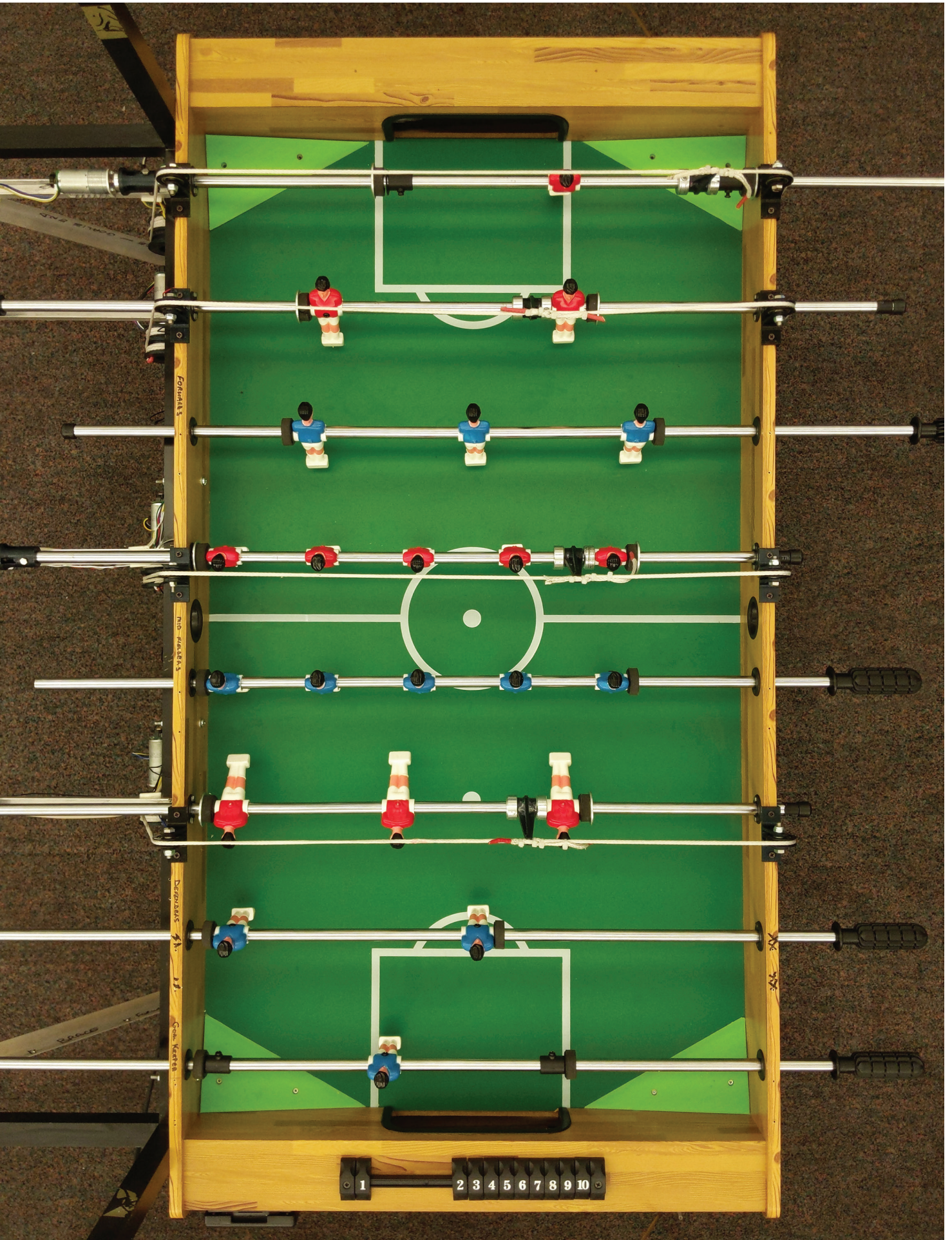


This robot foosball table is meant to serve as a benchmark test for neuromorphic algorithms and other technologies.

GOOAAALL!!!

