Bridging AI and Neuroscience to Build Energy-Efficient Computers

Computers are a powerful technology and pervasive fixture in modern life, so it is imperative that they are built to enhance, not harm, society. Yet, recent trends in semiconductor manufacturing and artificial intelligence (AI) indicate a significant, growing increase in the energy consumption of computing. This poses a clear and immediate risk to climate change, and it limits future technological advances to communities with financial access. In stark contrast, biological intelligence outperforms AI in many tasks while consuming a fraction of its power. Accordingly, ***I research how to build energy-efficient AI and non-traditional computing systems by studying the brain.***

Historically, computers have been designed based on the same underlying model, the von Neumann machine, and advancement in processor speed and efficiency can be largely attributed to trends in transistor scaling [1] (i.e. Moore's law and Dennard scaling [2]). The growth in the energy cost of computing is due to two separate but concurrent phenomena. As we have approached the physical limits of transistor size, scaling trends have slowed down or ended [3]. Simultaneously, neural networks have successfully replaced traditional programs in many domains, and the medium of improvement in modern machine learning (ML) is scale, both in the size of models and of datasets. This places an immediate and unexpected burden on our compute resources [4]. Effectively tackling this problem requires improvements to both phenomena—we must build computers that bypass the inefficiencies of traditional machines, and we must reduce the energy cost of training and deploying AI models.

Biology appears to have achieved both goals. Animal brains of many species match or surpass the capabilities of state-of-the-art AI models, all while learning with limited data and consuming only $\approx 20\,\mathrm{W}$ of power. Artificial neural networks offer a framework for building hardware and algorithms based on similar principles, but existing approaches avoid studying all facets of biological computing comprehensively. Animals are embodied computing systems that sense and interact with the physical world. To solve problems on an energy budget, their brains exploit the structure and statistics of the world to balance between specialization and adaptability. Thus, a deeper understanding of biological computation can teach us not only about novel, efficient primitives (e.g., spike encoding or content-addressable memories), but also how to discover such primitives from the structure of a problem and then effectively use them. To do this, ***I propose a research programme built on bidirectional transfer of knowledge between neuroscience and computer science.*** My programme is driven by three broad aims:

1. Apply AI/ML to address neuroscience data. Modern neuroscience has obtained high-throughput, high-dimensional brain datasets. ML models allow us to study these datasets without flattening their complexity, but measurement uncertainty and low sample size push current models to their limits. Working directly with neuroscientists, *I augment existing techniques to help answer important scientific questions while also providing short-term, immediate improvements to ML.*
2. Build AI/ML models with developmental and evolutionary principles. Both in biology and AI, structural priors embedded in the parameters of models help accelerate learning. Beyond simple examples, identifying the appropriate priors in AI continues to be a challenge. In contrast, biology has identified a diverse set of priors that balance specialization and adaptability. *I study two biological processes, neuronal development and evolution, through a computational lens to build AI models endowed with prior structure.* In turn, these models are more sample-efficient and consume less energy to train.
3. Describe neural processing using the language of computer science. Neuroscience can reveal new computational primitives, and a study of development & evolution can help guide the automated discovery and composition of primitives targeted to specific problems & domains. Yet, without describing these primitives in language of computer science—such as their space or time complexity—we cannot reasonably build brain-inspired hardware or software. *While neuroscientists study neural networks as solutions to ecological problems, I study them as solutions to computational problems using the same quantitative tools applied to traditional computers.* Thus, I can provide a unique and new perspective for computational neuroscience as well as build a theoretical foundation for neuro-inspired computing.

Through Aim 1, I use modern ML techniques to help make sense of neuroscience data, focusing on topics that can provide fruitful insights for Aims 2 and 3. In the process, I focus on making *current* models more sample-efficient, resulting in more accessible and efficient AI. In turn, successful projects in Aims 2 and 3 can provide guidance for future collaboration with neuroscientists, while also providing the foundation for building *future* energy-efficient AI and computers. I believe this program, based on a two-way transfer from computer science and AI to neuroscience, can meaningfully enrich both fields.

Graduate studies

During my Ph.D. under Dr. Mikko Lipasti at University of Wisconsin-Madison, I built unconventional computing systems that operate at ultra-low power budgets. These systems were based on two different computing paradigms—stochastic (or bitstream) computing and neuromorphic computing.

