MINDFUL: Safe, Implantable, Large-Scale Brain-Computer Interfaces from a System-Level Design Perspective

Guy Eichler
Columbia University
New York, NY, USA
guyeichler@cs.columbia.edu

Yatin Gilhotra
Columbia University
New York, NY, USA
yatin.gilhotra@columbia.edu

Nanyu Zeng
Columbia University
New York, NY, USA
nanyu@ee.columbia.edu

Martha Kim
Columbia University
New York, NY, USA
martha@cs.columbia.edu

Kenneth L. Shepard
Columbia University
New York, NY, USA
shepard@ee.columbia.edu

Luca P. Carloni
Columbia University
New York, NY, USA
luca@cs.columbia.edu

Abstract

Brain-computer interface (BCI) technology is among the fastest growing fields in research and development. On the application side, BCIs provide a deeper understanding of brain function, inspire the creation of complex computational models, and hold significant promise for assisting individuals with disabilities. On the system side, BCIs have evolved from non-invasive, low-resolution wearable devices to invasive, high-resolution, implantable systems-on-chip (SoCs) that offer higher-quality brain data, enabling more effective exploration of brain activity. However, implantable BCIs must acquire large-scale neural signals and run real-time BCI applications, all while relying on wireless communication for practical use. Unlike typical devices, BCIs must operate within strict power constraints to ensure safety, which is crucial for their deployment in real-world applications. This requires careful co-design and a balanced approach across the key components of the BCI system.

In this work, we discuss why BCIs present unique design challenges compared to conventional computing systems. We develop equations based on the system-level structure of modern BCIs to estimate power consumption and explore trade-offs among key system components: data acquisition, on-chip computation, and wireless communication. Using these equations, we analyze BCI SoC designs that support wireless communication and examine how scaling trends, design constraints, and optimization strategies may impact the feasibility of future BCIs. Specifically, we show a clear discrepancy between certain cutting-edge, BCI-centric computations and the feasibility of their on-chip integration in power-constrained BCI systems, revealing a significant gap between the development of deep learning methods for BCI and the design of safe BCI systems. However, with targeted optimizations in BCI system design and greater specialization for specific applications, future BCI systems will be able to successfully integrate modern BCI applications and advance toward widespread adoption.

MICRO '25, Seoul, Republic of Korea

ACM ISBN 979-8-4007-1573-0/2025/10
https://doi.org/10.1145/3725843.3756032

CCS Concepts

• **Computer systems organization → Special purpose systems**; **Embedded systems**; • **Hardware → Biology-related information processing**; **Application-specific VLSI designs**.

Keywords

Brain-Computer Interface (BCI), Implantable, Wireless Communication, Neural Interface, System-on-Chip, Deep Neural Network.

ACM Reference Format:
Guy Eichler, Yatin Gilhotra, Nanyu Zeng, Martha Kim, Kenneth L. Shepard, and Luca P. Carloni. 2025. MINDFUL: Safe, Implantable, Large-Scale Brain-Computer Interfaces from a System-Level Design Perspective. In *58th IEEE/ACM International Symposium on Microarchitecture (MICRO '25), October 18–22, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3725843.3756032

1 Introduction

Brain-computer interfaces (BCIs) create a connection between the brain and external devices [34, 66, 123]. They serve various purposes, such as acting as medical assistive technologies [73, 75, 81, 95, 124], and supporting the development of brain-inspired artificial intelligence (AI) [14, 67, 78, 79, 140].

With the human brain containing approximately 100 billion neurons, modern BCIs aim to enhance the scale and resolution of the sensing technology, known as the *neural interface* (or NI) [16, 64, 113, 119, 136, 139]. The ability of NIs to perform massively parallel recordings of neuronal activity through sensors, called *channels*, has doubled roughly every seven years [113, 119]. BCIs have evolved from relying on low-resolution, non-invasive electroencephalography (EEG) channels [2, 26, 117, 121] to integrating large-scale, high-resolution and implantable NIs [20, 82, 125, 126, 135, 136]. Invasive BCIs capable of supporting large numbers of NI channels are now at the forefront of research and development [51, 98, 125, 136, 139].

Fig. 1 illustrates a modern BCI system. It consists of an implanted system-on-chip (SoC) that, at a minimum, acquires data from the NI, and transmits it wirelessly to a non-implanted wearable SoC [35, 51, 115, 136]. Some BCI systems integrate computation into the implant [62, 97, 111], thereby reducing wireless communication demands, as the output of the computation is typically much smaller than the raw neural data. Other BCI systems integrate computation into the wearable [29, 30, 129], assuming that reducing wireless

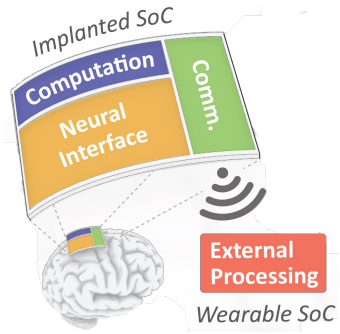


Fig. 1: Future implanted BCI SoCs pose a challenging design task. They need to manage increasing neural data rates, wireless communication, and complex deep-learning computational tasks while adhering to strict power density limits to prevent cellular damage to surrounding brain tissue.

communication is not yet necessary, and taking advantage of the more relaxed power constraints of wearable devices.

To date, no large-scale, implantable, computation-capable BCI system has been successfully tested in clinical trials, proven to be safe, and made available to the public as an assistive technology. One possible reason for this is that modern implanted BCI SoCs present a complex and restrictive design space. On one hand, they must evolve to handle increasing neural data volumes, wireless communication, and computationally intensive deep-learning applications [11, 75, 108]. On the other hand, they must limit power consumption for safety reasons, as the surrounding brain tissue cannot experience a temperature increase of more than 1–2 °C to prevent cellular damage [101, 128]. Designing BCI systems within this envelope requires a thorough understanding of the interplay between the primary components of the implant.

To address this, we present an analytical framework that captures these interactions. Our framework, derived from an extensive review of the literature on modern implantable BCIs, projects how existing designs scale as the number of channels increases. Through first-order system-level analysis, we examine how scaling the neural interface impacts the overall design of the implanted SoC, focusing on the interconnected stages of sensing, computation, and communication on the implant. Our evaluation of how published BCI SoCs scale leads to the following insights:

- To stream raw neural data at higher rates, scaling communication components with channel count would either exceed safety limits or reduce sensing capacity, limiting system feasibility.
- Advanced modulation schemes in the wireless transceiver can help support higher transmission data rates, but achieving this in practice faces significant design challenges.
- Modern computation with deep neural networks (DNNs), as used in advanced BCI applications, is unlikely to be integrated into current implanted SoCs without major optimizations.
- Partitioning DNNs between parts of the BCI system can help integrate more channels in the short term. However, to achieve long-term scalability, DNN models should be better customized for the constraints of implanted SoCs.
- Bridging the gap between computational demands and safety requirements requires tailoring BCI systems to application needs and adapting applications to system constraints.

2 Background

BCI systems support a wide variety of applications. ***Online applications*** focus on real-time brain data analysis to control external devices [23, 27, 70, 124]. These applications decode brain signals in real-time for use in areas like motion, vision, and speech. For example, brain data from the motor cortex can be decoded into commands to control a prosthesis or move a cursor on a screen [23, 108, 124, 134]. Brain activity in the anterior speech area can be used to synthesize audio [8, 11, 76], and neural data from the visual cortex can be decoded into shapes and images to assist with vision loss [61, 75, 120].

Real-time processing is crucial for online applications, and the system must detect, interpret, and respond to brain activity before the user perceives any delay [22]. The definition of "real-time" in BCI varies. Some define it as completing the entire application within the reaction time of the brain, roughly 0.18 seconds [29, 30], while others define it as completing a task within a specific time window determined by the application [129] or by processing data at a rate that matches the sampling rate of the NI [62, 111].

Offline applications collect and analyze large amounts of brain data to study brain functions and structure [25, 37, 137]. While offline applications do not need real-time processing, they require high-quality and diverse neural data to yield meaningful insights. The quality of the data depends on the type of NI and the number of its channels. Increasing the channel count of the NI is essential for developing new offline BCI applications, advancing brain research, and ultimately improving online applications as well.

Implantable BCIs typically follow the structure shown in Fig. 1 and consist of three main subsystems: the NI, which contains sensors that interact with the brain and its neurons; computation in the form of a digital data-processing pipeline; and communication with a radio frequency (RF) transceiver that primarily transmits neural data from the implant to a nearby wearable device. In the following subsections, we outline the key design constraints and the current state of the art for each of these subsystems.

2.1 Neural Interfaces

The primary objective of a neural interface (NI) is to capture high-quality, accurate brain data, which is essential for understanding neuron activity [126]. Another key goal is to ensure the NI provides reliable brain data over long periods, without degrading the signal quality. This requires both durable NI construction and minimal immune response from the brain tissue. Additionally, a good NI should have high-throughput and large-scale recording capabilities. This allows for the monitoring of many neurons at once, which helps researchers better understand brain functions [113, 119, 126]. The ability to distinguish individual neurons from large populations is also valuable for gaining insights into brain activity [83].

NIs can be sorted into two classes: *non-invasive* and *invasive*. Non-invasive NIs typically use Electroencephalogram (EEG) technology, which relies on large electrodes placed on the scalp to capture the electrical activity from large populations of neurons [2, 3, 121]. This method avoids surgery, making it a popular choice. However, non-invasive NIs typically utilize a small number of channels aggregating data from many neurons. This results in low resolution brain data and limits the quality of the signal available for

downstream analysis. For this reason, these NIs are best suited for higher-level tasks such as analyzing emotions and general cognitive state [4, 72, 133], but are less reliable for tasks that require multidimensional control, such as motion or vision [46, 100].

Invasive NIs interface directly with brain tissue, enabling higher-resolution data acquisition. Many implants use microelectrodes to sense electrical activity. Microelectrodes can be introduced in a variety of ways. One design features a shank that penetrates the brain cortex, with electrodes placed on its surface [138]. Another involves multiple wires that carry electrodes, strategically inserted into specific cortical regions [136]. Both technologies require craniotomy and brain penetration, which carry risks, especially when removing the device. A less invasive approach places an endovascular stent with microelectrodes positioned within the blood vessels of the brain [87]. While this method avoids the need for a craniotomy, it is limited in terms of electrode placement and the resolution of the neural data. Finally, a very promising family of minimally invasive NIs use electrocorticography (ECoG) with microelectrode arrays (MEAs) placed on the cortex of the brain [51, 115, 139]. Because the MEA lies on the brain surface, it reduces surgical risks and simplifies its removal. By increasing the density and number of electrodes, high-resolution brain data can be achieved [64].

A unique group of invasive NIs uses light to track neuronal activity. Instead of MEAs, they use neural imagers that integrate single-photon avalanche diodes (SPADs) to detect photons emitted by neurons during spiking activity [42]. In this case, neurons are modified via optogenetics to express light-emitting indicators [135]. With sufficient resolution and advanced light filtering, neural imagers allow for precise mapping of neuronal circuits in the brain.

2.2 Wireless RF Communication

BCI systems used in research commonly transmit neural data via wired connections, which limit mobility and reduce their real-world use outside of research labs, particularly in humans [56, 89, 112, 125]. The introduction of wireless technologies has resolved these issues, allowing BCIs to be used in practical applications like prosthetics and rehabilitation. However, standard wireless technologies, such as Wi-Fi and Bluetooth, are not suitable for implanted BCIs because they either consume too much power or cannot handle the required transmission data rates. Consequently, most BCIs require custom-designed RF protocols to meet the stringent energy, performance, and thermal requirements [42, 52, 57, 58, 139].

2.3 Computation for BCIs

Traditionally, BCI applications have used linear control algorithms, such as the Kalman and Wiener filters [31, 43, 114, 131]. However, new methods that use machine learning (ML), particularly non-linear deep neural networks (DNNs), offer higher accuracy for decoding brain data in BCI applications [43, 53, 108]. As more ML-based BCI applications emerge, they often face the "curse of dimensionality" [113, 127]. As the dimensionality of input data increases, the size and complexity of DNNs grow [113, 127]. This leads to computational demands that increase faster than the input data, making it more resource-intensive [69, 113]. As neural recordings are acquired from many channels simultaneously, continued increases in channel count are expected to further raise the computational demands of cutting-edge BCI applications [32, 96, 113, 118, 119].

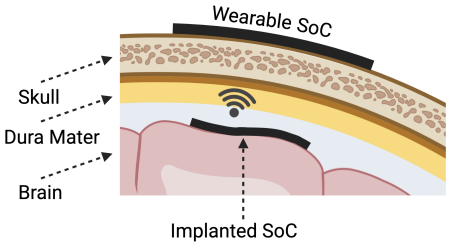


Fig. 2: In this paper, we focus on implantable BCI systems, where an implanted SoC is positioned on the cortex of the brain in the subdural space, between the dura mater and brain tissue. The SoC senses neural activity and transmits it via RF to a wearable SoC located outside the skull. In our study, the extent and type of processing performed on the implanted SoC remain an open design question, influenced by the constraints imposed by the biological context and analog system components.

The development of heterogeneous system-on-chip (SoC) architectures provides a design solution for running time-sensitive applications with lower resource usage on constrained devices [7, 21, 26, 38, 40, 48, 77]. Since modern BCIs rely on constrained implanted and wearable devices, some BCI SoC architectures are designed to be self-contained, integrating the neural interface, wireless communication, and some computation engines [29, 30, 62, 111, 129]. For non-invasive systems, such SoCs are ideal because the device is external to the body, and power consumption is only limited by the power source [122, 129]. However, for invasive systems, the power consumption of the implant must be carefully managed to avoid causing cellular damage in the brain [62, 101, 111, 128].

We can organize implanted BCI systems into two groups based on the type of computation performed on the implanted SoC. The first group limits on-implant computation to preparing the neural data for wireless transmission. In this case, the data transmitted consists of digitized raw neural data, and the communication transceiver must support a data rate that accommodates the total number of channels and the sampling rate of the NI [136, 139]. Any further computation or analysis is performed on the wearable SoC [26, 29, 117]. The second group performs some application-level computation on the implanted SoC. In this case, the final output of the computation, which is much smaller in size than the raw neural data, is transmitted to the wearable SoC, significantly reducing the amount of data that needs to be transmitted [62, 111, 118].

3 The Target BCI System

In this paper, we present MINDFUL, an analytical approach to capturing the constraints and requirements of the analog and biological context in BCI, enabling computer architects to better understand the design space of implantable architectures within the appropriate throughput, power, and area envelopes. Specifically, we analyze modern BCI systems utilizing an invasive NI.

The introduction of computation into implanted SoCs has become a subject of interest [54, 96, 97, 109–111]. However, any digital computation within the system must be part of a balanced pipeline that works seamlessly with two analog-based subsystems: the NI and the wireless RF communication. Moreover, this entire mixed analog and digital pipeline must adhere to strict safety criteria.