WHY WE BUILT A NEUROMORPHIC ROBOT TO PLAY FOOSBALL

BY GREGORY
COHEN





FOR THE PAST 25 YEARS or so, those of us who seek to mimic the brain's workings in silicon have held an annual workshop in the mountain town of Telluride, Colo. During those summer weeks, you can often find the participants unwinding at the bar of the New Sheridan Hotel on the town's main street. As far back as most can remember, there has been a foosball table in the bar's back room. During the weeks of the workshop, you'll usually find it surrounded by a cluster of neuromorphic engineers engaged in a friendly rivalry that has spanned many years. It was therefore almost a foregone conclusion that someone was going to build a neuromorphic-robot foosball table.

That someone was me.

It turns out there's more to the idea than simple fun. After all, why do we play competitive games like foosball? We are drawn to them for social reasons, but we also enjoy learning the mechanics and improving our performance. Games are how we boost our hand-eye coordination, tracking and predictive abilities, and strategic thinking. Those are all skills we want robots to have.

Humans have always been fascinated by the idea of machines playing our games. As far back as the late 18th century, the Mechanical Turk hoax enthralled and amazed audiences with its (fictitious) ability to beat humans at chess. But we were all just as amazed in 1997 when IBM's Deep Blue did it for real. Now, such triumphs are almost a regular occurrence, with DeepMind's AI systems first defeating a human champion at the board game Go, and then going on to victory with the video game *StarCraft II*. (An AI will probably have conquered another of your favorite games by the time you finish reading this.)

These feats of computing are pretty good measures of a system's abilities. But they fall short in some important ways. Robots need to operate in a real world full of noise, irregularities, and evolving environments. The rigid rules and constrained environment of Go will never

provide such challenges. Real-world games, certainly foosball and possibly pinball, might be a better measure of whether our efforts to match the might of the human brain are really on track.

Why are we so interested in learning biology's computational and sensing secrets? Frankly, it's because they are so superior to today's computing technology, which seems to be fast reaching its limits. Commodity sensors produce too much data for computers to understand, and those computers consume too much power trying to make sense of it. Biology outperforms all our technologies when it comes to sensing and perceiving the world, and biological organisms are orders of magnitude more power efficient, reliable, robust, and adaptable.

My colleague at Western Sydney University's International Centre for Neuromorphic Systems (ICNS), André van Schaik, gives a great example: the humble mosquito. Its brain is composed of only about 200,000 neurons, yet its flight control and obstacle avoidance are far superior to anything that we have built. Next, consider the dragonfly, which can capture a mosquito midflight. It has about five times as many neurons as the mosquito and consumes perhaps 30 mosquitoes' worth of energy per day,



about equivalent to a few grains of sugar.

One of the most straightforward examples of how neuromorphic technology can be used in sensing is vision, which happens to be my speciality. When it comes to building devices that need to see the world, cameras with CMOS imagers are almost always used. These cameras are such a commodity that it's easy to forget that a picture (which computer-vision researchers call a frame) is not the only way to perceive the visual world.

Cameras are built to capture a representation of a scene that's good enough to fool our visual system. We don't really know what features or information the visual system is using to understand the scene, so cameras simply capture as much information as possible. That approach is fine for taking static pictures, but it's not a great fit for doing things like tracking objects through space. For example, imagine trying to track an object—a foosball ball, for instance—that's moving so fast it completely leaves the edge of the image in the 33 milliseconds between one frame and another. Sure, you could use a camera with double the frame rate, but that means you've now got twice as much data to sort through just to keep track of that one object.

Biological eyes work differently. There are no frames in biology, and there are too

few nerves going between the eyes and the brain to transmit whole images anyway. Neuromorphic vision sensors draw inspiration from how the eye's photoreceptors work; they still use lenses to project the world onto a grid of pixels on a silicon chip, but it's what those pixels do with the information that's interesting.

