Communicating Research to the General Public

The WISL Award for Communicating PhD Research to the Public launched in 2010, and since then over 100 Ph.D. degree recipients have successfully included a chapter in their Ph.D. thesis communicating their research to non-specialists. The goal is to explain the candidate's scholarly research and its significance—as well as their excitement for and journey through their area of study—to a wider audience that includes family members, friends, civic groups, newspaper reporters, program officers at appropriate funding agencies, state legislators, and members of the U.S. Congress.

WISL encourages the inclusion of such chapters in all Ph.D. theses everywhere, through the cooperation of PhD candidates, their mentors, and departments. WISL offers awards of $250 for UW-Madison Ph.D. candidates in science and engineering. Candidates from other institutions may participate, but are not eligible for the cash award. WISL strongly encourages other institutions to launch similar programs.

Wisconsin Initiative for Science Literacy

The dual mission of the Wisconsin Initiative for Science Literacy is to promote literacy in science, mathematics and technology among the general public and to attract future generations to careers in research, teaching and public service.

Contact: Prof. Bassam Z. Shakhashiri

UW-Madison Department of Chemistry

bassam@chem.wisc.edu

www.scifun.org

Evolving the Neuromorphic Compute Stack: Cross-Layer Solutions for Next-Generation Brain-Inspired Computing

By

Ranganath "Bujji Setty" Selagamsetty

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2026

Date of final oral examination: July 2nd, 2026

The dissertation is approved by the following members of the Final Oral Committee:

Mikko Lipasti, Professor Emeritus, Electrical and Computer Engineering

Joshua San Miguel, Associate Professor, Electrical and Computer Engineering

Karthikeyan Sankaralingam, Professor, Computer Sciences

Mohit Gupta, Associate Professor, Computer Sciences

Georgios Tzimpragos, Assistant Professor, Electrical and Computer Engineering

Chapter 1

From Jingling Keys to Greener Chips: Translating Human Hearing into the Future of Brain-Inspired Computing

A chapter intended for non-scientists

Author: Ranganath (Bujji) Selagamsetty

Degree: Ph.D. in Computer Science, University of Wisconsin–Madison (2026)

Supported and Edited by: Elizabeth Reynolds, Cayce Osborne, Bassam Shakhashiri, and The Wisconsin Initiative for Science Literacy (WISL)

Think about how you can focus on a single voice in a noisy, crowded room. This chapter explores how the unique physical shape of the human ear filters sound, and how we can translate this biological trick into energy-

efficient, brain-inspired computer programs. However, because modern supercomputers were not designed to process information like a biological brain, training these models is incredibly slow and wasteful. To solve this challenge, I created new scheduling methods and a specialized hardware-software system called StAccato, bridging the gap between biology, software, and microchips to build the faster, greener technology of tomorrow.

1.1 Why I Wrote This Chapter and Acknowledgments

Science should never exist solely behind closed doors or within technical academic journals. As researchers, we have a responsibility to communicate our work to the public, whose taxes fund our laboratories and whose lives are impacted by technological advances. Translating complex scientific concepts into accessible stories fosters public trust, inspires future generations of scientists, and helps researchers see their own work from fresh perspectives.

I wrote this chapter to share both the technical discoveries and the personal journey behind my doctoral degree. I am deeply grateful to the Wisconsin Initiative for Science Literacy (WISL) at the University of Wisconsin–Madison for sponsoring this platform. In particular, I wish to express my sincere appreciation to Professor Bassam Z. Shakhashiri, Elizabeth Reynolds, and Cayce Osborne for their generous guidance, encouragement, and edito-

rial assistance in producing this non-specialist chapter.

Note on AI Assistance: Google's Gemini language model was used to aid in grammar correction and editing. Additionally, Google Gemini's image generation agent was to brainstorm non-technical visual concepts, generate visual illustrations, and help with caption writing in this chapter.