3.1 System Assumptions

Form Factor. Fig. 2 presents the implanted SoC as a flat surface that interfaces directly with the brain tissue. It is placed on the cortex of the brain, within the subdural space between the dura mater and the brain tissue. This is based on a very promising form factor that balances invasiveness and data quality and is used for a number of recent NI designs [42, 51, 56, 59].

This form factor offers several advantages over other invasive approaches. First, post-surgery, the skull is closed and heals naturally, maintaining the integrity of the native biological structure [85]. Second, there are no slits or active openings in the skull, as wireless communication eliminates the need for wires to be routed out from the implanted SoC [112]. Third, the implanted SoC directly interfaces with the cortex, engaging large populations of neurons and enabling a higher throughput of neural data [87]. However, for the purpose of our analysis, the implant can easily be adapted to other, less conventional NI shapes, as long as the total surface area in contact with brain is known. This is because heat dissipation and temperature increase primarily depend on that contact area [101].

Data Flows on the Implanted SoC. Each implanted SoC follows a predefined data pipeline that begins with the sensing of neural data at the NI and ends with the wireless transceiver. In between these endpoints, we are agnostic about the type of computation performed. However, we outline two general data-processing strategies that we later analyze in Section 5.

Today, most implanted BCI SoCs perform only the computation required to digitize the raw neural data and packetize it for wireless transmission. In this scenario, the transceiver must support a data rate that matches both the channel count and sampling rate of the NI. We call this the ***communication-centric*** dataflow. An alternative approach is to embed application-specific processing of the raw neural data on the implanted SoC, with the output of the computation transmitted by the transceiver. In this ***computation-centric*** dataflow, the computation must accommodate the raw neural data throughput of the NI, but typically reduces the volume of data to be transmitted. This results in a tradeoff, where more power is spent on computation while transceiver power and transmission data rate are reduced. Fig. 3 presents an illustration of the two data flows.

For BCI applications that aim to control external devices, the implanted SoC must frequently transmit neural data or computation results to the wearable SoC. The receiving direction is primarily used intermittently for tasks such as configuring the SoC or establishing communication protocols [42, 139]. The impact of the receiving direction on the overall architecture is assumed to be negligible compared to the transmitting direction.

3.2 System Requirements

Temperature and Power Density. The implanted SoC must be designed for long-term use with biocompatible materials and must not damage sensitive brain tissue through heat dissipation. Brain tissue is highly sensitive to temperature changes [45, 101], yet it benefits from one of the highest blood-flow rates in the body, which helps dissipate heat [128]. However, more research is needed to understand the effects of prolonged low-temperature heating on the brain. Preliminary studies suggest that an increase in temperature of up to 1–2 °C, approximately 39 °C, may be the upper limit of

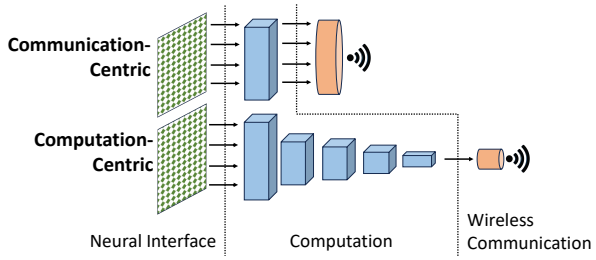


Fig. 3: Communication-centric implants allocate most of their resources to the neural interface and transceiver, while computation-centric implants process the raw data locally, dedicating more resources to computation and reducing communication requirements.

safety for the brain [128]. The power consumption of an implanted SoC results in heat dissipation from its surface area when a higher power consumption leads to increased heat. Increasing the surface area of the implant reduces its power density, helping to mitigate the rise in temperature around the device [99]. Due to blood flow, a power density of $P_d = 40 \frac{mW}{cm^2}$ is considered the upper limit for an implanted device [128]. Given the maximum allowable power density and the chip surface area, we estimate the maximum safe total power consumption, which we refer to as the ***power budget***.

This estimate assumes uniform power consumption and heat dissipation across the chip, which is typically not the case in integrated circuits, where non-uniform on-chip activity can cause hotspots. However, because the thermal conductivity of brain tissue is significantly lower than that of silicon, heat spreads more rapidly across the chip than into surrounding tissue [33, 130]. As a result, it is reasonable to assume uniform heat dissipation from the implant surface, even when power consumption is non-uniform. This aligns with the work of Serrano *et al.* [101], who propose a thermal model based on uniform heat dissipation for skull implants.

Flexibility and Area. Given the non-flat structure of the brain, ensuring stable contact with cortical neurons requires the implanted SoC to be flexible [115]. This flexibility can be achieved by thinning the chip after fabrication or using conformable materials [47, 105].

Furthermore, efficient use of chip area is important to support physical contact between the chip and, ideally, all neurons at the implant surface. To achieve this, the design goal is one channel per neuron with no more than 20 μm spacing between channels [64]. As the number of NI channels increases, the implant area must scale efficiently in such a way as to maintain sensing contact with the neurons. We address this requirement by maximizing ***volumetric efficiency*** [42, 59, 83]. This metric captures how effectively an implant places sensors in contact with neurons. At a basic level, high volumetric efficiency requires that the sensing area grow more rapidly than the non-sensing area. At a deeper level, higher volumetric efficiency demands that the sensing area become more compact to approach the 20 μm channel spacing constraint [64].

4 Power, Area, and Throughput Scaling

This section presents a simple analytical framework to address the primary implant design constraints of area, power, and throughput. Since BCI research typically reports these metrics on a per-channel

basis [96, 125, 136, 139], we adopt a similar approach, expressing them as functions of the number of channels to describe the high-level area, power, and throughput envelope for the implanted SoC.

For area and power, we define two scaling regimes: the first scales existing implanted SoC designs to the modern standard of 1024 channels (Section 4.1), while the second explores scaling these designs beyond 1024 channels (Section 4.2). Throughput scaling is addressed separately in Section 4.3.

4.1 Scaling Implants to 1024 Channels

Our analysis is based on reported area and power measurements of 11 implanted SoCs. Table 1 lists these designs, most of which have been experimentally validated either *in vivo* (live subjects) or *ex vivo* (biological tissue). Designs numbered 1–8 integrate wireless communication and fall within the scope of our target system presented in Fig. 2. Designs 9–11 are wired and do not support wireless communication. We include them in this initial analysis to demonstrate that they meet the power budget requirements defined in Section 3.2. However, we exclude them from later analyses because their transcutaneous wires introduce infection risks, restrict patient mobility, and make them impractical for widespread adoption [106].

Each SoC supports a specific number of active channels that can record neural data in parallel (denoted in the table as #Channels). For all designs, we consider only the SoC area that is expected to be in direct contact with brain tissue.

As the current standard for NI channel count is approximately 1024 [56, 65, 119, 136, 139], we start by scaling each SoC in Table 1 to 1024 channels. Some designs already meet this standard, namely SoCs 1, 3, and 10. SoCs 2 and 11 integrate a SPAD-based NI capable of recording from up to 49K channels, with a configurable sampling rate that can be reduced for larger-scale interfaces. The performance and data quality of these NIs are still under investigation. Therefore, for a fair comparison with electrode-based NIs, we use their nominal parameters for a 1024-channel configuration.

For the remaining designs, extrapolating the reported area and power to higher channel counts is non-trivial. We perform this scaling carefully, while respecting power density and volumetric efficiency requirements.

Simmich *et al.* [107] demonstrate that total power consumption in implantable BCIs scales roughly linearly with the number of channels, assuming constant signal quality as measured by the noise efficiency factor (NEF). Their analysis focused on amplifier topologies in current designs and did not account for second-order effects associated with high channel counts, such as increased design complexity, analog-to-digital conversion, or on-chip routing. Nonetheless, this model serves as a useful first-order approximation given the limited data on large-scale implanted BCI SoCs. For area, low-channel-count devices often suffer from poor volumetric efficiency due to wide inter-channel spacing. To maintain tighter spacing in high-channel-count designs, we assume area scales with the square root of the number of channels. Thus, for n channels, we estimate total area A_{SoC} and power P_{SoC} as:

$$A_{SoC}(n) \approx \sqrt{n} \cdot A_{\text{channel}}, \quad P_{SoC}(n) \approx n \cdot P_{\text{channel}} \tag{1}$$

where A_{channel} and P_{channel} are the reported area-per-channel and power-per-channel of the original SoC.

Table 1: Summary of implanted SoC Designs

#	SoC	NI Type	#Channels	Area (mm^2)	P_d ($\frac{mW}{cm^2}$)	f (kHz)	Wireless	In-Vivo/ Ex-Vivo
1	BISC [59, 139]	Electrodes	1024	144	27	8	Yes	Yes
2	Gilhotra *et al.* [42]	SPAD	49152	144	33	8	Yes	Yes
3	Neuralink [85, 136]	Electrodes	1024	20	39	10	Yes	Yes
4	Shen *et al.* [104]	Electrodes	16	1.34	2.2	10	Yes	Yes
5	Muller *et al.* [84]	Electrodes	64	5.76	2.5	1	Yes	Yes
6	Yang *et al.* [132]	Electrodes	4	4	1.3	20	Yes	Yes
7	WIMAGINE [80]	Electrodes	64	1960	3.8	30	Yes	Yes
8	HALO [62, 110]	Electrodes	96	1	1500	30	Yes	No
9	Neuropixels [125]	Electrodes	384	22	21	30	No	Yes
10	Jang *et al.* [56]	Electrodes	1024	3	17	20	No	Yes
11	Pollman *et al.* [89]	SPAD	49152	50	36	8	No	Yes

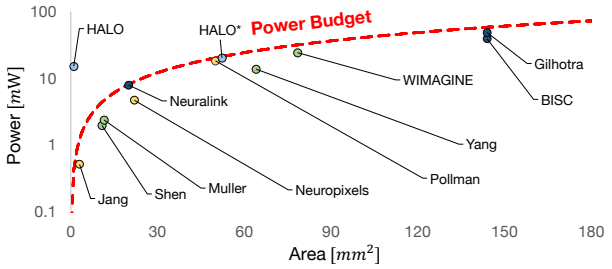


Fig. 4: Power and area results for the implanted SoC designs from Table 1, scaled to the current standard of 1024 channels using the method described in Section 4.1. Most designs demonstrate safe power density, with total power consumption remaining within the power budget.

However, this scaling does not always yield viable designs. For example, scaling SoC 5 to 1024 channels produces a power density of approximately 10 $\frac{\text{mW}}{\text{cm}^2}$, which we consider unrealistically low and indicative of area inefficiency. To address this, we apply an additional 2× reduction in area, resulting in a more plausible power density of 20 $\frac{\text{mW}}{\text{cm}^2}$. SoC 7 (WIMAGINE [80]) is considerably large for a 64-channel device. Scaling it to 1024 channels using Equation (1) yields an impractical design. A 2× area reduction results in a power density of 30 $\frac{\text{mW}}{\text{cm}^2}$, but channel spacing remains around 2 mm, far above the resolution requirement for high-density interfaces. To achieve a more realistic spacing of ~200 μm while preserving power density, we apply a 50× reduction in both power and area, modeling a more evolved design.

Lastly, there were two other special cases. SoC 9 (Neuropixels [91, 112, 125]) is one of the most widely used devices in BCI research. The NI shape is a group of shanks that penetrate the brain. In Table 1, we report the number of channels for a single shank; however, in this case, area and power are scaled linearly, rather than using Equation (1), since the design scales by simply adding more shanks. The second case is SoC 8 (HALO [110]), whose reported power density was too high for safe implantation. To bring it within the power budget, we scaled down both area and power, resulting in a modified version that we refer to as HALO*.

Fig. 4 depicts the area and power for each of the SoCs once scaled to 1024 channels. All designs fall below the red line representing the power budget, indicating compliance with a safe power density.

Our method for scaling to 1024 channels is not meant to precisely predict future SoC designs. Instead, it generates a set of plausible design points, based on the few existing 1024-channel SoCs (SoCs 1, 2, and 3). These points serve as a foundation for exploring how such designs could scale beyond 1024 channels.

4.2 Scaling Beyond 1024 Channels

To project the total area A_{SoC} and total power P_{SoC} for designs supporting $n > 1024$ channels, where n is the number of channels in the NI, we separate these totals into sensing and non-sensing functions, with the latter comprising communication and computation:

$$\begin{aligned} A_{SoC}(n) &= A_{sensing}(n) + A_{non\text{-}sensing}(n) \\ P_{SoC}(n) &= P_{sensing}(n) + P_{non\text{-}sensing}(n) \end{aligned} \tag{2}$$

Additionally, the design must never exceed the safe power density as described in Section 3:

$$\frac{P_{SoC}(n)}{A_{SoC}(n)} \leq 40\frac{mW}{cm^2}, \quad P_{budget}(n) = A_{SoC}(n) \cdot 40\frac{mW}{cm^2} \tag{3}$$

To improve volumetric efficiency as channel count increases, we aim for a larger proportion of the SoC area to be devoted to sensing:

$$\lim_{n\to\infty} \frac{A_{\text{sensing}}(n)}{A_{\text{SoC}}(n)} \approx 1 \tag{4}$$

When scaling designs to $n > 1024$, we assume that the power and area for sensing scale linearly with the number of channels in the NI. This aligns with prior work, which also applies linear scaling of power and area as a first-order estimate for implanted devices [107]. Additionally, inspection of existing large-scale SoC designs, such as those in Table 1, shows that they are generally symmetric around the integrated channels. This symmetry allows us to estimate the sensing area and power as:

$$\begin{aligned} A_{sensing}(n) &= n \cdot \frac{A_{sensing}(1024)}{1024} \\ P_{sensing}(n) &= n \cdot \frac{P_{sensing}(1024)}{1024} \end{aligned} \tag{5}$$

The area and power associated with non-sensing functions also scale with n, but not necessarily in a linear fashion, as they involve both computation and communication. Beyond 1024 channels, we cannot assume that the communication and computation implemented on the implanted SoC will scale smoothly with the number of channels. This is primarily because most current designs use communication-centric dataflows, relying on custom transceivers designed for specific data rates. As a result, scaling $P_{non\text{-}sensing}$ requires evaluating various design choices, including computation-centric dataflows. We explore these choices in detail in Section 5.

4.3 Real-Time Throughput

The non-sensing functions on the implant, namely computation and communication, must keep up with the data rates produced by the sensory channels. Neural data is sampled from all channels (n) at a specific sampling frequency ($f = \frac{1}{t}$), typically in the 1–30kHz range [11, 17, 108, 134]. Thus, the sensing data throughput is:

$$T_{sensing}(n) = \frac{d \cdot n}{t} \tag{6}$$

where d is the digitized sample bitwidth. This throughput sets a real-time constraint on the design.

The non-sensing components of the implanted SoC must process incoming neural data at the rate it is received. The computation stage may alter the data volume, which we capture using a computation-specific parameter, n_{out}, representing the number of output values produced by computation.