The pixels in neuromorphic sensors—also called event-based imagers—report only changes in illumination and only in the instant when the changes happen. They don't produce any data when nothing is changing in front of them. This approach drastically reduces the amount of data these cameras generate, which means less data to store, to transmit, and to process. These imagers therefore use less power both in the camera itself and for all the computation that needs to happen afterward.

Startups Prophesee and IniVation already have brands of event-based imagers on the market. And these sensors have even gone to space: Neuromorphic cameras from ICNS will spot satellites and space junk from orbit, and a different sensor was recently installed on the International Space Station to examine ephemeral atmospheric phenomena, such as sprites.

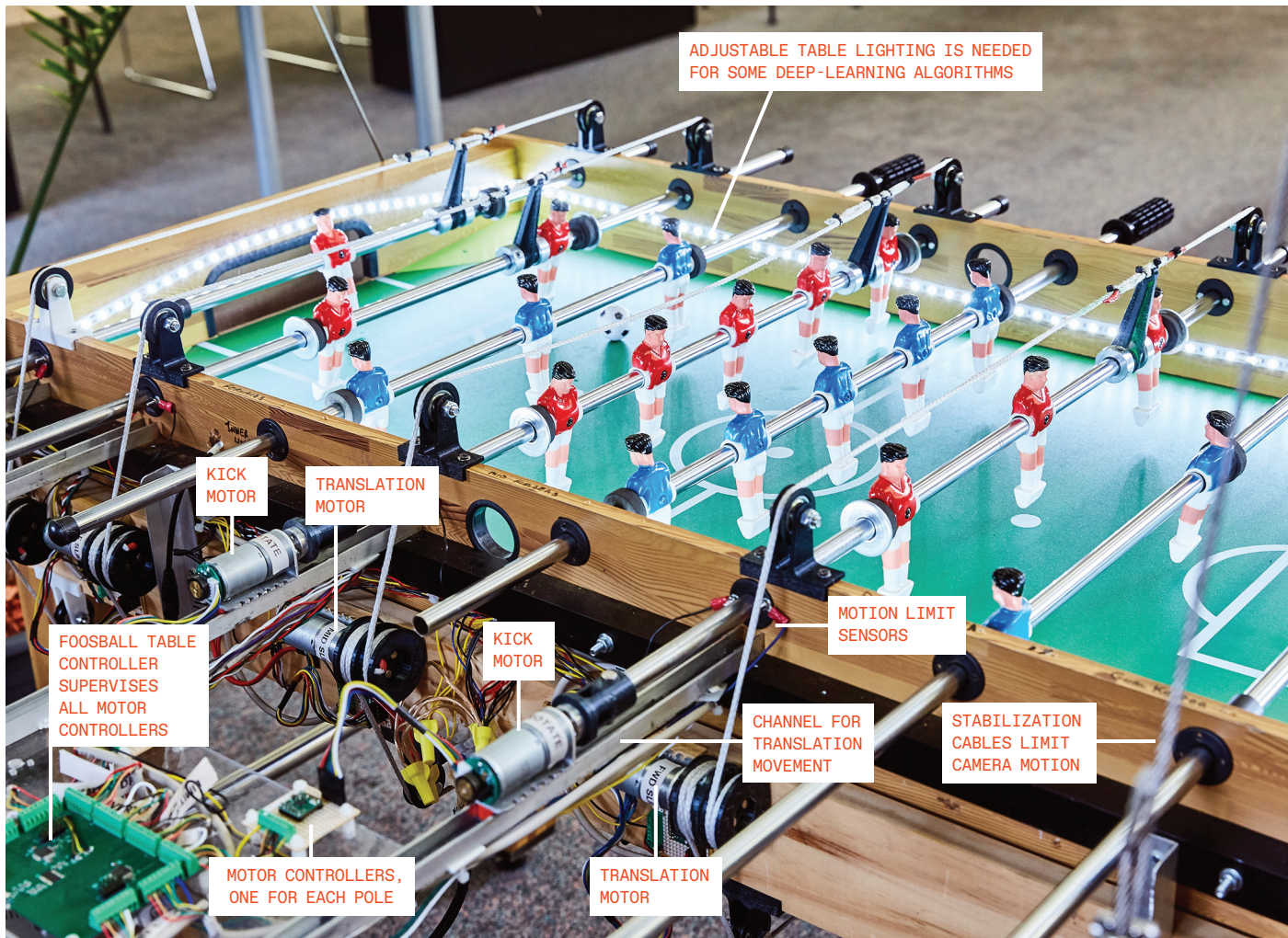
Neuromorphic researchers have also tackled our other senses. They've developed silicon cochleas to model hearing,

