

Emulating insect brains for neuromorphic navigation

Korbinian Schreiber^{1,3}, Timo Wunderlich¹, Philipp Spilger¹, Sebastian Billaudelle¹, Benjamin Cramer¹, Yannik Stradmann¹, Christian Pehle^{1,4}, Eric Müller¹, Mihai A. Petrovici², Johannes Schemmel^{§,1}, and Karlheinz Meier^{§,1,†}

[§]Shared senior authorship

[†]Deceased

¹Kirchhoff-Institute for Physics, Heidelberg University

²Department of Physiology, University of Bern

³Institut für Innovation und Technik (iit), Berlin

⁴Cold Spring Harbor Laboratory

January 2, 2024

Bees display the remarkable ability to return home in a straight line after meandering excursions to their environment. Neurobiological imaging studies have revealed that this capability emerges from a path integration mechanism implemented within the insect’s brain. In the present work, we emulate this neural network on the neuromorphic mixed-signal processor BrainScaleS-2 to guide bees, virtually embodied on a digital co-processor, back to their home location after randomly exploring their environment. To realize the underlying neural integrators, we introduce single-neuron spike-based short-term memory cells with axo-axonic synapses. All entities, including environment, sensory organs, brain, actuators, and the virtual body, run autonomously on a single BrainScaleS-2 microchip. The functioning network is fine-tuned for better precision and reliability through an evolution strategy. As BrainScaleS-2 emulates neural processes 1000 times faster than biology, 4800 consecutive bee journeys distributed over 320 generations occur within only half an hour on a single neuromorphic core.

path integration, navigation, central complex, bee, short-term memory, BrainScaleS, mixed-signal, neuromorphic, spiking, neural networks

1 Introduction

Insects are capable of navigating complex real-world environments with remarkable efficiency, speed and robustness. Besides the remarkable adaptation of their bodies for this task this is due to their neural anatomy which evolved over a period of at least 520 million years [1]. Being capable of mastering numerous difficult tasks such as homing [2], migrating over distances of more than 1000 kilometers [3, 4], remembering and communicating spatial locations [5], or learning visual cues [6], even large insect brains measure

only a few cubic millimeters in volume and often consist of less than a million neurons [7, 8].

Within the last decade, research in experimental neuroscience has brought about a completely new quality in the measurement of neuronal structures and processes, providing groundbreaking insights into their working principles. Advances in biological imaging and data processing resulted in almost complete connectomes of insect brains [9–12], while methodological combinations of genetic modification, bio-imaging, and virtual reality for fixated animals made it possible to record real-time cell-level neural activity of brains engaged in realistic interactions with the environment [13, 14]. Whereas earlier lesion-based studies have shown that insect brain structures like the *central complex* [15] are somehow involved in orientation and navigation [16], modern calcium-based imaging techniques [17] provide detailed time-resolved insight. For example, imaging the head-fixed fruit fly *Drosophila melanogaster in vivo* in virtual environments has revealed that subcomponents of the *central complex* track azimuthal visual cues [18] and indicate and integrate the fly’s heading direction [19]. Similar experiments on the sweat bee *Megalopta genalis* have even revealed enough insight to construct a fully functional physiological model for path integration-based navigation [13], deciphering a mechanism that lets the insect keep track of its home location by integrating over its current heading velocities.

While biological experiments continue to unveil intricate physiological details of these tiny nervous systems, studying such abstract neural network models in simulations can confirm and sharpen our understanding of how nervous structures correspond to their respective functions [13, 20, 21]. As those functions are often defined by interactions of the entire animal with its environment, faithful and valid simulations need to encompass also components beyond the neurons. This might include the environment, the animal’s body, its sensory and motoric organs, or the information de- and encoders translating between external physics and nervous signals. All those com-

ponents can be implemented virtually in simulations [22–25] or physically through robots [26, 27].

Similarly, the dynamics of the attached neural networks can be solved numerically within simulations [28–31], or emulated using physical, neuromorphic circuits [32–39]. Those circuits are usually densely integrated on CMOS-based neuromorphic microchips and mimic the architecture and properties of biological neural networks. Unlike common central processing units (CPUs), which typically operate on data within a single large block of memory, neuromorphic processors tend to perform many parallel computations using multiple processing units and distributed memory. While some implementations are based on specialized digital building blocks [40–45], others recreate the dynamics of physiological membrane potentials in terms of analog voltages and currents, thus, emulating physical models of the biological archetypes [46–52]. Also in the field of analog neuromorphic engineering, rapid advances and developments have taken place especially within the last decade, revealing unprecedented versatility at outstanding power efficiencies [33, 53–60].

In the present work we merge recent seminal discoveries in experimental neuroscience with a number of recent innovations in neuromorphic engineering into a cohesive neurorobotic agent. Relying on an accelerated analog neuromorphic nervous system, we reproduce the ability of bees to return to their nest’s location after excursions through a two-dimensional environment. While previous case studies have successfully mapped various subcomponents of the networks used by insects or mammals for path integration onto neuromorphic substrates [32, 35, 36, 61, 62], we demonstrate a fully embodied and autonomous neuromorphic agent within a single BrainScaleS-2 neuromorphic prototype chip [51, 54, 57, 58, 63]. The agent’s body and the environment are simulated on a digital on-chip coprocessor, while the neural network is physically emulated on the chip’s analog neuromorphic core with sensory inputs and motor outputs connecting the two domains via mixed-signal circuits. Due to the intrinsic time scales of the analog circuits, the neural dynamics proceed $1000\times$ faster than in biology. This speed-up is independent of the network size and allows for rapid prototyping during the initial design process, when architectural decisions are still to be made or when parallelizing stochastic parameter evaluations is not feasible. The acceleration also benefits optimization algorithms which improve the network’s performance by successive incremental parameter updates [33, 51, 53–56, 59, 60]. We demonstrate this by employing an evolution strategy algorithm for fine-tuning the agent’s performance through synaptic weight updates. As an example, spawning 320 successive generations of 15 neuromorphic insects and evaluating the navigation capabilities of each individual during 3.3 min flights takes only around 0.5 h on a single neuromorphic processor without parallelization. Executed in real time, the flight simulations alone would take more than 11 days.