Compilation of bitstream computing programs. Stochastic or bitstream computing is a power-efficient computing paradigm where information is represented by a random stream of bits over time. I developed BitSAD [5], [6], a compiler for bitstream computing programs, which enabled the creation of large-scale, complex programs. I wrote bitstream programs for navigation [7], [8], Bayesian inference [9], and deep learning [10]. Additionally, inspired by populations of biological neurons, I developed a parallel processing technique for bitstream computing leveraging the inherent randomness.

Hardware-friendly learning rules for neuromorphic computing. Neuromorphic computers are specifically designed to compute with spiking neural networks (SNNs)—a model where information is encoded as on/off events, similar to biological neurons. I developed learning rules for SNNs that are better-suited to neuromorphic hardware constraints [11].

My work demonstrates that is possible to translate ideas between neuroscience and computing systems, and I expect that similar techniques will be valuable for my future research goals.

Aim 1: Apply AI/ML to address neuroscience problems

Neuroscientists have advanced recording techniques to enable high-resolution, high-throughput acquisition from multiple sources. While this allows scientists to measure animal behavior in its full richness and complexity, making sense of the data requires advanced models. Unfortunately, the measurement noise and low sample size makes it difficult train ML models in these settings. To overcome these challenges, I leverage additional constraints available in scientific data that are largely absent in typical ML benchmarks. **Through active and future collaborations with neuroscientists, I build AI models tailored for neural datasets to answer scientific questions.**

Accurate pose-tracking for mouse facial movements (*current*). Pose-tracking models in neuroscience enable the study of complex animal behavior [12], [13]. **In collaboration with the Hou Lab at Cold Spring Harbor Lab, I apply 3D pose-tracking models to study facial expressions in mice [14].** Mouse facial movements can be small or large and occur over both short and long timescales. Using human-guided active learning, our models are reliable and accurate over the full spatiotemporal dynamic range, and they generalize across lighting conditions and mice. The predictions are accurate enough to use as a non-invasive readout of internal processes in the mouse face and brain, as we demonstrate in a series of behavioral experiments.

Sample-efficient multi-camera pose-tracking (*future*). In future extensions of this work, I aim to further increase the sample-efficiency of pose-tracking models by embedding the spatial constraints between cameras into the model training procedure. Currently, models must infer the relationship between cameras from video data alone. I plan to use available spatial calibration data between cameras to condition a model's behavior on relative camera angle. In addition to improving the sample efficiency, conditioning makes models robust to novel view angles, allowing them to generalize across labs.

Relating neural and behavioral dynamics through topological state space modeling (*future*). Computation through dynamics [15] is a popular modeling technique in neuroscience. In this framework, recurrent neural networks (RNNs) are trained to recapitulate a task or recorded data, then the dynamics of the internal state of the model are used to explain and test scientific hypotheses. Often, co-recorded auxiliary data is used for training, and one might predict the neural data from the state space dynamics of the RNN [16]. Yet, if the RNN is not carefully regularized, it is possible to train a model that predicts the auxiliary data, but is a poor proxy for the dynamics of the neural population. Topological shape analysis is a recently developed ML method for comparing high-dimensional datasets [17]. I propose applying these comparative tools to the state space trajectories of multiple independently trained RNNs on neural and auxiliary data, respectively. This approach will allow scientists to compare neural data with complex behavioral data, such as the mouse facial movements in my existing work.

Aim 2: Build AI/ML models with developmental and evolutionary principles

Deep neural networks are trained on billions of data samples, while biological networks are able to learn within a few training examples by leveraging innate priors encoded in an organism's genome via evolution. The importance of structural priors is well-known in ML, but a scalable mechanism for learning useful priors does not exist. Biology makes use of two processes for finding innate structure—neuronal development, which translates the information encoded in the genome into a functional network of neurons, and evolution, which mutates the genome to produce better networks. Both

processes are relatively under-studied in the context of AI. **I study the computational properties of development, evolution, and genomes to build AI models initialized with prior structure.** The resulting models can learn with fewer samples and transfer previously learned structure to novel tasks.

