computation and neural systems at the California Institute of Technology, Pasadena, California.

From 2022 to 2024, he was an Undergraduate Student Researcher at the Mixed-mode Integrated Circuits laboratory at the California Institute of Technology in Pasadena, California. He also worked on power-optimizing the original FENet "Enhanced control of a brain–computer interface by tetraplegic participants via neural-network-mediated feature extraction " (Nature Biomedical Engineering, 2024). His research interests include efficient machine learning and stable decoding for brain-machine interfaces.

Benyamin Haghi received the B.S. degree in electrical engineering and mathe- matics from the Sharif University of Technology, Tehran, Iran, in 2016, and the M.S. and Ph.D. degree in electrical engineering from the California Institute of Technology (Caltech) in 2018, and 2024, respectively.

Volnei A. Pedroni received his Bachelor degree in Electrical Engineering from the Federal University of Rio Grande do Sul (UFRGS), Brazil, and both the MSc and PhD degrees in Electrical Engineering from the California Institute of Technology (Caltech). His area of expertise is Microelectronics, particularly VLSI design, FPGAs, and VHDL, in which he had a number of Master and PhD students at Federal Technological University of Parana State (UTFPR), Brazil. Besides scientific papers and other books, his publications include two books published by MIT Press: Circuit Design with VHDL and Finite State Machines in Hardware: Theory and Design, with VHDL and SystemVerilog. Upon retiring as a Full Professor from UTFPR in 2017, he taught regularly the course EE125 Digital Electronics and Design with FPGAs and VHDL at Caltech for several years. Besides the collaboration with Caltech, Dr. Pedroni did collaboration also with University of Trento and University of Modena, both in Italy.

Richard A. Andersen received the B.Sc. degree in biochemistry from University of California, Davis, California, USA in 1973 and Ph.D. in physiology from University of California, San Francisco, California, USA in 1979.

He is currently is the James G. Boswell Professor of Neuroscience and the T&C Chen Brain-Machine Interface Center Leadership Chair at Caltech. He was a faculty member of the Salk Institute and MIT before joining Caltech. Andersen discovered gain-fields, the method the brain uses to transform signals between spatial representations. He also discovered neural signals of intention, proving that they are not sensory in nature but rather reflect the planning of the subject. He has applied this discovery of intention to advance research in brain-machine interfaces, showing that paralyzed patients' intentions can be decoded from brain activity to control assistive devices such as robotic limbs.

Andersen is a member of the National Academy of Sciences, the National Academy of Medicine, and the American Academy of Arts and Sciences.

Azita Emami received the B.Sc. degree from Sharif University of Technology, Tehran, Iran, in 1996, and the M.Sc. and Ph.D. in electrical engineering from Stanford University, Stanford, CA, USA, in 1999 and 2004, respectively.

From 2004 to 2006, she was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. She joined the California Institute of Technology (Caltech), Pasadena, CA, in 2007, where she is currently the Andrew and Peggy Cherng Professor of Electrical Engineering and Medical Engineering and the Director of the Center for Sensing to Intelligence. She also served as the Executive Officer (Department Head) of Electrical Engineering from 2017 to 2024. Her current research interests include integrated circuits and systems, integrated photonics, wearable and implantable devices for neural recording, neural decoding, neural stimulation, sensing, and drug delivery.

Dr. Emami was an IEEE SSCS Distinguished Lecturer from 2017 to 2018. From 2016 to 2021, she was an Associate Editor of the IEEE Journal of Solid State Circuits (JSSC).