In a communication-centric architecture, the computation stage is limited to packetizing raw data. Assuming that the number of digitized values produced by computation is approximately equal to the number of NI channels ($n_{out} \approx n$), both computation and communication operate at a data throughput of:

$$T_{comp}(n) \approx T_{comm}(n) \approx T_{sensing}(n) \tag{7}$$

A computation-centric architecture may include a more complex data-processing pipeline. While the computation stage must still process input data at the same rate as it is produced, it typically reduces the volume of output data significantly, such that $n_{out} \ll n$. As a result, the required communication data throughput becomes:

$$T_{comm}(n_{out}) = \frac{d \cdot n_{out}}{t} \tag{8}$$

Ultimately, the design of computation and communication for a BCI implant must adhere to throughput requirements, as well as the area and power constraints discussed in Section 4.2.

In Section 5, we explore various approaches to communication and computation, identifying promising strategies and highlighting areas where technical improvements could be most impactful.

5 Communication and Computation Beyond 1024 Channels

We begin our study of future implanted SoCs that support more than the current standard of 1024 channels by focusing on communication-centric architectures, which are currently the most commonly used in demonstrated devices. We then explore the alternative of computation-centric architectures, exploring how these SoCs might evolve beyond 1024 channels and assessing their potential benefits and limitations.

5.1 Communication-Centric Architectures with Energy-Efficient Modulation

In the communication-centric scenario, all neural data is transmitted from the SoC after digitization and packetization. The computation is limited to packetization, meaning that $n_{out} \approx n$. As a result, the required communication throughput is $T_{comm}(n) \approx T_{sensing}(n)$. The on-implant transceiver must be capable of supporting this required transmission data rate, and the SoC must also include an antenna with sufficient bandwidth to accommodate it. Additionally, the transceiver must implement a modulation technique that effectively utilizes this bandwidth and meets the required data rate.

Energy-efficient modulation techniques, such as On-Off Keying (OOK), are currently preferred in implanted SoCs [52, 58, 139]. In OOK, each symbol transmitted can encode a maximum of 1 bit of information per clock cycle, as long as the transmission data rate is optimized to fully utilize the bandwidth of the antenna [92]. For instance, if the antenna supports a bandwidth of 100 MHz, an ideal OOK transceiver could theoretically transmit up to 100 Mbps.

In practice, an OOK transceiver is customized for a sufficient signal-to-noise ratio and maintain a constant energy per bit (E_b), up to a data rate below the ideal bandwidth. This gap is due to practical design challenges in fully utilizing the antenna bandwidth. For example, a transceiver that was customized to a system targeting exactly $E_b = 50\frac{\text{pJ}}{\text{bit}}$, $n = 1024$ channels, $d = 10$ bits per sample, and

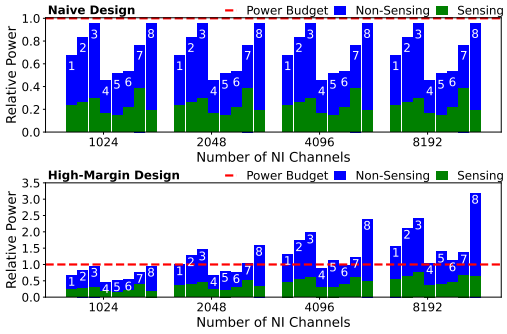


Fig. 5: SoC power relative to the power budget versus number of channels for two design scenarios: naive design and high-margin design. The naive design provides an advantage, with SoC power scaling linearly with the power budget.

a sampling frequency of $f = 8$ KHz, would support a transmission rate of 82 Mbps, even if the antenna bandwidth is 100 Mbps.

For communication-centric architectures using OOK modulation, the communication power can be expressed as:

$$P_{comm}(n) = T_{comm}(n) \cdot E_b = \frac{d \cdot n}{t} \cdot E_b \tag{9}$$

However, the original number of channels in a design, prior to scaling, may already correspond to the maximum transmission data rate that the customized transceiver can support while maintaining a constant energy per bit (E_b). In this case, Shannon's limit suggests that further increasing T_{comm}, which effectively represents the transmission data rate, would likely result in an increase in E_b and diminishing returns in terms of power consumption [44, 92].

To explore how communication-centric implants scale with the number of channels, we consider two design hypotheses:

- In what we refer to as a ***naive design***, the original transceiver is not designed to support a higher data rate (T_{comm}). As a result, both $A_{non\text{-}sensing}$ and $P_{non\text{-}sensing}$ must increase to accommodate the transmission of the added data volume and higher bandwidth requirements. We consider each channel as an independent unit, with its own dedicated non-sensing components. In this design, each additional channel contributes an increment to both the non-sensing power-per-channel and the non-sensing area-per-channel, similar to how sensing components scale as described in Equation (5). At larger scales, the naive design is effectively equivalent to scaling the number of implanted SoCs.
- In what we call the ***high-margin design***, we assume that the transceiver and antenna are able to support an f well above the one originally reported for the SoC. In such optimistic designs, it is possible to add more channels and increase T_{comm} while maintaining a constant E_b. Additionally, this can be done without increasing $A_{non\text{-}sensing}$, as no additional communication components are required. Furthermore, the increase in computation is negligible in these designs, allowing for efficient scaling.

Evaluation: We analyze the wireless SoCs 1–8 from Table 1 under both the naive and high-margin design hypotheses. Fig. 5 shows the projection of P_{SoC} relative to P_{budget} for each SoC. Each bar in the plots is labeled with the SoC number and is divided into $P_{sensing}$ and $P_{non\text{-}sensing}$. In the naive design, scaling the number of channels does not affect the ratio between P_{SoC} and P_{budget}.

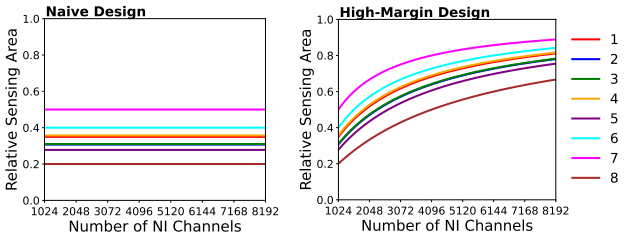


Fig. 6: Sensing area relative to total area versus channel count for the naive and high-margin designs. The high-margin design achieves superior volumetric efficiency for high channel counts, as the sensing area dominates overall SoC area.

This is because both P_{SoC} and A_{SoC} scale linearly with the number of channels, resulting in a consistent margin of unused P_{budget}. For the high-margin design, P_{SoC} eventually exceeds P_{budget} for all SoCs. This is because $P_{non\text{-}sensing}$ increases linearly at a faster rate than A_{SoC}, causing P_{SoC} to eventually surpass P_{budget}.

Fig. 6 illustrates the proportion of total chip area devoted to sensing, which serves as an indicator of volumetric efficiency. According to Equation (4), this proportion should be maximized when increasing the number of channels. In the naive design, the proportion of $A_{sensing}$ remains unchanged as n increases because $A_{non\text{-}sensing}$ grows at the same rate as $A_{sensing}$. This means that adding channels does not improve volumetric efficiency, which is a key limitation of the naive design. It significantly restricts neuron accessibility in large-scale NIs. From the perspective of volumetric efficiency, the high-margin design is preferable, as the normalized $A_{sensing}$ grows and eventually becomes the dominant area on the SoC.

These designs represent two opposing strategies: the naive design provides optimal power management on the SoC by minimizing power consumption per channel, but leads to an infeasible A_{SoC}. On the other hand, the high-margin design offers better volumetric efficiency, but results in an infeasible P_{SoC}. Neither strategy is fully viable on its own and a hybrid approach is needed to strike a more balanced trade-off between area and power.

5.2 Communication-Centric Architectures with Advanced Modulation

A common approach to support higher transmission data rates is to use modulation schemes that transmit multiple bits per symbol in each cycle, while maintaining stable antenna size and bandwidth. This is especially important for RF-based devices, as it helps comply with strict bandwidth regulations set by organizations such as the Federal Communications Commission (FCC) [36].

Quadrature Amplitude Modulation (QAM) is a widely used modulation scheme that enables the transmission of multiple bits per symbol [44, 92]. However, QAM requires a more complex transceiver design. In QAM, the energy per bit (E_b) varies depending on the number of bits transmitted per symbol. We assume that $A_{non\text{-}sensing}$ does not increase when the number of channels grows. This assumption is approximately valid because, when increasing the number of bits per symbol, only the modulation parameters change according to a known equation, while the antenna size remains constant.

To estimate the communication power, we solve the QAM equation and derive E_b for each number of bits per symbol. For example,

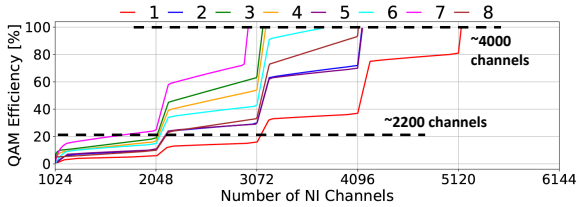


Fig. 7: QAM efficiency measures the power efficiency of the QAM implementation, with 15% representing the current achievable standard. At 100% efficiency, QAM supports an average of 4× more channels, while at 20% efficiency, a more realistic short-term target, it enables an average 2× increase.

in a design with $n = 1024$, we assume 1 bit per symbol for $n \leq 1024$. For $1024 < n \leq 2048$, communication will require 2 bits per symbol. For $2048 < n \leq 3072$, communication will require 3 bits per symbol, and so on. For each interval of 1024 channels, we solve the QAM equation, derive E_b, and calculate P_{comm} using the same expression as in Equation (9). Generally, adding a bit to the symbol is expected to significantly increase QAM power consumption.

For advanced modulation, similarly to the high-margin design, we assume that above $n = 1024$, the non-sensing area should not increase due to volumetric efficiency limitations. Instead, the existing $A_{non\text{-}sensing}$ for 1024 channels can be reused to implement QAM.

The solution of the QAM equation depends on parameters including bit error rate (BER), path loss, and additional margin that quantify the impact of biological tissue, such as the skull, and the distance between implanted and wearable SoCs on QAM transmission. However, this analysis remains optimistic, since real-world QAM implementations encounter additional challenges related to microarchitecture and antenna design.

Evaluation: We assume nominal QAM parameters of BER = 10^{-6}, path loss = 60 dB, and margin = 20 dB, consistent with assumptions about biological tissue reported in the context of BCIs [74, 93, 94].

In our analysis, we focus on *QAM efficiency*, a design parameter that quantifies the efficiency of the QAM implementation. Higher efficiency reduces P_{comm}, but due to design challenges, existing implementations for biomedical applications typically achieve only around 15% efficiency [74, 93, 94].

Fig. 7 shows the minimum QAM efficiency required to meet the power budget for a given number of channels. For each n, the reported QAM efficiency represents the minimum needed to keep $P_{SoC}(n)$ within $P_{budget}(n)$. Sharp increases in QAM efficiency indicate the addition of 1 bit per symbol in every QAM transmission cycle. At 20% QAM efficiency, a realistic short-term efficiency target, SoCs could double current channel counts on average. At the theoretical ideal of 100% efficiency, this increases to 4×. However, this is an optimistic analysis that overlooks the design challenges of QAM implementation and only partially accounts for RF impairments caused by biological tissue. It suggests that implanted SoCs cannot transmit full neural data at scale using advanced modulation.

While advanced modulation may meet the higher data rates needed for larger NIs, it requires overcoming significant design challenges. In the long term, even an ideal yet impractical QAM implementation would not support full neural data transmission, and realistic implementations are likely far more limited.

5.3 Computation-Centric Architectures with On-Implant DNNs

Given the challenge of transmitting raw neural data from a large number of channels while meeting throughput and power constraints, we turn to computation-centric architectures. By incorporating specialized computation on the implanted SoC, the amount of data that needs to be transmitted wirelessly is reduced. Recent works have adopted this approach to address the challenge of communication bottleneck [62, 111].

Since our focus is on future BCIs, we consider the most advanced applications to date, which rely on machine learning, and specifically DNNs, to achieve the highest classification accuracy in the BCI domain [8, 11, 43, 75, 76, 108]. DNN models are computationally intensive and scale super-linearly with input size [50, 60, 127]. As NIs scale, the resulting increase in input size may require larger DNNs to accommodate the additional data volume [113, 119]. Current BCI applications use relatively simple types of neural networks as multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), and even spiking neural networks (SNNs) [11, 54, 90, 103].

Although traditional algorithms like the Kalman filter remain important for BCI [9, 23, 53, 114] and have been explored in implanted SoCs [110], current trends suggest they will play a diminishing role in future BCI applications [43, 53]. We argue that integrating modern DNNs into implanted SoCs is both crucial and underexplored.

Methodology: Research on DNN accelerators beyond BCIs has explored both spatial and temporal computation flows [10, 41, 48, 49], often targeting diverse applications. These designs focus on optimizing the data flow between the accelerator, CPU, and main memory, which is usually the main bottleneck.

In contrast, implantable BCIs are typically designed for a single application, such as assisting with specific disability, and follow a computation-centric, one-way pipeline data flow [54, 96]. This allows for a non-Von Neumann architecture that excludes components like CPUs and centralized memory, which are unnecessary for the minimal, application-specific BCI data path [15, 39].

DNNs rely heavily on matrix multiplications and convolutions, which both use multiply-and-accumulate (MAC) operations extensively [19, 102]. Therefore, we focus on the MAC to establish a first-order lower bound on the power requirements of a DNN accelerator for an implant. We assume an architecture where weights remain stationary and are physically collocated with each MAC unit at design time, treating each MAC as a standalone processing element (PE) in a non-Von Neumann manner [68].

To reason about the number of MACs and their associated resources on an implant, we define the following variables:

- MAC_{op}: A sequence of steps, where each step involves multiplying two values and adding the result to a previous value.
- $\#MAC_{op}$: The number of independent MAC_{op} in a DNN layer.
- MAC_{seq}: The number of accumulation steps in a MAC_{op}.
- MAC_{hw}: The cost of adding one MAC unit capable of executing one MAC_{op}, including the MAC unit and any resources needed for its encapsulation as a PE.

Fig. 8 provides a simple example. At the top, a matrix-matrix multiplication between $A_{4\times3}$ and $B_{3\times4}$ involves $\#MAC_{op} = 4$ operations and a sequence length of $MAC_{seq} = 3$. At the bottom, a convolutional layer with two input channels, one output channel,

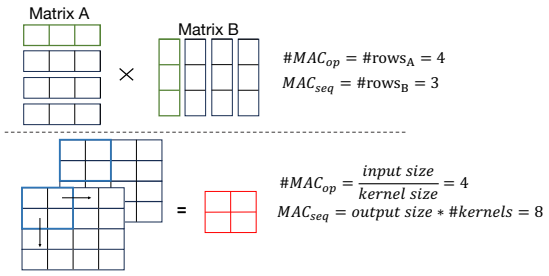


Fig. 8: $\#MAC_{op}$ and MAC_{seq} example.

a kernel size of 4, and an output size of 4 involves $\#MAC_{op} = 4$ operations and a sequence length of $MAC_{seq} = 8$.