Left: Neuromorphic-robot foosball was born from neuromorphic engineers playing at the New Sheridan Hotel in Telluride.

Right: In its two trips to the Telluride Neuromorphic Cognition Engineering Workshop, our robot foosball table has seen plenty of use.

tactomorphic sensors to explore touch, and even a silicon nose to identify odors and gases. Beyond sensing, neuromorphic engineering seeks to understand the fundamental ways in which brains process and store information. In fact, the origins of neuromorphic engineering lie in trying to build electronic neurons to better understand how real neurons in the brain operate.

Neuromorphic sensors, and the brain-inspired algorithms that work with the data they produce, allow for specialized systems built specifically for efficient performance on certain tasks. However, it can be difficult to know when these sensors are capturing the right information or when our algorithms are working properly. That's where the need for benchmarks comes into play.



We designed the neuromorphic-robot foosball table to be easy to assemble, so other groups could make their own. Motors at the end of the rods rotate the players for kicking. A separate set of motors and pulleys slide the players across the table. Controllers orchestrate the motion according to the output of a neuromorphic algorithm.

To help understand why we need foosball as a neuromorphic benchmark, take the example of how an event-based imager would handle a benchmark that today's deep-learning AIs deal with all the time, the MNIST database. MNIST (short for Modified National Institute of Standards and Technology) is like the "Hello, World!" of machine vision. Its data set of thousands of low-resolution images of handwritten numerals offers a baseline for how well an image-recognition neural network is working.

An event-based imager would momentarily see each MNIST numeral as it flashes in front of it. For such a sensor to continue to see the static numerals, either the camera must move

or the digit must, and in a controlled way. Eyes do something similar: Their focus moves from point to point until the brain understands what it's seeing.

Creating data sets like MNIST that are a suitable test for neuromorphic systems is not trivial, and the truth is they're not very useful. The process of linking motion to imaging can be so dynamic that for anything but the most constrained tasks, the number of possibilities would be quite large. So how can we determine if neuromorphic systems are working, and how can we compare them to other approaches?

There are, of course, benchmarks that are interactive simulations. For example, in autonomous driving simulations, the

view fed to the algorithm from the car's sensors changes as the car's position changes. But these simulations have their problems. The most significant is the contrast between controlling a simulation and controlling a physical system.