Dynamic generative models for neural network structure (*current*). Evolution does not optimize for individual brains, and instead, it selects for the optimal distribution (population) of networks for a given organism and environment. This is encoded in the genome, and neuronal development can be understood as a dynamic generative process for sampling from the learned distribution. **In on-going work with Dr. Anthony Zador at Cold Spring Harbor Lab, I modify denoising diffusion models (DDMs) to learn to sample from distributions of ML models trained with stochastic gradient descent.** The networks produced by our diffusion model have zero- or few-shot performance on tasks. Moreover, by conditioning the DDM on task descriptions, our model learns shared structure across different tasks, and we can use conditioning alone to generate networks for entirely unseen tasks. This work provides a foundation for studying the properties of development and the genome that are useful for tractably learning structure in networks.

Learning optimal genetic programs for initializing networks (*future*). During development, the genome is not translated in a single step. Instead, it is read in parts, and each stage can augment the cellular environment, affecting which portions of the genome are translated next. In this way, gene translation is akin to a program—complete with conditional execution and looping. Evolution, in turn, learns the best program for generating useful biological networks. Building on my existing work, I aim to build a technique for learning the optimal sequence of conditioning information to compose multiple DDMs (or "genetic subprograms") for a novel tasks. This will allow ML researchers to effectively leverage the structure in already trained models to generalize to new problems. The end result is AI that is accurate as well as energy- and sample-efficient.

Aim 3: Describe neural processing using the language of computer science

Unlike modern computers, brains employ a variety of different approaches for typical computation primitives such as memory, encoding, communication, and processing. This diversity allows an organism to balance between specialization and flexibility under its energy budget. A careful study of these neural circuits can be instructive for building energy-efficient computers, but neuroscientists largely focus on functional and biological explanations. **I bring the lens of a computer scientist to neural processing—understanding it in terms of computational tools and metrics such as space complexity, time complexity, capacity, and bandwidth.** By describing neural computation in the language of computer science, I aim to build a theoretical bridge between conventional and neuro-inspired computing. This is essential to build heterogeneous hardware and algorithms that mix both paradigms.

Emergent communication protocols for distributed computing (*future*). Language is central to humanity's success, but humans are far from the only species with sophisticated communication. Honeybees utilize a complex dance sequence to communicate the direction and distance of food sources to the rest of their hive. While the ecological advantage of this communication protocol is clear—better odds of survival—how this protocol emerges and how it relates to the computation distributed across the whole colony is non-trivial. Inspired by my work on mouse facial expressions, I suggest that the symbols of this communication channel match accidental motor movements that correlate with the bee's internal state. Under the limited compute budget of a single bee, using such movements to infer and communicate internal state becomes advantageous towards the hive-wide goal of collecting food. I propose to study this hypothesis with multiple interacting agents in a reinforcement learning (RL) setting with a shared objective. Agents have no explicit communication but make spurious actions revealing their state to others. Then, I aim to study the properties of the emergent communication as a function of the channel capacity, computational capacity of each agent, and shared objective. Through this research, I will arrive at a computational explanation for how communication might emerge in animals, and any theoretical insights can be used to build ad-hoc networking protocols for distributed computing systems.

Higher order functions in neural circuits (*future*). Higher order functions are an essential primitive that make composition easy in programming languages; yet, no analogue exists for neural circuits. While neural networks can be trained to implement particular functions, it is non-trivial to combine multiple pre-trained networks to perform more complex tasks. Recurrent neural networks implement dynamical systems, and the trajectory of their state space can be understood as implementing a particular function or computation. Recent work demonstrates how additional control signals can be used to manipulate the shape or geometry of these state space trajectories [18], [19]. Since higher order functions can be understood as control flow around the execution of lower order functions, I propose building a theoretical framework for higher order functions in recurrent networks via manipulating control signals. This work can help explain the computational role of top-down signaling in the brain, as well as blend the flexibility of learnable neural network functions with the compositionality of conventional programs.

[1] A. Fuchs and D. Wentzlaff, "The Accelerator Wall: Limits of Chip Specialization," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Washington, DC, USA: IEEE, Feb. 2019, pp. 1–14. doi: 10.1109/HPCA.2019.00023.

[2] G. E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, pp. 114–117, Apr. 1965.

[3] T. N. Theis and H.-S. P. Wong, "The End of Moore's Law: A New Beginning for Information Technology," *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41–50, Mar. 2017, doi: 10.1109/MCSE.2017.29.