By definition, all MAC_{op} operations within an individual layer of the DNN are independent and can be computed in parallel. Furthermore, the value of MAC_{seq} is consistent for every MAC_{op} in a given layer. For simplicity, we assume that a complete MAC_{op} is executed by the same MAC_{hw}. Additionally, we assume that a single MAC_{hw} can time multiplex MAC_{op} operations within timing constraints. This approach allows us to minimize the number of MAC_{hw} units and achieve the lower bound on resource usage.

To demonstrate our approach, we design a DNN accelerator that implements a DNN layer. It includes a finite state machine (FSM) to control data movement and time multiplexing of PEs, and each PE contains a MAC unit, a ReLU, a small FSM, and a read-only memory (ROM) storing its assigned weights. The accelerator can be configured at design time to support various layer sizes for MLPs or CNNs in BCI applications [11]. We synthesize the accelerator for an 8-bit datatype across a range of $\#MAC_{op}$, MAC_{seq}, and MAC_{hw} values, using parameters based on SoC 1 in Table 1, targeting 130nm TSMC technology at 100 MHz. Power estimates are obtained with Cadence Genus 20.1 and Joules.

Fig. 9 shows the accelerator architecture (DNN layer), the PE design, a table summarizing 12 design configurations, and their power analysis. In smaller designs (1-5), where only $\#MAC_{op}$ increases, the relative PE power stays low at around 25%. When MAC_{hw} increases to match $\#MAC_{op}$ (designs 6-9), total power rises, with PE power reaching roughly 80% of the total. Further scaling of $\#MAC_{op}$, MAC_{hw}, and MAC_{seq} (designs 10-12) raises relative PE power from about 80% to 96%, showing that the PE is the primary driver of power consumption and supporting our lower bound estimates that scale proportionally to MAC_{hw}.

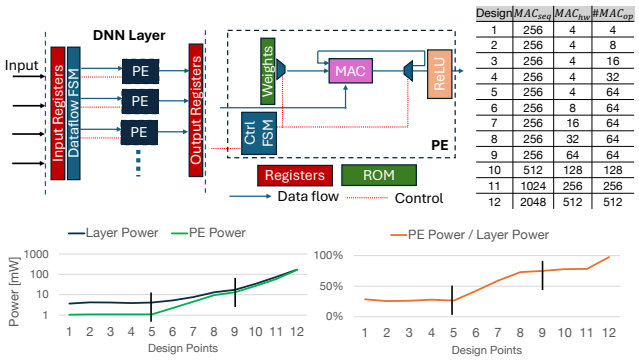
Optimization: The maximum allowable time to execute the DNN under real-time constraints is determined by the NI sampling frequency ($f = \frac{1}{t}$). To find a hardware-based DNN capable of executing N layers within t, we must first determine $\#MAC_{op}$ and MAC_{seq} per DNN layer. We define a function f_{MAC} for this purpose:

$$[MAC^i_{seq}], [\#MAC^i_{op}] = f_{MAC}(n, DNN),\ i \in [1, N] \quad (10)$$

where MAC^i_{op} and MAC^i_{seq} are the MAC_{op} and MAC_{seq} of the i-th layer, respectively. We then define the total runtime of the i-th layer, t_i , in terms of these variables:

$$t_i = (MAC^i_{seq} \cdot t_{MAC}) \cdot \left\lceil \frac{\#MAC^i_{op}}{\#MAC_{hw}} \right\rceil, \quad \sum_{i=1}^{N} t_i \le t \quad (11)$$

where t_{MAC} is the time of executing one step with one MAC_{hw}. The final constraints require that the number of MAC_{hw} is non-zero



Design	MAC_{seq}	MAC_{hw}	$\#MAC_{op}$
1	256	4	4
2	256	4	8
3	256	4	16
4	256	4	32
5	256	4	64
6	256	8	64
7	256	16	64
8	256	32	64
9	256	64	64
10	512	128	128
11	1024	256	256
12	2048	512	512

Fig. 9: A DNN accelerator built from PEs that include a MAC unit, ReLU, and ROM. Synthesizing twelve design variants by varying MAC_{seq}, MAC_{hw}, and $\#MAC_{op}$ shows that total power consumption tracks increases in MAC_{hw} closely, with PEs accounting for over 80% of total power in large configurations.

and it does not exceed the maximum $\#MAC_{op}$ in any single layer:

$$\#MAC_{hw} > 0, \quad \#MAC_{hw} \le max(\#MAC^i_{op}) \quad (12)$$

This results in a lower bound on P_{comp} of:

$$P_{comp} = \#MAC_{hw} \cdot P_{MAC} \quad (13)$$

where P_{MAC} is the power consumption of a single MAC_{hw}.

If we permit pipelining of the DNN layers, Equation (11) and Equation (12) would become:

$$t_i = (MAC^i_{seq} \cdot t_{MAC}) \cdot \left\lceil \frac{\#MAC^i_{op}}{\#MAC^i_{hw}} \right\rceil, \quad max(t_i) \le t \quad (14)$$

$$\#MAC_{hw} > 0, \quad \#MAC_{hw} = \sum_{i=1}^{N} \#MAC^i_{hw} \le \sum_{i=1}^{N} \#MAC^i_{op} \quad (15)$$

where $\#MAC^i_{hw}$ is the number of MAC units for the i-th layer in the DNN and the new objective is to minimize $\#MAC^i_{hw}$ for each layer, while $\#MAC_{hw}$ must not exceed the total $\#MAC_{op}$ in the DNN.

After DNN processing, the amount of data to transmit is significantly reduced, which should decrease P_{comm}, helping to offset the increase in P_{comp}, particularly when the DNN is scaled to handle a very large number of channels. Our goal is to understand under which conditions investing in computation is more effective than investing in higher-throughput communication.

Scaling Factor: In most classification DNNs, the output size remains unaffected by the input size, as the output is typically a vector of probabilities, one for each label in a fixed set. While changes in input size may alter the probabilities, they do not affect the number of labels. However, changes in input size do influence the structure of the intermediate layers [50, 60, 113, 127]. As the number of channels, n, increases, we scale each layer size (weights) and the network depth (number of layers) according to a parameter we name as $\boldsymbol{\alpha} = \frac{n}{\text{original input size}}$.

Evaluation: To evaluate the scaling of DNN integration, we synthesize a MAC unit to obtain its latency (t_{MAC}) and power consumption (P_{MAC}), which are used directly in our equations. We exclude overheads from local memory, routing, and data movement, as these depend on specific accelerator designs. Instead, our

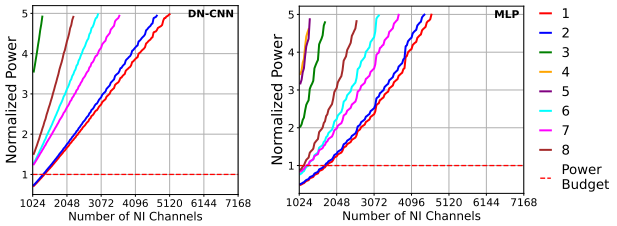


Fig. 10: Power consumption normalized to the power budget of implanted SoCs containing DNNs.

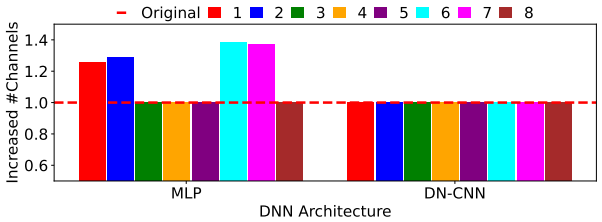


Fig. 11: The increase in allowable channel count after partitioning DNNs between the implant and the wearable.

approach provides a fundamental, architecture-independent *lower bound* on DNN power consumption based on the minimum number of MAC units required for correct execution, which we show earlier in the paper to be the main driver of power consumption in large designs. We use this conservative lower bound to determine whether a given DNN can potentially meet the power and performance constraints of an implant, as defined in Section 4. If it can, second-order microarchitectural factors may be incorporated using the margin between the lower bound and the total power budget. This method balances broad applicability across architectures with feasibility within implanted BCI SoC constraints.

Results: We synthesize a MAC unit in 45nm technology using the NanGate Open Cell Library [116] with a target clock frequency of 100 MHz, resulting in $t_{MAC} = 2$ ns and $P_{MAC} = 0.05$ mW.

We experiment with a DenseNet Convolutional Neural Network (DN-CNN) and a Multi-Layer Perceptron (MLP), both trained for speech synthesis using ECoG neural data [11]. These DNNs are representative of modern BCI applications [90, 103]. We begin with the original implementation of these DNNs designed for 128 channels sampled at 2 kHz ($n = 128$, $f = 2$ kHz). The output of both networks consists of 40 labels, each corresponding to a speech frequency that can be used to generate audio [11]. For each DNN, we report the best result between a pipelined and a non-pipelined design.

Fig. 10 shows the lower-bound for P_{SoC} relative to P_{budget} for SoCs 1–8, each scaled for different numbers of channels. We observe that, even at the current standard of 1024 channels, some designs do not have enough power budget to integrate the lower bound of DNN power. Only SoCs 1 and 2 can integrate the DN-CNN for 1024 channels, while SoCs 4 and 5 exceed the power budget by a factor of 5× and fall outside the bounds of the plot. For the MLP model, only SoCs 3–5 cannot integrate it at 1024 channels.

For those SoCs that can accommodate the DNNs, the average maximum channel count appears at $n \approx 1800$ for MLP and $n \approx 1400$ for DN-CNN. In comparison, when resources were invested in advanced modulation, $n \approx 1800$ is achievable with only 13% QAM efficiency (Fig. 7). This suggests that, when scaling to only twice the current standard number of channels (2048), optimizing a communication-centric architecture for the implanted SoC may prove more effective than adopting a computation-centric approach.

Even a first-order lower bound power estimate of modern DNNs used in cutting-edge BCI applications, considering only the MAC units, cannot scale to twice the current standard channel count without exceeding the power budget. This raises serious concerns about integrating full DNN accelerators into implanted SoCs with current state-of-the-art design methods.

6 Computation-Centric Architectures with Combined Optimizations

Modern DNNs, in their current form, are unsuitable for integration into implanted SoCs with only twice the current standard channel count. Therefore, leveraging DNNs as part of computation-centric architectures requires alternative solutions. In this section, we explore potential optimization strategies that, when combined, could bring the overall SoC within the power budget.

6.1 DNN Partitioning

DNN architectures, particularly CNNs and MLPs, rely on a unidirectional data flow that propagates through a series of layers [5, 6, 55]. Given that the communication between the implanted SoC and the wearable SoC is wireless, short-range, and capable of supporting high data rates [42, 139], a natural optimization would be to partition the DNN, placing only the initial layers on the implant and the remaining layers on the wearable. This approach would reduce P_{comp} but increase the required transmission rate T_{comm}, because the output data from intermediate DNN layers is typically larger than that of the final layer. Consequently, P_{comm} would increase.

For each implanted SoC, we partition the DNN at the earliest layer for which the required T_{comm} does not exceed the transmission data rate of a 1024-channel communication-centric design. We call this optimization ***layer reduction***.

For each SoC, Fig. 11 shows the increase in feasible channel count enabled by the layer reduction optimization, relative to the maximum supported by the full DNN. The best result is a 40% increase in channel count for SoC 6 with the MLP model. On average, layer reduction provides an improvement of ~20%, enabling the short-term goal of supporting 2048 channels. In contrast, the DN-CNN shows no benefit from layer reduction across the SoC designs. These pessimistic results mainly arise from the increased transmission data rate. While offloading some DNN layers reduces P_{comp}, it increases P_{comm} because the final on-chip DNN layer generates a larger output and a higher T_{comm}.

These results suggest that partitioning DNNs can help integrate more NI channels in the short term. However, benefits vary by computation type and decrease as the system scales, indicating that modern BCIs need further optimization in both application and architecture.

6.2 Complementary Optimizations

Based on current BCI trends, additional optimization strategies can be considered:

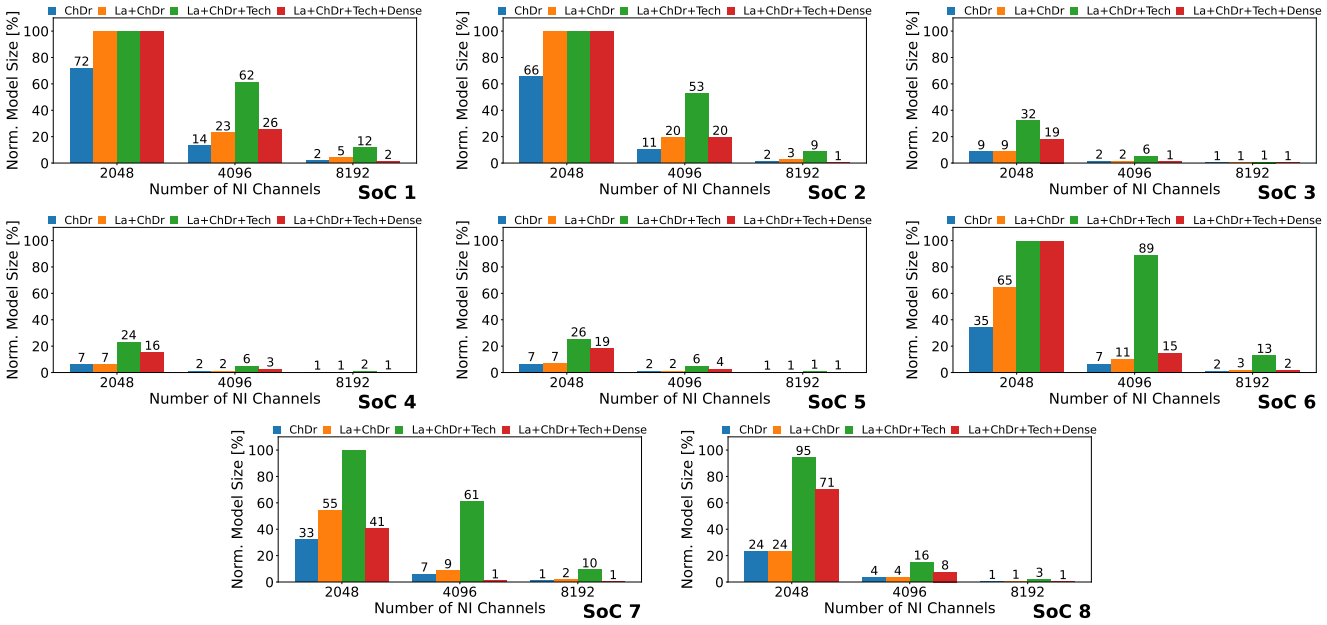


Fig. 12: Feasible MLP model sizes on implanted SoCs 1-8 as a function of channel count, after applying optimizations.