1.2 My Journey Navigating Graduate School

My fascination with computing began during my undergraduate and master's studies in Electrical and Computer Engineering at Carnegie Mellon University. Interested in how computer hardware interacts with software, I joined the University of Wisconsin–Madison to pursue my Ph.D. in Computer Science, aiming to bridge the worlds of biological intelligence and microchip design.

My path through graduate school brought unique structural challenges. I became one of the final Ph.D. students in my research group as my advisor officially retired in 2024. While my advisor and co-advisor provided essential high-level vision, funding, and access to supercomputing infrastructure, I conducted the day-to-day experimental work, coding, and hardware simulations independently. Navigating research in an esoteric topic like brain-inspired computing required building self-reliance, overcoming im-

poster syndrome, and learning to bridge three distinct disciplines: systems neuroscience, supercomputer scheduling, and computer architecture.

1.3 Taking a Classroom Lesson and Running with It

My research journey began unexpectedly during my first year of graduate school, sitting in a systems neuroscience class taught by Professor Tom C. Yin. We were beginning a unit on auditory processing, which is the science of how our brains make sense of what we hear.

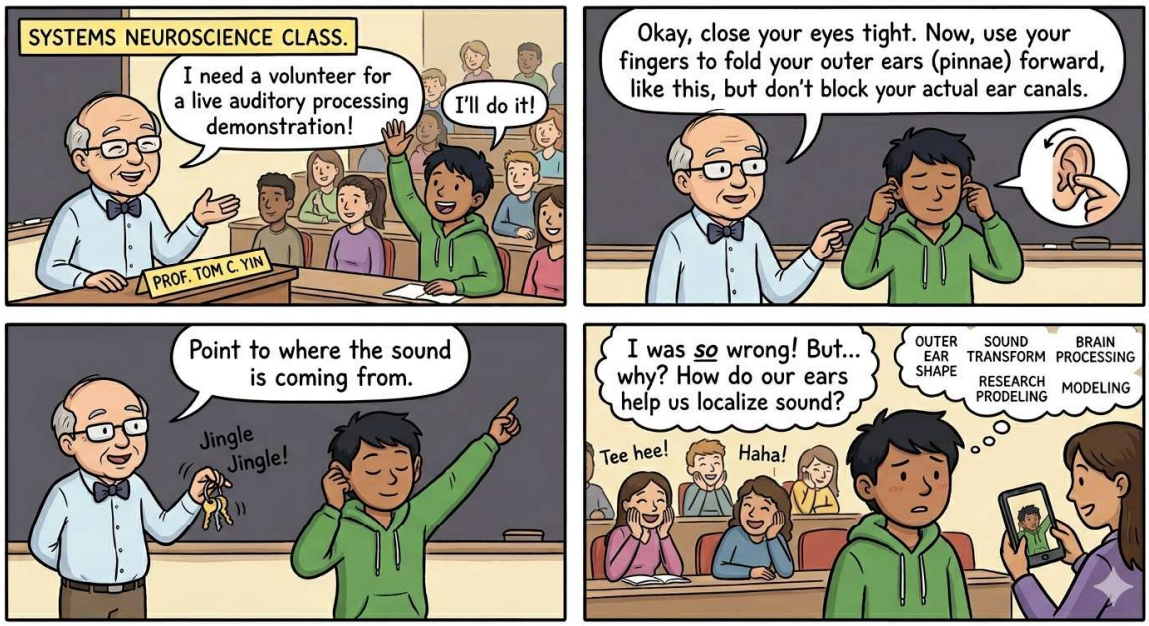

Figure 1.1: Cartoon for Key Jingle Classroom Example. A cartoon generated by Gemini to depict the classroom example that inspired my thesis journey during my Ph.D. Panel 1: The professor asks for a volunteer from the class. Panel 2: I stand at the front of a classroom with eyes closed. I use my hands to fold my outer ears forward (pinnae bent) while leaving the ear canal open. Panel 3: The professor jingles the keys and I confidently point high up toward the ceiling. Panel 4: My classmates in the background chuckle and the location I'm pointing is incorrect.