[4] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 3645–3650. doi: 10.18653/v1/P19-1355.

[5] **K. Daruwalla**, H. Zhuo, and M. Lipasti, "BitSAD v2: Compiler Optimization and Analysis for Bitstream Computing," *Transcations on Architecture and Code Optimization*, vol. 16, no. 4, Nov. 2019, doi: 10.1145/3364999.

[6] **K. Daruwalla**, H. Zhuo, and M. Lipasti, "BitSAD: A Domain-Specific Language for Bitstream Processing," in *First ISCA Workshop on Unary Computing - June 2019*, Phoenix, AZ, USA, Jun. 2019.

[7] **K. Daruwalla**, H. Zhuo, C. Schulz, and M. Lipasti, "BitBench: a benchmark for bitstream computing," in *Proceedings of the 20th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems - LCTES 2019*, Phoenix, AZ, USA: ACM Press, 2019, pp. 177–187. doi: 10.1145/3316482.3326355.

[8] **K. Daruwalla** and M. Lipasti, "Resource Efficient Navigation Using Bitstream Computing," in *First ISCA Workshop on Unary Computing - June 2019*, Phoenix, AZ, USA, Jun. 2019.

[9] S. Khoram, **K. Daruwalla**, and M. Lipasti, "Energy-Efficient Bayesian Inference Using Bitstream Computing," *IEEE Computer Architecture Letters*, pp. 1–4, 2023, doi: 10.1109/LCA.2023.3238584.

[10] N. Joshi, **K. Daruwalla**, and M. Lipasti, "BitFit: Bitstream-Aware Training for Stochastic Neural Networks," in *Second Workshop on Unary Computing*, San Diego, CA, Apr. 2024.

[11] **K. Daruwalla** and M. Lipasti, "Information bottleneck-based Hebbian learning rule naturally ties working memory and synaptic updates," *Frontiers in Computational Neuroscience*, vol. 18, p. 1240348, May 2024, doi: 10.3389/fncom.2024.1240348.

[12] A. Mathis *et al.*, "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, vol. 21, no. 9, pp. 1281–1289, Sep. 2018, doi: 10.1038/s41593-018-0209-y.

[13] P. Karashchuk *et al.*, "Anipose: A toolkit for robust markerless 3D pose estimation," *Cell Reports*, vol. 36, no. 13, p. 109730, Sep. 2021, doi: 10.1016/j.celrep.2021.109730.

[14] **K. Daruwalla**, I. N. Martin, A. Frankel, D. Naglič, Z. Ahmad, and X. H. Hou, "A 3D whole-face movement analysis system to uncover underlying physiology in mice." Accessed: Sep. 17, 2024. [Online]. Available: http://biorxiv.org/lookup/doi/10.1101/2024.05.07.593051

[15] S. Vyas, M. D. Golub, D. Sussillo, and K. V. Shenoy, "Computation Through Neural Population Dynamics," *Annual Review of Neuroscience*, vol. 43, no. 1, pp. 249–275, Jul. 2020, doi: 10.1146/annurev-neuro-092619-094115.

[16] D. Sussillo, M. M. Churchland, M. T. Kaufman, and K. V. Shenoy, "A neural network that finds a naturalistic solution for the production of muscle activity," *Nature Neuroscience*, vol. 18, no. 7, pp. 1025–1033, Jul. 2015, doi: 10.1038/nn.4042.

[17] A. H. Williams, E. Kunz, S. Kornblith, and S. Linderman, "Generalized shape metrics on neural representations," in *Advances in neural information processing systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., Curran Associates, Inc., 2021, pp. 4738–4750. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf

[18] J. Z. Kim and D. S. Bassett, "A neural machine code and programming framework for the reservoir computer," *Nature Machine Intelligence*, Jun. 2023, doi: 10.1038/s42256-023-00668-8.

[19] J. Z. Kim, Z. Lu, E. Nozari, G. J. Pappas, and D. S. Bassett, "Teaching recurrent neural networks to infer global temporal structure from local examples," *Nature Machine Intelligence*, vol. 3, no. 4, pp. 316–323, Apr. 2021, doi: 10.1038/s42256-021-00321-2.