- In some applications, recording data simultaneously from all accessible neurons has been found to be redundant due to temporal and spatial dependencies between the activities of different neurons [11, 113, 119]. As a result, computational methods such as spike sorting [71] are often used to reduce the amount of neural data [1, 56, 112, 136]. These methods filter out data from inactive neurons, effectively reducing the computational load. Such methods are generally more suitable for implant-based BCI systems than standard compression techniques, as they eliminate the need for additional memory and reduce computational steps. We refer to this optimization as ***channel dropout***.
- The computation on an implant could potentially be implemented in a more advanced technology node [13, 24], which would reduce P_{comp} and allow for larger DNNs on the implanted SoC. However, the NI and communication components are built with analog circuits, which do not benefit from more advanced nodes [63]. We refer to this optimization as ***technology scaling***.
- To improve the resolution of neural data, NI channels are being integrated more densely [64]. This approach reduces $A_{sensing}$ and enhances chip flexibility [115]. However, it is important to note that reducing the total chip area also decreases the power budget [128]. We refer to this optimization as ***channel density***.

Evaluation: To evaluate the proposed optimizations, we experiment with the MLP model. For each SoC and channel count, we estimate the maximum computational workload that fits within the power budget by applying channel dropout. This involves setting the value of $\boldsymbol{\alpha}$ from Section 5.3 using a reduced number of active channels, $n' < n$, which mitigates NI scaling and reduces the MLP model size. This approach helps determine how much computation a given DNN architecture can realistically support on the SoC. While it does not directly indicate accuracy or overall application performance, we prioritize the largest feasible value of n'.

We then apply the remaining optimizations sequentially and reevaluate the minimum channel dropout needed to fit the DNN. Specifically, we apply the optimizations in four steps:

- **Channel Dropout Only:** `ChDr`. We determine the minimum dropout value while ensuring that P_{SoC} remains within P_{budget}.
- **Adding Layer Reduction:** `La+ChDr`. We split the DNN to reduce its computational complexity.
- **Adding Technology Scaling:** `La+ChDr+Tech`. We scale the technology node to 12nm (from 45nm) by resynthesizing the MAC, resulting with t_{MAC} = 1 ns and P_{MAC} = 0.026 mW.
- **Adding Channel Density:** `La+ChDr+Tech+Dense`. We reduce the sensing area per channel by a factor of two.

Fig. 12 shows the resulting MLP model size after applying each optimization step, normalized to the original model size without any optimizations. The results are based on channel counts of n = 2048, 4096, and 8192. From left to right, the bars represent the sequential addition of each optimization step for each value of n.

For 2048 channels, `ChDr` reduces the DNN model size to an average of 32% to meet the system constraints. For 4096 channels, the model size is reduced to an average of 6%, and for 8192 channels, the reduction reaches 2% of the original model size.

In most cases, applying `La` enables the use of larger DNNs in the BCI system, increasing model size by an average of 30% for 2048 channels, 28% for 4096 channels, and 44% for 8192 channels.

In all cases, applying `Tech` reduces P_{SoC}, enabling a significant increase in the DNN model size. For 2048 channels, the model size increases to an average of 72% of the original, or 2.3× compared to applying just `ChDr+La`. For 4096 channels, it reaches 37%, or 6×. For 8192 channels, the average model size remains at 6%, or 3×.

Applying `Dense` reduces A_{SoC}, which lowers the overall P_{budget}. For 2048 channels, the model size drops to an average of 58% of the

original. For 4096 channels, it falls to 7%. For 8192 channels, only an average of 2% of the original model size fits within the BCI system.

With the exception of `Dense`, which lowers P_{budget}, optimizations generally increase the allowable computational workload on the implant. However, some reduction in workload size is typically required and maintaining accuracy and overall performance depends on factors such as model training and algorithm design.

Modern BCI computation should be reevaluated to meet system constraints, while system-level optimizations must adapt to specific applications. Aligning application demands with system limitations is essential for developing safe, implantable, and scalable BCI systems.

7 Related Work

State-of-the-art BCIs: Implantable BCI systems focus on integrating large-scale neural interfaces with thousands of channels [12, 42, 51, 59, 85, 98, 112]. The Utah array [12] is the most widely used implantable BCI platform in neuroengineering research [17, 18, 88]. It relies on penetrating electrodes and a wired connection. Neuropixels [112, 125] is another popular device for recording brain data. It relies on a wired connection to external processing units [91]. In contrast to wireless BCI systems, wired systems have distinct power constraints. Notably, the Neuralink design [85, 136] incorporates both wireless communication and on-chip spike detection. It features an implanted SoC that replaces part of the skull and interfaces with the brain using electrode-carrying wires.

Computation-capable BCIs: BCI systems have been developed with the capability to perform real-time computations [28–30, 62, 111, 129]. Some of these systems implement such computations on the implant for small-scale NIs with approximately 100 channels [62], while others scale by employing multiple implanted SoCs [111]. However, recent findings suggest that these designs may not scale effectively for individual SoCs when dealing with large-scale neural interfaces [118]. Some systems assume that the implanted SoC can transmit all neural data in real time, with the computation being offloaded to the wearable SoC [29, 30]. This solution is viable if we can support higher data rates through wireless communication or reduce the data rate using hardware-efficient methods to detect patterns in neural activity [1, 56].

For non-invasive BCIs, an alternative approach is to decompose the computation into smaller operations, which can then be executed closer to the data source [129].

Research has focused on defining low-power thresholds for implantable, computation-capable BCIs [54, 96, 97, 109]. Silowowski *et al.* find that additional neural data can improve DNN accuracy [109]. However, they suggest using techniques to evaluate the relevance of added neural data to minimize the size of the DNN model. Saad *et al.* analyze BCI circuits for motor decoding and review low-power optimizations [96, 97]. They highlight the challenge of analytically estimating computational cost due to the diversity of BCI algorithms and implementations, and observe that power consumption scales with the number of channels in MEA-based BCIs. Our analysis complements their decoder-focused perspective by providing a holistic system-level view across sensing, computation, communication, and channel scaling, helping to guide future BCI system designs.

Closed-loop BCIs: Hueber *et al.* provide a benchmark for estimating the overhead of hardware implementations in closed-loop BCI computations [54]. Their analysis examines various computational methods, including linear techniques, artificial neural networks (ANNs), and spiking neural networks (SNNs). To estimate the power consumption for each method, they calculate the number of MAC operations and memory accesses. Their experiments involve a neural interface with up to 192 channels. They conclude that hardware implementations for BCI computations must carefully consider trade-offs between complexity, latency, and power consumption. In addition, they highlight the advantages of using SNNs in closed-loop BCIs, as they offer improved power efficiency. In our study, we focus on open-loop BCI applications, which, unlike closed-loop applications, rely heavily on wireless communication to interact with external devices. Despite this, the approach to modeling MAC operations is similar for both types of applications. In the future, we plan to extend this work to accommodate closed-loop BCIs and explore additional computational models, such as SNNs.

8 Future Considerations

MINDFUL builds upon established standards and emerging trends in the BCI field that are essential for advancing future development. Several challenges and opportunities deserve continued attention from the computer architecture community.

As BCI systems become more standardized, NI scaling is expected to accelerate beyond the current trend of doubling the channel count approximately every seven years [85, 98], further increasing the need to move computation into the implant. Embedded computation, however, introduces challenges beyond integration. For example, DNNs in BCI applications often require periodic retraining and recalibration to accommodate ongoing changes in biological tissue [23, 70, 109]. As designs grow, secondary power effects such as routing overhead and thermal constraints will become more significant and must be addressed [35, 86, 101]. Wireless power transfer (WPT) is increasingly used to power implants, but it raises questions about power efficiency and heat generation [42, 139]. The form factor of NIs must also evolve; future interfaces may need to be soft, flexible, and penetrating to access more neurons while maintaining biocompatibility and safety [105]. Analog components remain a key scaling limitation. Unlike digital logic, analog elements do not scale efficiently below 65nm technology and can be expected to constrain designs [63]. Finally, real-time performance must be evaluated at the application level rather than only by data rate or sampling frequency [22, 29, 129]. BCI systems must provide fast and reliable responses to brain activity, which calls for tighter co-design between system architecture and application development.

9 Conclusion

MINDFUL is an analytical framework to support research and development in computer architectures for implantable BCI systems, a field undergoing rapid transformation as neural interfaces, communication techniques, and application demands continue to evolve. By establishing a clearer understanding of the fundamental constraints and trade-offs in BCI system design, we hope this work will help shape future research directions and foster the development of safe, scalable, and application-aware BCI architectures.

References

[1] Ameer MS Abdelhadi, Eugene Sha, Ciaran Bannon, Hendrik Steenland, and Andreas Moshovos. 2021. Noema: Hardware-Efficient Template Matching for Neural Population Pattern Detection. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*. 522–534.

[2] Reza Abiri, Soheil Borhani, Eric W Sellers, Yang Jiang, and Xiaopeng Zhao. 2019. A Comprehensive Review of EEG-based Brain–Bomputer Interface Paradigms. *Journal of Neural Engineering* 16, 1 (2019), 011001.

[3] Rahul Ajmeria, Mayukh Mondal, Reya Banerjee, Tamesh Halder, Pallav Kumar Deb, Debasish Mishra, Pravanjan Nayak, Sudip Misra, Surjya Kanta Pal, and Debashish Chakravarty. 2022. A Critical Survey of EEG-based BCI Systems for Applications in Industrial Internet of Things. *IEEE Communications Surveys & Tutorials* (2022).

[4] Abeer Al-Nafjan, Manar Hosny, Yousef Al-Ohali, and Areej Al-Wabil. 2017. Review and Classification of Emotion Recognition based on EEG Brain-Computer Interface System Research: A Systematic Review. *Applied Sciences* 7, 12 (2017), 1239.

[5] Cesare Alippi, Simone Disabato, and Manuel Roveri. 2018. Moving Convolutional Neural Networks to Embedded Systems: The Alexnet and VGG-16 Case. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 212–223.

[6] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. 2018. The History Began from Alexnet: A Comprehensive Survey on Deep Learning Approaches. *arXiv preprint arXiv:1803.01164* (2018).

[7] Alon Amid, David Biancolin, Abraham Gonzalez, Daniel Grubb, Sagar Karandikar, Harrison Liew, Albert Magyar, Howard Mao, Albert Ou, Nathan Pemberton, et al. 2020. Chipyard: Integrated Design, Simulation, and Implementation Framework for Custom SoCs. *IEEE Micro* 40, 4 (2020), 10–21.

[8] Miguel Angrick, Christian Herff, Emily Mugler, Matthew C Tate, Marc W Slutzky, Dean J Krusienski, and Tanja Schultz. 2019. Speech Synthesis from ECoG Using Densely Connected 3D Convolutional Neural Networks. *Journal of Neural Engineering* 16, 3 (2019), 036019.

[9] Matin Asgharpour, Reza Foodeh, and Mohammad Reza Daliri. 2021. Regularized Kalman Filter for Brain-Computer Interfaces Using Local Field Potential Signals. *Journal of Neuroscience Methods* 350 (2021), 109022.

[10] Colby Banbury, Vijay Janapa Reddi, Peter Torelli, Jeremy Holleman, Nat Jeffries, Csaba Kiraly, Pietro Montino, David Kanter, Sebastian Ahmed, Danilo Pau, et al. 2021. MLPerf Tiny Benchmark. *arXiv preprint arXiv:2106.07597* (2021).

[11] Julia Berezutskaya, Zachary V Freudenburg, Mariska J Vansteensel, Erik J Aarnoutse, Nick F Ramsey, and Marcel AJ van Gerven. 2023. Direct Speech Reconstruction from Sensorimotor Brain Activity with Optimized Deep Learning Models. *Journal of Neural Engineering* 20, 5 (2023), 056010.

[12] Blackrock Neurotech. 2025. Utah Array. https://blackrockneurotech.com/products/utah-array/ Accessed: 2025-04-08.

[13] Shekhar Borkar. 1999. Design Challenges of Technology Scaling. *IEEE Micro* 19, 4 (1999), 23–29.

[14] Santiago A Cadena, Konstantin F Willeke, Kelli Restivo, George Denfield, Fabian H Sinz, Matthias Bethge, Andreas S Tolias, and Alexander S Ecker. 2024. Diverse Task-Driven Modeling of Macaque V4 Reveals Functional Secialization Towards Semantic Tasks. *PLOS Computational Biology* 20, 5 (2024), e1012056.

[15] Ermao Cai, Da-Cheng Juan, Dimitrios Stamoulis, and Diana Marculescu. 2017. NeuralPower: Predict and Deploy Energy-Efficient Convolutional Neural Networks. In *Asian Conference on Machine Learning*. PMLR, 622–637.

[16] Ji-Hyoung Cha, Jee-Ho Park, Yongjae Park, Hyogeun Shin, Kyeong Seob Hwang, Il-Joo Cho, and Seong-Jin Kim. 2022. A Reconfigurable Sub-Array Multiplexing Microelectrode Array System with 24,320 Electrodes and 380 Readout Channels for Investigating Neural Communication. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 65. IEEE, 342–344.

[17] Xing Chen, Aitor Morales-Gregorio, Julia Sprenger, Alexander Kleinjohann, Shashwat Sridhar, Sacha J Van Albada, Sonja Grün, and Pieter R Roelfsema. 2022. 1024-Channel Electrophysiological Recordings in Macaque V1 and V4 During Rresting State. *Scientific Data* 9, 1 (2022), 77.

[18] Xing Chen, Feng Wang, Roxana Kooijmans, Peter Christiaan Klink, Christian Boehler, Maria Asplund, and Pieter Roelf Roelfsema. 2023. Chronic Stability of a Neuroprosthesis Comprising Multiple Adjacent Utah Arrays in Monkeys. *Journal of Neural Engineering* 20, 3 (2023), 036039.

[19] Harsh Chhajed, Gopal Raut, Narendra Dhakad, Sudheer Vishwakarma, and Santosh Kumar Vishvakarma. 2022. Bitmac: Bit-Serial Computation-based Efficient Multiply-Accumulate Unit for DNN Accelerator. *Circuits, Systems, and Signal Processing* (2022), 1–16.

[20] Jaebin Choi, Adriaan J Taal, Eric H Pollmann, Changhyuk Lee, Kukjoo Kim, Laurent C Moreaux, Michael L Roukes, and Kenneth L Shepard. 2019. A 512-Pixel, 51-KHz-Frame-Rate, Dual-Shank, Lens-Less, Filter-Less Single-Photon Avalanche Diode CMOS Neural Imaging Probe. *IEEE Journal of Solid-State Circuits* 54, 11 (2019), 2957–2968.

[21] Judicael Clair, Guy Eichler, and Luca P Carloni. 2023. SpikeHard: Efficiency-Driven Neuromorphic Hardware for Heterogeneous Systems-on-Chip. *ACM Transactions on Embedded Computing Systems* 22, 5s (2023), 1–22.