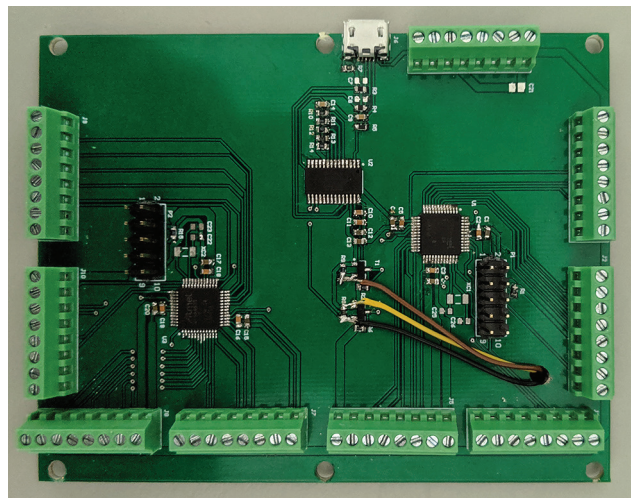
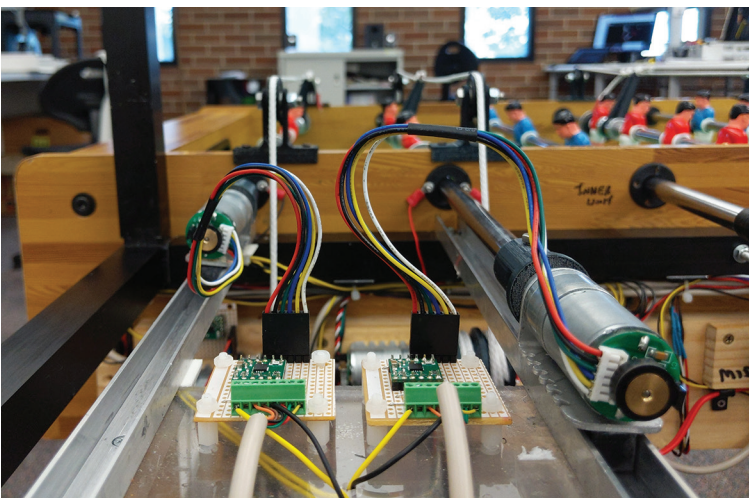
The major difference between simulated systems and reality lies in the amount and nature of noise in the real world. For most AI systems, noisy data is a big problem. But there's reason to believe that neuromorphic systems thrive with noise, and perhaps even need it. That's not as strange as it might seem. Our own sense of movement and body position is actually enhanced by a certain amount of noise. Attempts to mitigate noise in neuromorphic systems, either through extra pro-



Left: Like biological eyes, event-based cameras register only the changes in a scene. This greatly reduces the data needed for tracking the ball.

Below left: Each motor has its own controller.

Below right: The table control system supervises the motor controllers and performs other duties.



cessing or by designing real-world systems that are closer to our idealized simulations, may have held us back.

So what we need to move neuromorphic systems forward are benchmarks that are physically embedded in the real world.

Let's start with something simple: pinball. It's actually a very good choice for a benchmarking problem because the game is so straightforward. There are only two outputs, one for each flipper, and the game largely revolves around timing. The realities of the physical system are unforgiving, and you cannot simply pause or slow the movement of

the ball to allow an algorithm to catch up. Most important, there's a score in pinball and a clear objective to maximize that score. So whichever system gets the highest score at pinball is unequivocally the better robotic pinball algorithm.

We can also make the problem more difficult by tweaking the game a little. For example, we can add multiple balls at the same time, or even decoy balls or balls of a material that will behave differently on the pinball table. This allows us to include a wider range of tasks such as tracking, detecting, segmenting, and identifying the balls while still maintaining the score as the ultimate metric for success.

Our approach to building a robotic foosball table aims to re-create the same inputs as those experienced by a human player.

ICNS has built a demo using a robotic pinball machine that can keep three balls on the table with about the same effectiveness as a human player. Amazingly, unlike the hundreds of thousands or millions of artificial neurons found in common deep-learning-based systems, this tiny neuromorphic brain interprets and acts on the input from an event-based imager using just two artificial neurons.

Pinball is great, but my team felt there was a need for a more complicated and demanding task to further push the neuromorphic research community. Also, we like playing foosball at the New Sheridan Hotel's bar.

Foosball looks like an easy game for robots: All the action happens in two dimensions, and it takes only eight motors to control all the little figures on the table. But it's much more difficult than it seems.

There have been a few attempts over the years at building a robotic foosball table with varying degrees of success, but none using neuromorphic sensors and algorithms. The prior robotic systems often needed to modify the game to give the robot an advantage. For example, the foosball table built by Brigham Young University made use of a color-segmented tracking algorithm and required that the ball be the only green object on the table. The robotic foosball table at École Polytechnique Fédérale de Lausanne (EPFL), in Switzerland, is impressive, but it simplifies the task dramatically by replacing the bottom of the foosball table with a transparent plastic sheet and letting the camera look up, thereby always providing an unobstructed view of the ball.