Professor Yin walked to the front of the classroom and asked for a volunteer for a live demonstration. I raised my hand. He called me up, told me to close my eyes, and instructed me to fold my outer ears forward using my fingers, making sure not to cover my actual ear canals.

With my eyes firmly shut and my ears pinned forward, I heard a crisp metallic sound: the jingling of Professor Yin's keys.

"Point to where the sound is coming from," he instructed.

I confidently pointed up and to the right. A wave of giggles rippled through the lecture hall. He jingled them again. I pointed down to the left. More laughter. When I opened my eyes and my friend showed me a video of the demonstration, I realized my guesses had been accurate horizontally, but wildly inaccurate vertically. The keys had been jingled at different heights and angles than where I was pointing.

It felt like a simple party trick, but it triggered a cascade of questions that defined the next several years of my life: How do our outer ears transform sound waves to tell our brains where things are? How is the brain organized to use that information? Can I build a computer model that mimics this biological superpower to clean up noisy audio? And if I do, how do I design current and future computer systems to run these brain-like models efficiently?

This chapter tells the story of how that key-jingling demonstration led to a full-stack engineering solution spanning biology, artificial intelligence, supercomputers, and microchip design.

1.4 Teaching Computers to Hear Through the Ear's Shape

To understand why I failed Professor Yin's test, we have to look at the anatomy of the ear. The cartilage-filled part of your ear on the outside of your head is called the **pinna**. The pinna is not just a passive funnel; it is a sophisticated acoustic filter. Its unique ridges, bumps, and valleys shape incoming sound waves.

Depending on whether a sound comes from above, below, behind, or in front of you, the sound waves bounce off these physical structures differently, as shown in Figure 1.2. This bouncing creates tiny, localized echoes and alters the pitch frequencies of the sound before it ever reaches your eardrum. Your brain learns these subtle changes over a lifetime, allowing you to determine a sound's location, especially its height (elevation), in three-dimensional space. When I folded my ears forward, I flattened those ridges, stripping away the spatial cues my brain relied on.

The pinna also helps us isolate important sounds in a crowded room,

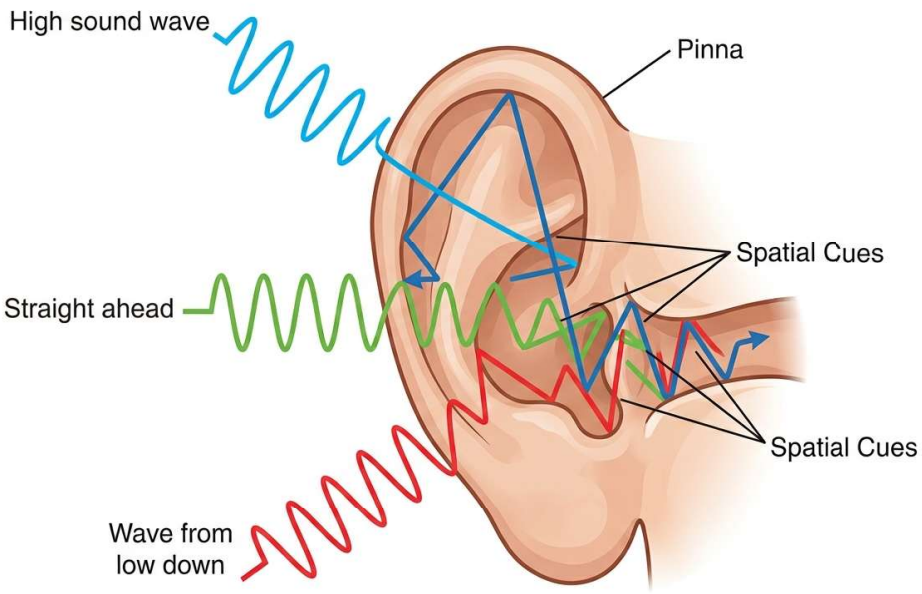

Figure 1.2: Audio Reflections off the Pinna. Sound waves bouncing off the ridges of the outer ear (pinna) create elevation cues that help the brain locate sounds.

an ability scientists call the "cocktail party effect." In a noisy environment, your brain uses these directional filters to zone in on a speaker's voice while tuning out background chatter.