[22] John P Cunningham, Paul Nuyujukian, Vikash Gilja, Cindy A Chestek, Stephen I Ryu, and Krishna V Shenoy. 2011. A Closed-Loop Human Simulator for Investigating the Role of Feedback Control in Brain-Machine Interfaces. *Journal of Neurophysiology* 105, 4 (2011), 1932–1949.

[23] Alan D Degenhart, William E Bishop, Emily R Oby, Elizabeth C Tyler-Kabara, Steven M Chase, Aaron P Batista, and Byron M Yu. 2020. Stabilization of a Brain-Computer Interface via the Alignment of Low-Dimensional Spaces of Neural Activity. *Nature Biomedical Engineering* 4, 7 (2020), 672–685.

[24] Robert H Dennard, Jin Cai, and Arvind Kumar. 2018. A Perspective on Today's Scaling Challenges and Possible Future Directions. In *Handbook of Thin Film Deposition*. Elsevier, 3–18.

[25] Zhuokun Ding, Paul G Fahey, Stelios Papadopoulos, Eric Y Wang, Brendan Celii, Christos Papadopoulos, Alexander B Kunin, Andersen Chang, Jiakun Fu, Zhiwei Ding, et al. 2023. Functional Connectomics Reveals General Wiring Rule in Mouse Visual Cortex. *bioRxiv* (2023).

[26] Ahmed Yasir Dogan, Jeremy Constantin, Martino Ruggiero, Andreas Burg, and David Atienza. 2012. Multi-Core Architecture Design for Ultra-Low-Power Wearable Health Monitoring Systems. In *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 988–993.

[27] Eva L Dyer, Mohammad Gheshlaghi Azar, Matthew G Perich, Hugo L Fernandes, Stephanie Naufel, Lee E Miller, and Konrad P Körding. 2017. A Cryptography-based Approach for Movement Decoding. *Nature Biomedical Engineering* 1, 12 (2017), 967–976.

[28] Guy Eichler. 2025. *System-Level Design in the Era of Brain-Computer Interfaces*. Columbia University.

[29] Guy Eichler, Luca Piccolboni, Davide Giri, and Luca P Carloni. 2021. MasterMind: Many-Accelerator SoC Architecture for Real-Time Brain-Computer Interfaces. In *2021 IEEE 39th International Conference on Computer Design (ICCD)*. IEEE, 101–108.

[30] Guy Eichler, Biruk Seyoum, Kuan Lin Chiu, and Luca P Carloni. 2023. MindCrypt: The Brain as a Random Number Generator for SoC-Based Brain-Computer Interfaces. In *2023 IEEE 41th International Conference on Computer Design (ICCD)*. IEEE.

[31] Guy Eichler, Joseph Zuckerman, and Luca P Carloni. 2025. An Energy-Efficient Kalman Filter Architecture with Tunable Accuracy for Brain-Computer Interfaces. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*. 1863–1875.

[32] Guy Eichler, Joseph Zuckerman, and Luca P Carloni. 2025. KalmMind: A Configurable Kalman Filter Design Framework for Embedded Brain-Computer Interfaces. In *2025 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 1–2.

[33] Maged M Elwassif, Qingjun Kong, Maribel Vazquez, and Marom Bikson. 2006. Bio-Heat Transfer Model of Deep Brain Stimulation-Induced Temperature Changes. *Journal of Neural Engineering* 3, 4 (2006), 306.

[34] Thelma Estrin. 1965. On-line Electroencephalographic Digital Computing System. *Electroencephalography and Clinical Neurophysiology* 19, 5 (1965), 524–526.

[35] Nir Even-Chen, Dante G Muratore, Sergey D Stavisky, Leigh R Hochberg, Jaimie M Henderson, Boris Murmann, and Krishna V Shenoy. 2020. Power-Saving Design opportunities for Wireless Intracortical Brain–Computer Interfaces. *Nature Biomedical Engineering* 4, 10 (2020), 984–996.

[36] Federal Communications Commission. 2025. Wireless Telecommunications. https://www.fcc.gov/wireless-telecommunications Accessed: 2025-04-11.

[37] Eduardo Fernández, Arantxa Alfaro, Cristina Soto-Sánchez, Pablo Gonzalez-Lopez, Antonio M Lozano, Sebastian Peña, Maria Dolores Grima, Alfonso Rodil, Bernardeta Gómez, Xing Chen, et al. 2021. Visual Percepts Evoked with an Intracortical 96-Channel Microelectrode Array Inserted in Human Occipital Cortex. *The Journal of Clinical Investigation* 131, 23 (2021).

[38] Abdoulaye Gamatié, Guillaume Devic, Gilles Sassatelli, Stefano Bernabovi, Philippe Naudin, and Michael Chapman. 2019. Towards Energy-Efficient Heterogeneous Multicore Architectures for Edge Computing. *IEEE Access* 7 (2019), 49474–49491.

[39] Antara Ganguly, Rajeev Muralidhar, and Virendra Singh. 2019. Towards Energy Efficient Non-Von Neumann Architectures for Deep Learning. In *20th international symposium on quality electronic design (ISQED)*. IEEE, 335–342.

[40] Angelo Garofalo, Yvan Tortorella, Matteo Perotti, Luca Valente, Alessandro Nadalini, Luca Benini, Davide Rossi, and Francesco Conti. 2022. DARKSIDE: A Heterogeneous RISC-V Compute Cluster for Extreme-Edge On-Chip DNN Inference and Training. *IEEE Open Journal of the Solid-State Circuits Society* 2 (2022), 231–243.

[41] Hasan Genc, Seah Kim, Alon Amid, Ameer Haj-Ali, Vighnesh Iyer, Pranav Prakash, Jerry Zhao, Daniel Grubb, Harrison Liew, Howard Mao, et al. 2021. Gemmini: Enabling Systematic Deep-Learning Architecture Evaluation via Full-Stack Integration. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 769–774.

[42] Yatin Gilhotra, Henry Overhauser, Heyu Yin, Eric Pollmann, Guy Eichler, Andrew Cheng, Jung Taesung, Nanyu Zeng, Luca P. Carloni, and Kenneth Shepard. 2024. A Wireless Subdural Optical Cortical Interface Device with 768 Co-Packaged Micro-LEDs for Fluorescence Imaging and Optogenetic Stimulation. In *IEEE Custom Integrated Circuits Conference (CICC)*. IEEE.

[43] Joshua I Glaser, Ari S Benjamin, Raeed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. 2020. Machine Learning for Neural Decoding. *Eneuro* 7, 4 (2020).

[44] Andrea Goldsmith. 2005. *Wireless Communications*. Cambridge University Press.

[45] LS Goldstein, MW Dewhirst, M Repacholi, and L Kheifets. 2003. Summary, Conclusions and Recommendations: Adverse Temperature Levels in the Human Body. *International Journal of Hyperthermia* 19, 3 (2003), 373–384.

[46] Bernhard Graimann, G Townsend, JE Huggins, A Schlögl, SP Levine, and G Pfurtscheller. 2005. A Comparison between Using ECoG and EEG for Direct Brain Communication. *Proceedings of the EMBEC05* (2005).

[47] Shoubhik Gupta, William Taube Navaraj, Leandro Lorenzelli, and Ravinder Dahiya. 2018. Ultra-Thin Chips for High-Performance Flexible Electronics. *npj Flexible Electronics* 2, 1 (2018), 8.

[48] Cong Hao, Jordan Dotzel, Jinjun Xiong, Luca Benini, Zhiru Zhang, and Deming Chen. 2021. Enabling Design Methodologies and Future Trends for Edge AI: Specialization and Codesign. *IEEE Design & Test* 38, 4 (2021), 7–26.

[49] Jude Haris, Perry Gibson, José Cano, Nicolas Bohm Agostini, and David Kaeli. 2023. SECDA-TFLite: A Toolkit for Efficient Development of FPGA-based DNN Accelerators for Edge Inference. *J. Parallel and Distrib. Comput.* 173 (2023), 140–151.

[50] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. 2017. Deep Learning Scaling is Predictable, Empirically. *arXiv preprint arXiv:1712.00409* (2017).

[51] Mark Hettick, Elton Ho, Adam J Poole, Manuel Monge, Demetrios Papageorgiou, Kazutaka Takahashi, Morgan LaMarca, Daniel Trietsch, Kyle Reed, Mark Murphy, et al. 2022. The Layer 7 Cortical Interface: A Scalable and Minimally Invasive Brain–Computer Interface Platform. *bioRxiv* (2022), 2022–01.

[52] Chee Keong Ho, Jia Hao Cheong, and Yuan Gao. 2015. A High Datarate Wideband OOK Transmitter for Wireless Neural Signal Recording. In *2015 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*. IEEE, 178–181.

[53] Tommy Hosman, Marco Vilela, Daniel Milstein, Jessica N Kelemen, David M Brandman, Leigh R Hochberg, and John D Simeral. 2019. BCI Decoder Performance Comparison of an LSTM Recurrent Neural Network and a Kalman Filter in Retrospective Simulation. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 1066–1071.

[54] Paul Hueber, Guangzhi Tang, Manolis Sifalakis, Hua-Peng Liaw, Aurora Micheli, Nergis Tömen, and Yao-Hong Liu. 2023. Benchmarking of Hardware-Efficient Real-Time Neural Decoding in Brain-computer Interfaces. *Authorea Preprints* (2023).

[55] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and< 0.5 MB Model Size. *arXiv preprint arXiv:1602.07360* (2016).

[56] MoonHyung Jang, Wei-Han Yu, Changuk Lee, Maddy Hays, Pingyu Wang, Nick Vitale, Pulkit Tandon, Pumiao Yan, Pui-In Mak, Youngcheol Chae, et al. 2023. A 1024-Channel 268 nW/pixel 36x36 μm 2/ch Data-Compressive Neural Recording IC for High-Bandwidth Brain-Computer Interfaces. In *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 1–2.

[57] Yaoyao Jia, Ulkuhan Guler, Yen-Pang Lai, Yan Gong, Arthur Weber, Wen Li, and Maysam Ghovanloo. 2020. A Trimodal Wireless Implantable Neural Interface System-on-Chip. *IEEE Transactions on Biomedical Circuits and Systems* 14, 6 (2020), 1207–1217.

[58] Jaeyoung Jung, Siqi Zhu, Peng Liu, Yi-Jan Emery Chen, and Deukhyoun Heo. 2010. 22-pJ/bit Energy-Efficient 2.4-GHz Implantable OOK Transmitter for Wireless Biotelemetry Systems: In Vitro Experiments Using Rat Skin-Mimic. *IEEE Transactions on Microwave Theory and Techniques* 58, 12 (2010), 4102–4111.

[59] Taesung Jung, Nanyu Zeng, Jason D Fabbri, Guy Eichler, Zhe Li, Konstantin Willeke, Katie E Wingel, Agrita Dubey, Rizwan Huq, Mohit Sharma, et al. 2024. Stable, Chronic In-Vivo Recordings from a Fully Wireless Subdural-Contained 65,536-Electrode Brain-Computer Interface Device. *bioRxiv* (2024), 2024–05.

[60] Daniel Justus, John Brennan, Stephen Bonner, and Andrew Stephen McGough. 2018. Predicting the Computational Cost of Deep Learning Models. In *2018 IEEE international conference on big data (Big Data)*. IEEE, 3873–3882.

[61] Sidrat Tasawoor Kanth and Supratim Ray. 2020. Electrocorticogram (ECoG) is Highly Informative in Primate Visual Cortex. *Journal of Neuroscience* 40, 12 (2020), 2430–2444.

[62] Ioannis Karageorgos, Karthik Sriram, Ján Veselỳ, Michael Wu, Marc Powell, David Borton, Rajit Manohar, and Abhishek Bhattacharjee. 2020. Hardware-Software Co-Design for Brain-Computer Interfaces. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 391–404.

[63] Peter R Kinget. 2015. Scaling Analog Circuits into Deep Nanoscale CMOS: Obstacles and Ways to Overcome Them. In *2015 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 1–8.

[64] David Kleinfeld, Lan Luan, Partha P Mitra, Jacob T Robinson, Rahul Sarpeshkar, Kenneth Shepard, Chong Xie, and Timothy D Harris. 2019. Can One Concurrently Record Electrical Spikes from Every Neuron in a Mammalian Brain? *Neuron* 103, 6 (2019), 1005–1015.

[65] Peter E Konrad, Kate R Gelman, Jesse Lawrence, Sanjay Bhatia, Jacqueline Dister, Radhey Sharma, Elton Ho, Yoon Woo Byun, Craig H Mermel, and Benjamin I Rapoport. 2025. First-in-Human Experience Performing High-Resolution Cortical Mapping Using a Novel Microelectrode Array Containing 1024 Electrodes. *Journal of Neural Engineering* (2025).

[66] Andrea Kübler. 2020. The history of BCI: From a Vision for the Future to Real Support for Personhood in People with Locked-in Syndrome. *Neuroethics* 13, 2 (2020), 163–180.

[67] Dhireesha Kudithipudi, Catherine Schuman, Craig M Vineyard, Tej Pandit, Cory Merkel, Rajkumar Kubendran, James B Aimone, Garrick Orchard, Christian Mayr, Ryad Benosman, et al. 2025. Neuromorphic Computing at Scale. *Nature* 637, 8047 (2025), 801–812.

[68] Arvind Kumar, Zhe Wan, Winfried W Wilcke, and Subramanian S Iyer. 2017. Toward Human-Scale Brain Computing Using 3D Wafer Scale Integration. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 13, 3 (2017), 1–21.

[69] Frances Y Kuo and Ian H Sloan. 2005. Lifting the Curse of Dimensionality. *Notices of the AMS* 52, 11 (2005), 1320–1328.

[70] John Lee, Max Dabagia, Eva Dyer, and Christopher Rozell. 2019. Hierarchical Optimal Transport for Multimodal Distribution Alignment. *Advances in Neural Information Processing Systems* 32 (2019).

[71] Michael S Lewicki. 1998. A Review of Methods for Spike Sorting: The Detection and Classification of Neural Action Potentials. *Network: Computation in Neural Systems* 9, 4 (1998), R53.

[72] Chin-Teng Lin, Che-Jui Chang, Bor-Shyh Lin, Shao-Hang Hung, Chih-Feng Chao, and I-Jan Wang. 2010. A Real-Time Wireless Brain-Computer Interface System for Drowsiness Detection. *IEEE Transactions on Biomedical Circuits and Systems* 4, 4 (2010), 214–222.

[73] Feng Lin, Kun Woo Cho, Chen Song, Wenyao Xu, and Zhanpeng Jin. 2018. Brain Password: A Secure and Truly Cancelable Brain Biometrics for Smart Headwear. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 296–309.