Our approach aims to re-create the same inputs as those experienced by a human player. The camera looks down on the table, giving it an obstructed view that's similar to what a human would have. And we use a regulation ball, not one with special markings or colors.

Our robotic foosball table has so far made two trips from Australia to the mountains of Colorado. For three weeks at a time, teams of fresh neuromorphic engineers have descended upon the problem with gusto, taking up the challenge of programming the table to achieve the highest score. The results highlighted the difficulties of the task and the shortfalls of conventional AI methods.

For one thing, tracking the ball with a neuromorphic sensor should be easy, and in the trivial case of the pinball machine, it clearly is. However, foosball is a more dynamic game, especially when a human player is involved. Human players each have different strategies, and their movements are not always logical or even necessary.

Attempts to use non-neuromorphic solutions, such as deep learning, led to a few interesting lessons. First, it became apparent that the way deep-learning neural networks are processed—usually on a GPU—is not well suited to this type of task. GPUs operate best on batches of images rather than a single frame at a time. This is a problem, because we don't care about where the ball was in the past, and we don't really even care about where the ball is at the moment; what we really care about is where it's going to be next. So the deep-learning solutions were processing



Pinball is a simple test of neuromorphic systems. It's so simple, in fact, that we built a two-artificial-neuron system that could keep three balls in play at once.

a lot of unnecessary information.

Second, we found that the deep-learning methods were extremely sensitive to small variations in the problem. A slight shake of the camera, a bit of skew in the table from players pulling it in different directions, or even a shift in lighting conditions caused the elegant performance of deep-learning ball trackers to break down. It's likely that we could increase the amount of training to handle all these small deviations—there's a whole field of research devoted to building networks that are resilient to these sorts of things—but that would take many, many more games.

Our latest approaches look toward simpler and faster neuromorphic networks. These algorithms process every event—also called a “spike” in neuromorphic computing—from the camera and use them to update the estimation of the position of the ball.

Instead of deep learning's vast layers of neurons, these networks use 16 small pattern-recognition networks of 18 x 18 pixels each, so only 364 pixels are being considered at any point in the game. This makes them very fast and mostly accurate. And being fast is critically important, because event-driven algorithms need to keep up with the time-sensitive data being produced by the camera. Each event

necessitates no more than some small and simple calculations. While this system doesn't pose much of a threat to an experienced player, our network's tracking has improved to the point where it can quite reliably block the ball. Goal scoring, however, is still a work in progress.

Deep learning could perform a similar operation, in principle, but it needs to look at the entire image, and it performs orders of magnitude more calculations for each layer of the network. Not only is this far more data than our system uses, but it also effectively turns the event-driven output back into frames.

Currently, our algorithm is trained offline from recorded event-based data. It uses a genetic algorithm—one that evolves toward an optimal solution—to both learn what the ball looks like and to create good estimates of where it will be next. The algorithm learns how to recognize the ball from the data itself, rather than through any coding on our part. It also learns from how the ball really moves, rather than our own expectation of it. These are both important points, as our preconceptions of a good model for the ball turned out to be very far from those that work well. We also found that our simulations and expectations for the motion of the ball were wildly off.

Our next step is to move our learning from offline training to real-time online learning, allowing the network to continuously learn and adapt while the game is in progress. Among other things, that may help with a sensitivity the system has now to the specific table it's trained on.

These event-driven algorithms are an intermediate step toward algorithms designed to work using so-called spike-based neuromorphic hardware. These brain-inspired processors, such as Intel's Loihi and BrainChip's Akida, encode information as the timing of spikes and are a natural fit with event-based sensors. Once we have stable spike-based algorithms, we'll be able to make improvements more quickly.

Hopefully, we won't be the only ones making these improvements. In designing the robot foosball table, we focused on keeping the costs down and made the entire project open source. With luck, other neuromorphic research groups will see enough value in having their own robot benchmarks. And if not, they'll be able to find us and our foosball table in Telluride later this year. ■