Isolating speech from background noise is called **speech denoising**. Traditional artificial intelligence (AI) models can clean up audio incredibly well, but they require massive amounts of power. Think of standard AI models like industrial lightbulbs that stay turned on at maximum brightness all day, drawing huge amounts of electricity. The human brain, on the other hand, runs on roughly the same amount of power as a dim refrigerator lightbulb (about 20 watts).

The brain achieves this efficiency because of how its cells, or neurons, communicate. Instead of constantly exchanging streams of complex numbers, biological neurons remain quiet until they have something important to say. When they do, they emit a tiny, rapid burst of electricity called a **spike**.

Inspired by this, my first major research project was to build a **Spiking Neural Network (SNN)**, a brain-like computer program designed to mimic these electrical spikes. To make this network lightweight, I programmed it to use the mathematical equivalents of the "pinna transforms" observed in Professor Yin's classroom. These mathematical equivalents were procured from open-source, prior works that compactly captured the effects of generic sounds incident to a pinna. I used the data from these files to transform audio to represent speech originating from one location, and noise from another, personalized by a specific subject's ear properties.

By encoding noisy audio with the same spatial cues that our outer ears naturally provide, my tiny brain-like model was able to successfully isolate and clean up human speech. The model was able to detect the latent properties in the audio that revealed the audio source locations, made disentangling the two tracks easier. By copying the physical shape of the human ear, I showed that it was possible to make smarter, more compact software tools to succeed in the speech denoising task.

1.5 Misalignment of Modern Cluster Computing and Biological Processing

Developing a green, brain-like program on paper is one thing; actually training it is another. To teach my software how to distinguish speech from noise, I had to feed it hundreds of hours of audio examples. This required the muscle of modern supercomputers, specifically clusters of **GPUs** (Graphics Processing Units), which are the heavy-duty microchips that power modern AI.

However, I quickly hit a major bottleneck. Modern supercomputers are built like highly rigid factories. They expect data to arrive in perfectly uniform, dense blocks. But my biological, spike-based data was completely different: it consisted of long stretches of silence punctuated by unpredictable, sparse bursts of activity.

To a traditional supercomputer, processing these long, sparse spike vectors is inefficient. It is like trying to pack circular boxes into a delivery truck built exclusively for square crates, as illustrated in Figure 1.3. Because the supercomputer's hardware does not speak the language of spikes natively, the system spends a massive amount of time shuffling data around, translating formats, and waiting idle.