[74] Xiayun Liu, Mehran M Izad, Libin Yao, and Chun-Huat Heng. 2014. A 13 pJ/bit 900 MHz QPSK/16-QAM Band Shaped Transmitter based on Injection Locking and Digital PA for Biomedical Applications. *IEEE Journal of Solid-State Circuits* 49, 11 (2014), 2408–2421.

[75] Antonio Lozano, Juan Sebastián Suárez, Cristina Soto-Sánchez, Javier Garrigós, J Javier Martínez-Alvarez, J Manuel Ferrández, and Eduardo Fernández. 2020. Neurolight: A Deep Learning Neural Interface for Cortical Visual Prostheses. *International Journal of Neural Systems* 30, 09 (2020), 2050045.

[76] Shiyu Luo, Qinwan Rabbani, and Nathan E Crone. 2022. Brain-Computer Interface: Applications to Speech Decoding and Synthesis to Augment Communication. *Neurotherapeutics* 19, 1 (2022), 263–273.

[77] Paolo Mantovani, Davide Giri, Giuseppe Di Guglielmo, Luca Piccolboni, Joseph Zuckerman, Emilio G Cota, Michele Petracca, Christian Pilato, and Luca P Carloni. 2020. Agile SoC development with open ESP. In *Proceedings of the 39th International Conference on Computer-Aided Design*. 1–9.

[78] Mackenzie Weygandt Mathis, Adriana Perez Rotondo, Edward F Chang, Andreas S Tolias, and Alexander Mathis. 2024. Decoding the Brain: From Neural Representations to Mechanistic Models. *Cell* 187, 21 (2024), 5814–5832.

[79] Adnan Mehonic and Anthony J Kenyon. 2022. Brain-Inspired Computing Needs a Master Plan. *Nature* 604, 7905 (2022), 255–260.

[80] Corinne S Mestais, Guillaume Charvet, Fabien Sauter-Starace, Michael Foerster, David Ratel, and Alim Louis Benabid. 2014. WIMAGINE: Wireless 64-Channel ECoG Recording Implant for Long Term Clinical Applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23, 1 (2014), 10–21.

[81] José del R Millán, Rüdiger Rupp, Gernot Mueller-Putz, Roderick Murray-Smith, Claudio Giugliemma, Michael Tangermann, Carmen Vidaurre, Febo Cincotti, Andrea Kubler, Robert Leeb, et al. 2010. Combining Brain–computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges. *Frontiers in Neuroscience* (2010), 161.

[82] Sajjad Moazeni, Eric H Pollmann, Vivek Boominathan, Filipe A Cardoso, Jacob T Robinson, Ashok Veeraraghavan, and Kenneth L Shepard. 2021. A Mechanically Flexible, Implantable Neural Interface for Computational Imaging and Optogenetic Stimulation Over 5.4× 5.4 mm 2 FoV. *IEEE Transactions on Biomedical Circuits and Systems* 15, 6 (2021), 1295–1305.

[83] Laurent C Moreaux, Dimitri Yatsenko, Wesley D Sacher, Jaebin Choi, Changhyuk Lee, Nicole J Kubat, R James Cotton, Edward S Boyden, Michael Z Lin, Lin Tian, et al. 2020. Integrated Neurophotonics: Toward Dense Volumetric Interrogation

of Brain Circuit Activity—at Depth and in Real Time. *Neuron* 108, 1 (2020), 66–92.

[84] Rikky Muller, Hanh-Phuc Le, Wen Li, Peter Ledochowitsch, Simone Gambini, Toni Bjorninen, Aaron Koralek, Jose M Carmena, Michel M Maharbiz, Elad Alon, et al. 2014. A Minimally Invasive 64-Channel Wireless μECoG Implant. *IEEE Journal of Solid-State Circuits* 50, 1 (2014), 344–359.

[85] Elon Musk et al. 2019. An Integrated Brain-Machine Interface Platform with Thousands of Channels. *Journal of medical Internet research* 21, 10 (2019), e16194.

[86] Scott F Owen, Max H Liu, and Anatol C Kreitzer. 2019. Thermal Constraints on in vivo Optogenetic Manipulations. *Nature Neuroscience* 22, 7 (2019), 1061–1065.

[87] Thomas J Oxley, Nicholas L Opie, Sam E John, Gil S Rind, Stephen M Ronayne, Tracey L Wheeler, Jack W Judy, Alan J McDonald, Anthony Dornom, Timothy JH Lovell, et al. 2016. Minimally Invasive Endovascular Stent-Electrode Array for High-Fidelity, Chronic Recordings of Cortical Neural Activity. *Nature Biotechnology* 34, 3 (2016), 320–327.

[88] Paras R Patel, Elissa J Welle, Joseph G Letner, Hao Shen, Autumn J Bullard, Ciara M Caldwell, Alexis Vega-Medina, Julianna M Richie, Hope E Thayer, Parag G Patil, et al. 2023. Utah Array Characterization and Histological Analysis of a Multi-Year implant in Non-Human Primate Motor and Sensory Cortices. *Journal of Neural Engineering* 20, 1 (2023), 014001.

[89] Eric H Pollmann, Yatin Gilhotra, Heyu Yin, and Kenneth L Shepard. 2022. Fully Implantable 192× 256 SPAD Sensor with Global-Shutter and Micro-LEDs for Bidirectional Subdural Optical Brain-Computer Interfaces. In *ESSCIRC 2022-IEEE 48th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 205–208.

[90] Nalini Pusarla, Anurag Singh, and Shrivishal Tripathi. 2022. Learning DenseNet Features from EEG based Spectrograms for Subject Independent Emotion Recognition. *Biomedical Signal Processing and Control* 74 (2022), 103485.

[91] Jan Putzeys, Bogdan C Raducanu, Alain Carton, Jef De Ceulaer, Bill Karsh, Joshua H Siegle, Nick Van Helleputte, Timothy D Harris, Barundeb Dutta, Silke Musa, et al. 2019. Neuropixels Data-Acquisition System: A Scalable Platform for Parallel Recording of 10000+ Electrophysiological Signals. *IEEE Transactions on Biomedical Circuits and Systems* 13, 6 (2019), 1635–1644.

[92] Theodore S Rappaport. 2024. *Wireless Communications: Principles and Practice*. Cambridge University Press.

[93] James Rosenthal and Matthew S Reynolds. 2019. A 1.0-Mb/s 198-pJ/bit Bluetooth Low-Energy Compatible Single Sideband Backscatter Uplink for the NeuroDisc Brain–Computer Interface. *IEEE Transactions on Microwave Theory and Techniques* 67, 10 (2019), 4015–4022.

[94] James Rosenthal, Apoorva Sharma, Eleftherios Kampianakis, and Matthew S Reynolds. 2019. A 25 Mbps, 12.4 pJ/b DQPSK Backscatter Data Uplink for the Neurodisc Brain–Computer Interface. *IEEE Transactions on Biomedical Circuits and Systems* 13, 5 (2019), 858–867.

[95] Rüdiger Rupp, Sonja C Kleih, Robert Leeb, José del R. Millan, Andrea Kübler, and Gernot R Müller-Putz. 2014. Brain–computer interfaces and assistive technology. *Brain-Computer-Interfaces in their Ethical, Social and Cultural Contexts* (2014), 7–38.

[96] Joe Saad, Adrian Evans, Ilan Jaoui, Victor Roux-Sibillon, Emmanuel Hardy, and Lorena Anghel. 2025. Comparison Metrics and Power Trade-Offs for BCI Motor Decoding Circuit Design. *Frontiers in Human Neuroscience* 19 (2025), 1547074.

[97] Joe Saad, Adrian Evans, Ivan Miro Panades, Tetiana Aksenova, and Lorena Anghel. 2023. Towards Low-Power Embedded ECoG Decoding. In *GDR SoC2*.

[98] Kunal Sahasrabuddhe, Aamir A. Khan, Aditya P. Singh, Tyler M. Stern, Yeena Ng, Aleksandar Tadić, Peter Orel, Chris LaReau, Daniel Pouzzner, Kurtis Nishimura, Kevin M. Boergens, Sashank Shivakumar, Matthew S. Hopper, Bryan Kerr, Mina-Elraheb S. Hanna, Robert J. Edgington, Ingrid McNamara, Devin Fell, Peng Gao, Amir Babaie-Fishani, Sampsa Veijalainen, Alexander V. Klekachev, Alison M. Stuckey, Bert Luyssaert, Takashi D. Y. Kozai, Chong Xie, Vikash Gilja, Bart Dierickx, Yifan Kong, Malgorzata Straka, Harbaljit S. Sohal, and Matthew R. Angle. 2020. The Argo: A 65,536 Channel Recording System for High Density Neural Recording in Vivo. *bioRxiv* (2020).

[99] Cristiano Santos, Pascal Vivet, Jean-Philippe Colonna, Perceval Coudrain, and Ricardo Reis. 2014. Thermal Performance of 3D ICs: Analysis and Alternatives. In *2014 International 3D Systems Integration Conference (3DIC)*. IEEE, 1–7.

[100] Gerwin Schalk and Eric C Leuthardt. 2011. Brain-Computer Interfaces Using Electrocorticographic Signals. *IEEE Reviews in Biomedical Engineering* 4 (2011), 140–154.

[101] Claudia Serrano-Amenos, Frank Hu, Po T Wang, Spencer Kellis, Richard A Andersen, Charles Y Liu, Payam Heydari, An H Do, and Zoran Nenadic. 2020. Thermal Analysis of a Skull Implant in Brain-Computer Interfaces. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 3066–3069.

[102] Hardik Sharma, Jongse Park, Naveen Suda, Liangzhen Lai, Benson Chau, Joon Kyung Kim, Vikas Chandra, and Hadi Esmaeilzadeh. 2018. Bit Fusion: Bit-Level Dynamically Composable Architecture for Accelerating Deep Neural Network. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 764–775.

[103] Rahul Sharma, Minju Kim, and Akansha Gupta. 2022. Motor Imagery Classification in Brain-Machine Interface with Machine Learning Algorithms: Classical Approach to Multi-Layer Perceptron Model. *Biomedical Signal Processing and Control* 71 (2022), 103101.

[104] Yili Shen, Changgui Yang, Yunshan Zhang, Weixiao Wang, Yuxuan Luo, Chaonan Yu, Kedi Xu, Gang Pan, and Bo Zhao. 2024. A Battery-Free Neural-Recording Chip Achieving 5.5 cm Fully-Implanted Depth by Galvanically-Switching Passive Body Channel Communication. *IEEE Journal of Solid-State Circuits* (2024).

[105] Hao Sheng, Ren Liu, Qiang Li, Zuwan Lin, Yichun He, Thomas S Blum, Hao Zhao, Xin Tang, Wenbo Wang, Lishuai Jin, et al. 2025. Brain Implantation of Soft Bioelectronics via Embryonic Development. *Nature* (2025), 1–11.

[106] John D Simeral, Thomas Hosman, Jad Saab, Sharlene N Flesher, Marco Vilela, Brian Franco, Jessica N Kelemen, David M Brandman, John G Ciancibello, Paymon G Rezaii, et al. 2021. Home Use of a Percutaneous Wireless Intracortical Brain-Computer Interface by Individuals with Tetraplegia. *IEEE Transactions on Biomedical Engineering* 68, 7 (2021), 2313–2325.

[107] Sebastian Simmich, Andreas Bahr, and Robert Rieger. 2021. Noise Efficient Integrated Amplifier Designs for Biomedical Applications. *Electronics* 10, 13 (2021), 1522.

[108] Maciej Śliwowski, Matthieu Martin, Antoine Souloumiac, Pierre Blanchart, and Tetiana Aksenova. 2022. Decoding ECoG Signal into 3D Hand Translation Using Deep Learning. *Journal of Neural Engineering* 19, 2 (2022), 026023.

[109] Maciej Śliwowski, Matthieu Martin, Antoine Souloumiac, Pierre Blanchart, and Tetiana Aksenova. 2023. Impact of Dataset Size and Long-Term ECoG-based BCI Usage on Deep Learning Decoders Performance. *Frontiers in Human Neuroscience* 17 (2023), 1111645.

[110] Karthik Sriram, Ioannis Karageorgos, Xiayuan Wen, Ján Veselỳ, Nick Lindsay, Michael Wu, Lenny Khazan, Raghavendra Pradyumna Pothukuchi, Rajit Manohar, and Abhishek Bhattacharjee. 2023. HALO: A Hardware–Software Co-Designed Processor for Brain–Computer Interfaces. *IEEE Micro* 43, 3 (2023), 64–72.

[111] Karthik Sriram, Raghavendra Pradyumna Pothukuchi, Michał Gerasimiuk, Muhammed Ugur, Oliver Ye, Rajit Manohar, Anurag Khandelwal, and Abhishek Bhattacharjee. 2023. SCALO: An Accelerator-Rich Distributed System for Scalable Brain-Computer Interfacing. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*. 1–20.

[112] Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. 2021. Neuropixels 2.0: A Miniaturized High-Density Probe for Stable, Long-Term Brain Recordings. *Science* 372, 6539 (2021), eabf4588.

[113] Ian H Stevenson and Konrad P Kording. 2011. How Advances in Neural Recording Affect Data Analysis. *Nature Neuroscience* 14, 2 (2011), 139–142.

[114] Peter Sykacek, Stephen J Roberts, and Maria Stokes. 2004. Adaptive BCI based on Variational Bayesian Kalman Filtering: An Empirical Evaluation. *IEEE Transactions on Biomedical Engineering* 51, 5 (2004), 719–727.

[115] Xin Tang, Hao Shen, Siyuan Zhao, Na Li, and Jia Liu. 2023. Flexible Brain–Computer Interfaces. *Nature Electronics* 6, 2 (2023), 109–118.

[116] Christopher Torng. 2020. FreePDK45 and the Nangate Open Cell Library. https://mflowgen.readthedocs.io/en/latest/stdlib-freepdk45.html Accessed: 2025-03-27.

[117] Goran Udovičić, Ante Topić, and Mladen Russo. 2016. Wearable Technologies for Smart Environments: A Review with Emphasis on BCI. In *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 1–9.

[118] Muhammed Ugur, Raghavendra Pothukuchi, and Abhishek Bhattacharjee. 2024. Swapping-Centric Neural Recording Systems. *15th Annual Non-Volatile Memories Workshop* (2024).

[119] Anne E Urai, Brent Doiron, Andrew M Leifer, and Anne K Churchland. 2022. Large-Scale Neural Recordings Call for New Insights to Link Brain and Behavior. *Nature Neuroscience* 25, 1 (2022), 11–19.

[120] Maureen van der Grinten, Jaap de Ruyter van Steveninck, Antonio Lozano, Laura Pijnacker, Bodo Rueckauer, Pieter Roelfsema, Marcel van Gerven, Richard van Wezel, Umut Güçlü, and Yağmur Güçlütürk. 2024. Towards Biologically Plausible Phosphene Simulation for the Differentiable Optimization of Visual Cortical Prostheses. *Elife* 13 (2024), e85812.

