NER '25 Problem Outline

Shaan, Bernardo, Alejandro, Muhammed

1 Introduction

```
Introduction:
Compute on BCIs is important
What methods fit on device?
X tool solves this problem
easy to integrate into existing workflows
validated with real hardware

Motivation:
2 algorithms contrast them with hardware availability

Description  of the tool:
GUI

Validation results

Future work and other uses:
FDA and others
```

Brain-computer Interfaces (BCIs) provide opportunities to treat neurological disorders, understand brain functionalities and interface brains with the digital world. As the field of BCIs has matured, it has given rise to more complex and data-intensive models that require efficient, real-time processing to be effective. These models can no longer rely on generic processors to meet their bespoke computational demands; instead, they require the design of custom hardware to support their models.

Neuroengineers commonly use software simulations to prototype and evaluate novel algorithms for BCIs. However, neuroengineers are not usually experts in hardware design and so often can't make the best choices for their use cases. This leaves a lot of performance and potential errors in their algorithms, originally designed at the software level yet implemented at the hardware level. Furthermore, today there is no clear consensus on what constitutes an optimal BCI algorithm, such as for seizure prediction, with many different ideas present and no clear way of comparing them.

All these issues necessitate a development and evaluation platform for BCIs – one that allows neuroengineers to prototype and evaluate hardware-level BCI algorithms without being expert hardware engineers.

2 Significance

The field of BCIs is growing at a rapid rate, with increasingly larger and a wider variety of computation models such as SCALO [**10.1145/3579371.3589107**]. However, there is no standardised method for prototyping and evaluating these models, leading to an increased workload for neuroengineers to test their ideas, particularly in the case of creating hardware based tests. There is a need for a standard method of developing complex hardware based BCI systems that will lead to greater productivity and product quality. Furthermore, ensuring the quality of BCI software is crucial for its acceptance and integration by a wider audience. A well-designed and tested pipeline increases trust and confidence in the technology, leading to broader adoption.

3 Similar work

There is a large variety of BCI simulators available in literature, however, they are mostly non-modular and none support hardware-level simulation of BCI algorithms. Furthermore, they are generally crafted towards the use of non-invasive BCIs and EEG signals, although it can be seen how they could be modified with a little bit of work to work with iEEG signals.

The simulators available seem to be very solution specific (ie machine learning applications, signal similarity based etc). Some of the simulators provide very good visualization and user-friendly methods of providing telemetry that would be beneficial to learn from.

Some of the notable simulators are summarised below.

BCI simulator [**bci-sim-github**] - for closed-loop, real-time decoder algorithms. It provides very good real-time visualizations, and is open-source.

Intracranial EEG Analyser [**DUBARRY2022119251**] - a software package for the analysis and visualization of intracranial signals. Has a user-friendly graphical interface that conveys information in a very interpretable manner.

Neural Data Simulator [**neural-sim-github**] - a real-time system for generating electrophysiology data from behavioral data (e.g. arm kinematics).

Can test and validate closed-loop brain-computer interface systems without the need for a human in the loop, generate synthetic data for algorithm optimization, and provide a platform on which to develop BCI decoders.

4 Differentiation

We aim to be different through 2 main functionalities: (1) hardware-based simulation and (2) providing a growing modular library of BCI elements.

We want to provide a simulator that will allow neuroengineers to develop, evaluate and test multiple different hardware-based BCI pipelines without a needing to be hardware experts. The aim is to streamline the neuroengineer ideation to implementation and evaluation pipeline.

The aim of the paper will be to show neuroengineers the benefits of hardware based simulation, and how it can impact them. Given the 4 page limit, we will try to focus on neuroengineer relevant information rather than systems innovations.

5 Visualizations

These are examples of some visualizations we would use to support the usefulness our work and ideas. The idea is to have visualizations that neuroengineers in particular can relate to and see the benefit of, less emphasis on showing the systems/architecture side.

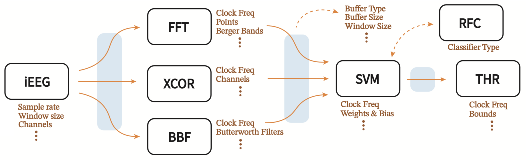

Figure 1: Automated pipeline generation

6 Validation

FPGA-based validation

Tool outputs on the x-axis, FPGA-based outputs on the y-axis. Relate the two results to one another for validation.

7 Extrapolation

ITRS Roadmap for technology scaling

Looking at other circuits papers for ”validation”, show similarity/closeness of extrapolation to existing designs in these conferences.