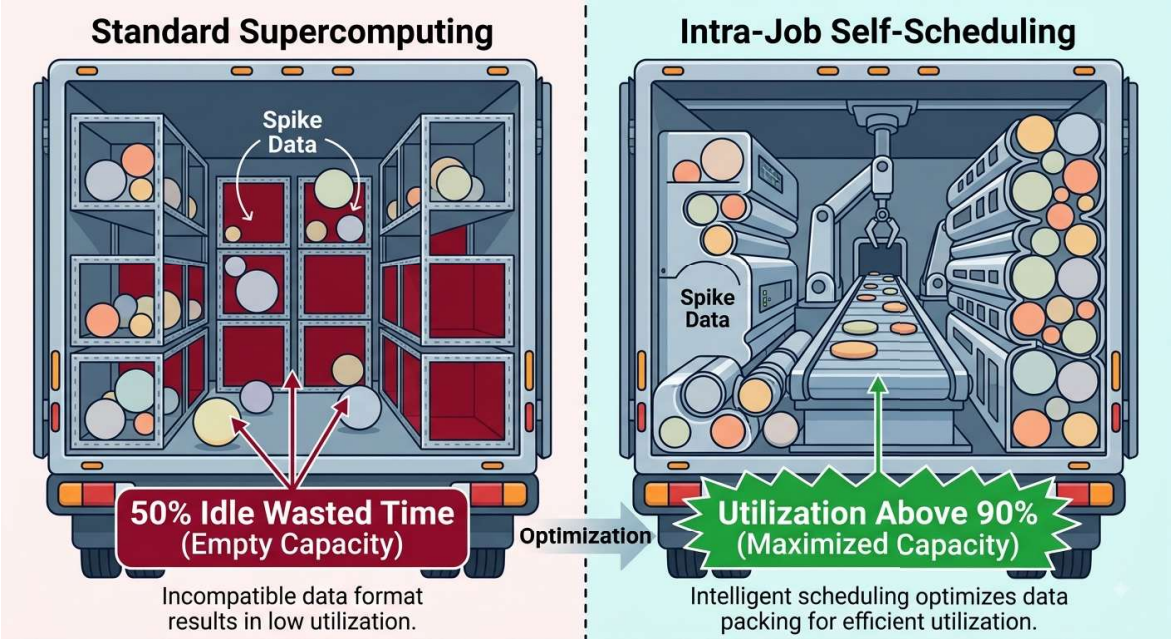

Figure 1.3: Mapping Spiking Neural Network Training to Standard Compute Clusters. Sparse biological spike data creates empty space in standard supercomputer trucks, which intelligent scheduling reorganizes. Left panel: A delivery truck filled with rigid square shelves trying to hold loose, circular objects labeled "Spike Data," leaving huge empty gaps. Right panel: A smart conveyor belt system neatly repacking the circular spike objects into tight, organized rows inside the same truck, filling the space efficiently.

When I measured how my brain-like networks trained on standard supercomputer clusters, I discovered a major inefficiency: more than half of the requested cluster compute time was not being utilized. The chips were sitting idle for more than half of their job slots, delaying the collection of results and wasting precious computing slots.

To fix this supercomputer traffic jam, I shifted my focus to the system orchestration layer, which is the digital coordinator that schedules and manages how jobs run on a supercomputer. I observed three sources of inefficiencies in

the training deployment pipeline, and then further designed three solutions to overcome them:

1. - **Insight:** The progress of a training run depends on the dynamic, runtime condition of the machine in the cluster. This can vary from a variety of things like: temperature in the room that day, number of other users trying to use the machine, physical location from the dataset and machine, etc.
   - **My Solution:** *Intra-job self-scheduling:* A smart system that monitors how fast a training job is moving in real time and dynamically adjusts the workload so it finishes exactly when the time slot expires. The monitor tracks training progressions, and uses the latencies of the past few training epochs to predict how many more training epochs can be completed before time runs out.

2. - **Insight:** Not all machines train at the same rate. Some machines are newer, some are older, some are bigger, some are more powerful, some have more memory. The configuration that works best for one machine may not be optimal for another.
   - **My Solution:** *Machine-specific tuning:* Automated adjustments that configure my brain-like networks to perfectly match the unique

hardware characteristics of the specific supercomputer machine they are running on.

3.
   - **Insight:** All jobs operate on the same dataset. So if two jobs get allocated onto the same machine, it's a waste to keep two different copies of the same dataset on the machine.
   - **My Solution:** *Inter-job data transfer protocols:* A method that allows data to be handed off smoothly between consecutive tasks, minimizing the time chips spend sitting empty.

By introducing this digital traffic coordinator, I slashed the wasted time, boosting supercomputer utilization to over 90%. I proved that even if current hardware is not built for brain-like algorithms, intelligent software coordination can bridge the gap.

1.6 Building Future Hardware for Randomness

While fixing supercomputer schedules helped in the short term, the ultimate goal of neuromorphic computing is to build new kinds of physical microchips designed from the ground up to think like the brain.

As I pushed further down the computing stack into the microarchitecture layer, meaning the actual physical layout of silicon chips, I ran into a

fundamental property of biological systems: **randomness**.