[121] Kaido Värbu, Naveed Muhammad, and Yar Muhammad. 2022. Past, Present, and Future of EEG-based BCI Applications. *Sensors* 22, 9 (2022), 3331.

[122] Naveen Verma, Ali Shoeb, Jose Bohorquez, Joel Dawson, John Guttag, and Anantha P Chandrakasan. 2010. A Micro-Power EEG Acquisition SoC with Integrated Feature Extraction Processor for a Chronic Seizure Detection System. *IEEE Journal of Solid-State Circuits* 45, 4 (2010), 804–816.

[123] Jacques J Vidal. 1973. Toward Direct Brain-Computer Communication. *Annual Review of Biophysics and Bioengineering* 2, 1 (1973), 157–180.

[124] Marco Vilela and Leigh R Hochberg. 2020. Applications of Brain-Computer Interfaces to the Control of Robotic and Prosthetic Arms. *Handbook of Clinical Neurology* 168 (2020), 87–99.

[125] Shiwei Wang, Seyed Kasra Garakoui, Hosung Chun, Didac Gomez Salinas, Wim Sijbers, Jan Putzeys, Ewout Martens, Jan Craninckx, Nick Van Helleputte, and Carolina Mora Lopez. 2019. A Compact Quad-Shank CMOS Neural Probe with

5,120 Addressable Recording Sites and 384 Fully Differential Parallel Channels. *IEEE Transactions on Biomedical Circuits and Systems* 13, 6 (2019), 1625–1634.

[126] Yang Wang, Xinze Yang, Xiwen Zhang, Yijun Wang, and Weihua Pei. 2023. Implantable Intracortical Microelectrodes: Reviewing the Present with a Focus on the Future. *Microsystems & Nanoengineering* 9, 1 (2023), 7.

[127] Stephan Wojtowytsch and E Weinan. 2020. Can Shallow Neural Networks Beat the Curse of Dimensionality? a Mean Field Training Perspective. *IEEE Transactions on Artificial Intelligence* 1, 2 (2020), 121–129.

[128] Patrick D Wolf and WM Reichert. 2008. Thermal Considerations for the Design of an Implanted Cortical Brain–Machine Interface (BMI). *Indwelling Neural Implants: Strategies for Contending with the In Vivo Environment* (2008), 33–38.

[129] Di Wu, Jingjie Li, Zhewen Pan, Younghyun Kim, and Joshua San Miguel. 2022. uBrain: A Unary Brain Computer Interface. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 468–481.

[130] Fan Wu, Eran Stark, Pei-Cheng Ku, Kensall D Wise, György Buzsáki, and Euisik Yoon. 2015. Monolithically Integrated μLEDs on Silicon Neural Probes for High-Resolution Optogenetic Studies in Behaving Animals. *Neuron* 88, 6 (2015), 1136–1148.

[131] Wei Wu, M Black, Yun Gao, M Serruya, A Shaikhouni, J Donoghue, and Elie Bienenstock. 2002. Neural Decoding of Cursor Motion Using a Kalman Filter. *Advances in Neural Information Processing Systems* 15 (2002).

[132] Changgui Yang, Yunshan Zhang, Ziyi Chang, Zhuhao Li, Tianyu Zheng, Yuxuan Luo, Shaomin Zhang, Kedi Xu, Gang Pan, and Bo Zhao. 2022. A 0.4 mm 3 Battery-Less Crystal-Less Neural-Recording SoC Achieving 1.6 cm Backscattering Range with 2mm× 2mm On-Chip Antenna. In *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 164–165.

[133] Yilong Yang, Qingfeng Wu, Yazhen Fu, and Xiaowei Chen. 2018. Continuous Convolutional Neural Network with 3D Input for EEG-based Emotion Recognition. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part VII 25*. Springer, 433–443.

[134] Lin Yao, Bingzhao Zhu, and Mahsa Shoaran. 2022. Fast and accurate decoding of finger movements from ECoG through Riemannian features and modern machine learning techniques. *Journal of Neural Engineering* 19, 1 (2022), 016037.

[135] Ofer Yizhar, Lief E Fenno, Thomas J Davidson, Murtaza Mogri, and Karl Deisseroth. 2011. Optogenetics in Neural Systems. *Neuron* 71, 1 (2011), 9–34.

[136] Do-Yeon Yoon, Sonal Pinto, SungWon Chung, Paul Merolla, Thong-Wei Koh, and Dongjin Seo. 2021. A 1024-Channel Simultaneous Recording Neural SoC with Stimulation and Real-Time Spike Detection. In *2021 Symposium on VLSI Circuits*. IEEE, 1–2.

[137] Anthony M Zador. 2019. A Critique of Pure Learning and what Artificial Neural Networks can learn from Animal Brains. *Nature communications* 10, 1 (2019), 3770.

[138] Hamed Zaer, Ashlesha Deshmukh, Dariusz Orlowski, Wei Fan, Pierre-Hugues Prouvot, Andreas Nørgaard Glud, Morten Bjørn Jensen, Esben Schjødt Worm, Slávka Lukacova, Trine Werenberg Mikkelsen, et al. 2021. An Intracortical Implantable Brain-Computer Interface for Telemetric Real-Time Recording and Manipulation of Neuronal Circuits for Closed-Loop Intervention. *Frontiers in Human Neuroscience* 15 (2021), 618626.

[139] Nanyu Zeng, Taesung Jung, Mohit Sharma, Guy Eichler, Jason Fabbri, R James Cotton, Eleonora Spinazzi, Brett Youngerman, Luca Carloni, and Kenneth L Shepard. 2023. A Wireless, Mechanically Flexible, 25μm-Thick, 65,536-Channel Subdural Surface Recording and Stimulating Microelectrode Array with Integrated Antennas. In *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 1–2.

[140] Weihao Zhang, Songchen Ma, Xinglong Ji, Xue Liu, Yuqing Cong, and Luping Shi. 2024. The Development of General-Purpose Brain-Inspired Computing. *Nature Electronics* (2024), 1–12.

A Artifact Appendix

A.1 Abstract

This artifact appendix describes how to evaluate system-on-chip (SoC) architectures using the MINDFUL analytical framework and how to reproduce the results presented for communication-centric and computation-centric architectures in Section 5 and Section 6.

A.2 Artifact check-list (meta-information)

- **Algorithm:** Custom algorithms based on the equations described in Section 4.2, Section 4.3, and Section 5.
- **Program:** Python implementation of MINDFUL through multiple scripts. Each script generates a specific set of experimental results or visualizes a previously generated set of results.
- **Data set:** Parameters correspond to SoCs 1–8 from Table 1.
- **Run-time environment:** The program was tested using Python 3.9 running on macOS Sequoia 15.5. The program is expected to run successfully on any PC and operating system as long as installation passed successfully (Section A.4).
- **Execution:** Command-line Python scripts for generating numerical and visual results.
- **Metrics:** Power consumption, number of integrated neural interface (NI) channels, QAM efficiency, and DNN model size,
- **Output:** Raw text files containing numerical results corresponding to the scaling of each SoC design. The results are parsed into visual plots.
- **Experiments:** Communication-centric architectures with energy-efficient modulation (Fig. 5 and Fig. 6). Communication-centric architectures with advanced modulation (Fig. 7). Computation-centric architectures with on-implant DNNs (Fig. 10). Computation-centric architectures with DNN partitioning (Fig. 11). Computation-centric architectures with combined optimizations (Fig. 12).
- **How much disk space required (approximately)?** With the default experimental setup around 500 MB.
- **How much time is needed to complete experiments (approximately)?** Up to 45 minutes to run all the scripts.
- **Publicly available?** Yes.
- **Code licenses (if publicly available)?** Apache 2.0.
- **Archived (provide DOI)?:** https://doi.org/10.5281/zenodo.16734141

A.3 Description

A.3.1 How to access. The MINDFUL framework is publicly available on GitHub (https://github.com/GuyEichler/MINDFUL_MICRO25/). The repository includes all Python scripts, configuration files, and sample parameter sets needed to reproduce the results presented in the paper. A DOI will be provided upon archival.

A.3.2 Software dependencies. The code has been tested with Python 3.9 and requires installing the following packages:

- `matplotlib`
- `numpy`
- `scipy`
- `sympy`
- `shapely`
- `math` (built-in)
- `sys` (built-in)
- `os` (built-in)

A.3.3 Data sets. This artifact does not use external data sets. Instead, it includes a set of internal configuration parameters representing implanted BCI SoC designs. These configurations correspond to SoCs 1–8 from Table 1 and are used to reproduce the results in Section 5 and Section 6. Additional SoC configurations can be added by modifying or extending the parameter files provided in the repository. The artifact also includes our MAC post-synthesis parameters using 45nm and 12nm technology nodes. Finally, the artifact executes the experiments with the parameters of the MLP and DN-CNN computation models described in the paper. Additional computational models can be tested as well but require implementing specialized methods for each model (more details in Section A.7).

A.4 Installation

Before running the framework, ensure that all required Python packages are installed. The core dependencies are listed in Section A.3.2. Once the environment is ready, create the necessary output directories by running:

```
$ python create_dirs.py
```

A.5 Experiment workflow

The Python scripts at the top level of the repository correspond directly to the experiments and figures in the paper:

- To generate the results corresponding to communication-centric architectures with energy-efficient modulation (Fig. 5 and Fig. 6):

  ```
  $ python comm_centric_ook.py naive
  $ python comm_centric_ook.py high_margin
  ```

  where `comm_centric_ook.py` generates the results according to the type of design given as argument.

- To generate the results corresponding to communication-centric architectures with advanced modulation (Fig. 7):

  ```
  $ python comm_centric_qam.py
  ```

  where `comm_centric_qam.py` generates the results.

- To generate the results corresponding to computation-centric architectures with on-implant DNNs (Fig. 10):

  ```
  $ python comp_centric_dnn.py
  ```

  where `comp_centric_dnn.py` generates the results.

- To generate the results corresponding to computation-centric architectures with DNN partitioning (Fig. 11):

  ```
  $ python comp_centric_dnn.py
  $ python comp_centric_dnn_layers.py
  ```

  where `comp_centric_dnn.py` generates the results without DNN partitioning, `comp_centric_dnn_layers.py` generates the results with DNN partitioning. Running these scripts generates two files:
  - `socs_intersections.py`
  - `socs_intersections_layers.py`

The first file includes the the maximum number of NI channels per SoC without DNN partitioning and the second file with DNN partitioning. Both files are required to generate Fig. 11.

- To generate the results corresponding to computation-centric architectures with combined optimizations (Fig. 12):

```
$ python comp_centric_optimization.py
```

where `comp_centric_optimization.py` generates the results.

A.5.1 Output Generation. The framework saves output in three different folders, each serving a specific purpose:

- **`data/`** — Contains the raw numerical results generated by the analysis scripts. These include intermediate computations and final values used in plots, saved in plain text format.
- **`logs/`** — Contains logs produced during the execution of the scripts. These logs capture runtime information, debug traces, and configuration details used in each experiment run.
- **`figures/`** — Contains the plots and visualizations generated by the framework. These include all figures corresponding to the results presented in the paper, saved in PDF format.

A.6 Evaluation and Expected Results

After running the analysis scripts described in Section A.5 to reproduce the results, follow the steps below to parse and view the results as presented in Figures 5-7 and Figures 10-12:

- Running the following commands creates the subfigures of Fig. 5 and Fig. 6 describing the power and area scaling results of SoCs 1-8 in the cases of naive and high-margin designs:

```
$ python comm_centric_ook_show.py naive
$ python comm_centric_ook.py high_margin
```

The subfigures are saved under:
– `figures/high_margin_power_scaling_socs.pdf`
– `figures/high_margin_area_scaling_socs.pdf`
– `figures/naive_power_scaling_socs.pdf`
– `figures/naive_area_scaling_socs.pdf`
Data files: `data/ook_data<SoC#>_<0_or_1>.txt` (0 for high-margin and 1 for naive).
Log file: `logs/log_ook_output.txt`.

- Running the following command creates Fig. 7 describing the QAM efficiency results after scaling SoCs 1-8:

```
$ python comm_centric_qam_show.py
```

The figure is saved under:
– `figures/qam_comm_scaling_socs.pdf`
Data files: `data/qam_data<SoC#>.txt`.
Log file: `logs/log_qam_output.txt`.

- Running the following command creates the subfigures of Fig. 10 describing the power consumption results after scaling SoCs 1-8, while integrating on-implant DNNs (MLP and DN-CNN):

```
$ python comp_centric_dnn_show.py
```

The subfigures are saved under:
– `figures/mlp_comp_scaling_socs.pdf` (MLP)
– `figures/dense_comp_scaling_socs.pdf` (DN-CNN)
Data files: `data/<mlp_or_dense>_comp_data<SoC#>.txt`.
Log file: `logs/log_dnn_output.txt`.

- Running the following command creates Fig. 11 describing the gains in NI channel counts when scaling SoCs 1-8, while integrating on-implant DNNs and enabling the DNN partitioning optimization:

```
$ python comp_centric_dnn_layers_compare.py
```

The figure is saved under:
– `figures/compare_layers_scaling_socs.pdf`
Data files: `data/<mlp_or_dense>_comp_layers_data<SoC#>.txt`.
Log file: `logs/log_dnn_layers_output.txt`.

- Running the following command creates the subfigures in Fig. 12 describing the MLP model size when scaling SoCs 1-8, while using combined optimization techniques:

```
$ python comp_centric_optimization_show.py
```

The subfigures are saved under:
– `figures/<SoC#>_MLP_all_opt_scale_model.pdf`
Data files: `data/<1_or_2>_MLP_all_opt_data<SoC#>.txt` (1 for power consumption data and 2 for model size data).
Log file: `logs/log_opt_output.txt`.

A.7 Experiment Customization

The framework supports customization for evaluating additional SoC designs and introducing new computation models.

A.7.1 Customizing SoC designs. SoC designs can be added to the analysis by editing the file:

- `settings/socs_to_test.py`

The file includes the post-synthesis parameters of our MAC units, and the parameters of all SoC designs to be tested through the experiments described in Section A.5. Each SoC is represented by a Python dictionary structure and users can:

- Add a new SoC entry with a custom name and parameter values.
- Modify or replace existing entries to test alternative designs.
- Update post-synthesis hardware parameters, such as MAC unit characteristics, using values obtained from synthesis reports.

A.7.2 Adding On-Implant Computation Models. To evaluate additional computation models, users must modify the following components:

- `Settings/structs.py` – Add a new data structure that describes the architecture of the network to be tested.
- `framework/networks.py` – Implement all relevant functions for the new model, following the structure of existing examples such as the MLP and DN-CNN networks.

The implementation in `networks.py` should include all required logic to compute the parameters used by the analytical framework. Users are encouraged to follow the format of `calc_mlp` and `calc_densenet` as templates.

Once added, the new model can be referenced in the evaluation scripts for integration into the MINDFUL analysis flow.