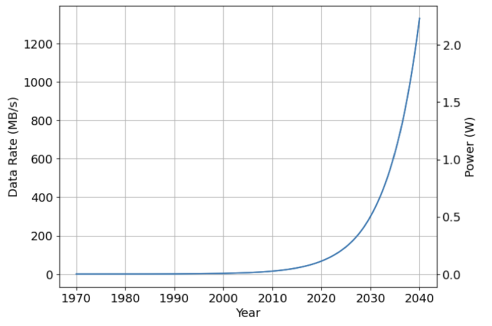

Figure 1. Exponential growth of data rates for electrophysiology-based neural recording with estimated power consumption of wireless transmission.

Figure 2: Graph showing increasing neural data rates, and so the need for custom hardware based compute

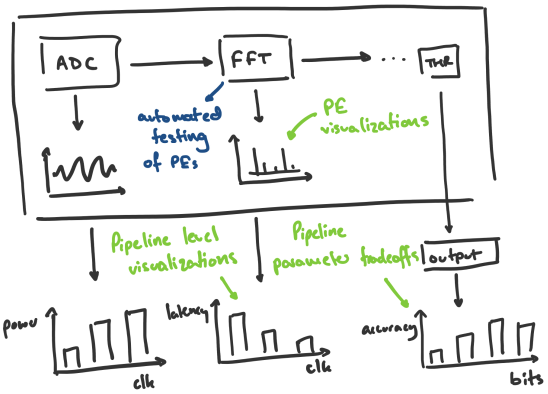

Figure 3: Illustration showing the different visualizations at different ”levels” of the pipeline

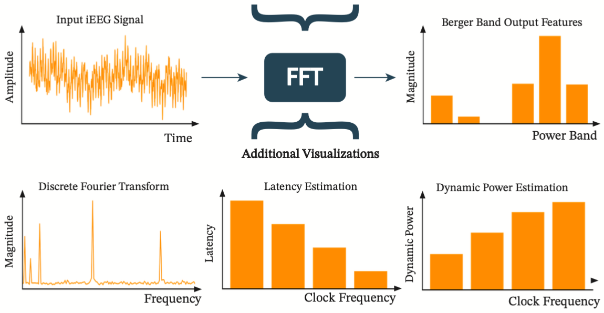

Figure 4: PE level visualizations. For NER, we want to mostly focus on algorithm validation from a hardware perspective and one or two tools (ie power estimation) to show the potential of this field.

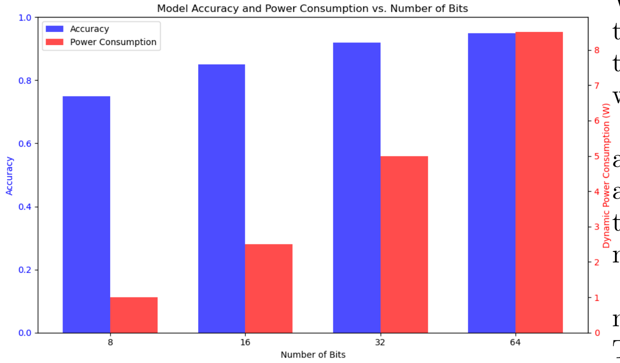

Figure 5: An example visualization that would be generated at the pipeline level for seeing the accuracy vs power tradeoff as data representation changes.

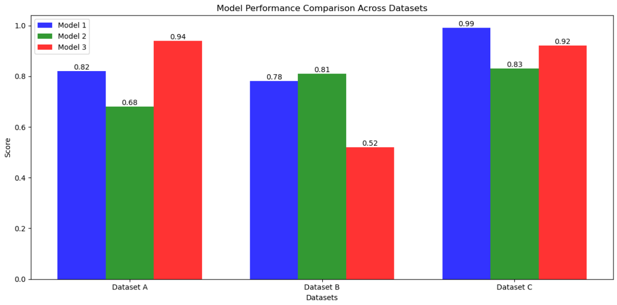

Figure 6: An example visualization that would be generated at the pipeline level to show the performance of different models across different datasets.

8 Execution Time

Include plots on the execution time of the tool.

9 Other notes

Want to keep it brief, only 4 pages for NER. Idea is to show the usefulness of the simulator specifically to neuroengineers and not go into the details of how we did things (lack of space, audience).

The paper shown by Raghav (which was a tool accepted by NER, link), seems to be very simple and very specific. The bar seems to be creating a tool that can be specifically useful for neuroengineers.

Also, for tools currently thinking of power estimation and accuracy vs amt of hardware tradeoff. Thinking of making external more accurate proprietary tools (ie tools on grace) usable within our simulator for power estimation, while providing a basic solution ourselves.

The interface of gem5 is a really good way of interfacing simulators, perhaps aim to create the equivalent of gem5 for BCI hardware with ”SimObject” like structure?

MATLAB conversion as well as Python.

10 Future Work

Making power estimation more reliable. Power estimation without Verilog. Connect the tool with an actual chip/FPGA in real-time. Automatic RTL generation from high-level code like Python.