Human brains and advanced neuromorphic networks thrive on stochasticity, which is a scientific word for unpredictability or randomness. Our neurons fire with a degree of healthy randomness, which helps us learn, adapt to new environments, and avoid getting stuck in mental ruts. Simulating the level of randomness seen in the brain on a modern computer chip is a challenge.

Computers struggle to simulate this as they are constructed to be deterministic. The value of a computer is that it's able to execute a sequence of instruction to produce very predictable results. Thus, when trying to simulate what happens in the brain, computer chips really struggle with randomness. Modern computer processors are built for strict predictability. To achieve fast speeds, they look at the code you are running and try to guess what choice or action is coming next. If the program is predictable, the chip flies through the task. But when a program relies on random values, the chip's guessing engine falls apart. It guesses wrong, encounters what scientists call a "misprediction penalty," and is forced to throw away its current work, clear its memory pipes, and start all over again. Randomness cripples standard computing performance.

To conquer this, I designed **StAccato**, a hardware-software tag team

built to take the penalty out of randomness. StAccato works through two cooperative innovations:

- **The Hardware Component:** I designed a dedicated, ultra-fast Hardware Random Number Generator (HWRNG) directly into the chip architecture. Instead of forcing the main computer processor to stop what it is doing and run a slow piece of software to calculate a random number, my custom hardware component produces a readily available stream of pseudo-random numbers whenever a program needs them. The hardware sits right next to the main processor, and feeds values directly to it when requested.
- **The Software Component:** Think of this like a digital assistant running slightly ahead of the main program. The assistant looks ahead, sees where random calculations or random memory lookups are going to happen, handles them in advance, and clears all the obstacles out of the way. By the time the main program reaches a random decision, the path is already perfectly laid out. There is no guessing, no stumbling, and no wasted time.

I tested StAccato against standard benchmarks that had varied widely in the degree to which they required random values. These benchmarks

included Monte Carlo simulations for digital option pricing (essentially simulating market changes to determine whether to make trades), artificial intelligence game solves (some games are too complex for a program to solve completely, so they may rely on heuristics or probabilities to determine optimal moves), and even in the chip design process itself (determining which design component of computer chip gets printed to a specific location on a wafer is an intractable problem for modern computers, so other heuristics and stochastic approaches are used to estimate the best placement strategies). StAccato delivered an immediate **1.34x speedup** from the benefits of the hardware generator alone. When I turned on the digital assistant software thread, that speed jumped to a **1.49x speedup**, capturing 97.9% of the maximum possible speed achievable if randomness had zero cost.

1.7 Connecting the Dots and Future Horizons

What started as an embarrassing moment in Professor Yin's neuroscience class, where I mispointed at a set of jingling keys, unlocked an inspiring line of research. Nature has spent millions of years engineering elegant, ultra-efficient shortcuts for processing the messy, noisy world around us.

By looking closely at the human ear, I did not just learn how humans hear; I learned how to build a whole new paradigm of technology. I proved

that:

1. **At the application layer,** copying the physical acoustics of the outer ear allows us to build ultra-lightweight, brain-like software that cleans up speech using minimal power.
2. **At the system layer,** we can reorganize how computing clusters operate so that they can handle these biological, sparse networks without wasting massive amounts of time and energy.
3. **At the hardware layer,** we can design entirely new chip architectures like StAccato that embrace biological randomness instead of running away from it.

True innovation cannot happen in an isolated silo. To unlock the future of energy-efficient, brain-like artificial intelligence, we must build a bridge that spans all the way from the biological quirks of human anatomy, through the orchestration of supercomputers, down to the very silicon chips that power our world.

Following my Ph.D. defense in 2026, I intend to apply these interdisciplinary principles to advance sustainable computing systems in industry and research. As artificial intelligence models grow exponentially in size, the energy required to power them is becoming a critical global challenge.

Translating the energy-saving tricks of biological brains into next-generation hardware will be essential for building a greener technological future.