Rendering the foundational neural network model derived from *Megalopta genalis* [13] compatible with the neuromorphic circuits on BrainScaleS-2 additionally required some essential adaptations and adjustments. This in-

cluded, most importantly, the transition from abstract, analytical firing rate neurons to spike-based leaky integrate-and-fire (LIF) neurons. Besides common firing rate models which map summed input activities through a nonlinear response function to an output activity, the original model crucially relies on heuristically defined equations responsible for storing and integrating the activities of presynaptic neurons. To biologically plausibly implement a functionally identical mechanism based on spiking LIF neurons, we introduce a single-cell short-term memory mechanism based on axo-axonic synapses. While the resulting model has been developed for BrainScaleS-2, it is flexible enough to be ported to other neuromorphic systems as well. For example, sub-threshold neuromorphic platforms, like ROLLS [47] or DYNAPs [48], could adapt the network to facilitate ultra-low-power robotic control units. Other applications requiring more robustness and repeatability could instead make use of digital neuromorphic systems like SpiNNaker [40, 64], Intel’s Loihi [42, 45] or IBM’s TrueNorth [41]. Being compatible with a wide range of spiking neural models, the presented model can also serve as a building block in larger insect-inspired systems or in neuro-inspired control units.

2 Results

2.1 BrainScaleS-2

BrainScaleS-2 is a mixed-signal neuromorphic computing system [51, 63]. The second-generation prototype chip used in this work contains an analog neuromorphic core with 32 physical neurons [63] with 32 synapses each.¹ The neuron dynamics follow the LIF equation $C_m \dot{V}_m = -g_l(V_m - V_l) + I_{\text{syn}}$, where C_m is the membrane capacitance, V_m the dynamic membrane voltage, g_l the leak conductance, V_l the static leak or resting potential, and I_{syn} the dynamic synaptic input current. Whenever the membrane potential crosses the threshold voltage V_{th} , it is reset to V_{reset} and the neuron’s spike counter is incremented. The post-synaptic spike is represented as a digital data package that carries the unique address of the firing neuron. It is sent off-chip to an field-programmable gate array (FPGA) responsible for recurrent spike routing (later versions of the platform include an on-chip routing engine) and turned into a pre-synaptic spike with a newly assigned target label.

Pre-synaptic spikes arrive at the mixed-signal synapse circuits where their target label is compared against the programmable synaptic label. If the labels match, the circuit triggers a synaptic event (i.e., a current pulse) with an amplitude proportional to a 6 bit programmable synaptic weight w . The resulting current pulse creates a post-synaptic potential on the dendritic capacitance which is finally transferred to the membrane capacitance of the receiving neuron. Synapses can be either excitatory or inhibitory, resulting in a positive or negative postsynaptic potential, respectively.

¹The full-scale BrainScaleS-2 chip has 512 neurons with 256 synapses each.

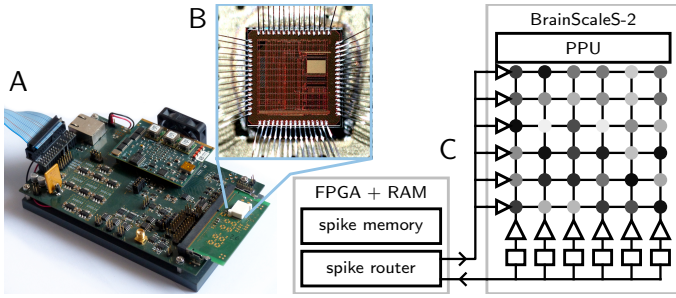


Figure 1: BrainScaleS-2 prototype system. A) Entire system with FPGA, BrainScaleS-2 chip and periphery. B) BrainScaleS-2 prototype chip of the second generation. C) System schematic depicting the FPGA and the chip. Events coming from the FPGA are distributed to the synapse rows. The synaptic columns are connected to the triangular-shaped neurons that send spikes to the spike counters at the bottom and then back to the FPGA.

The prototype chip includes an embedded digital processor clocked at 100 MHz^2 that can access all on-chip components [65]. Its single instruction, multiple data (SIMD) vector unit can be used to efficiently read and write synaptic addresses and weights. Since the processor’s primary purpose is to compute synaptic weight updates, it is called the plasticity processing unit (PPU). In this work, we employ the PPU to simulate the agent and its environment, to inject spikes into the neural network from virtual spike sources, to manage the weight dynamics of the integration mechanism, for experiment control and for recording simulation states.

2.2 Network model and embodiment

The presented network model is based on Stone et al. [13] where a biologically plausible mechanism for path integration in bees is proposed. We refer the reader to the referenced publication for an exposition of the biological background. This section explains all functional aspects of the model and describes details about the presented implementation.

The state of the virtual insect is represented by four variables: the two-dimensional spatial position (x, y) , the head direction ϕ and the velocity v . Each experiment starts with a spread-out phase lasting for 50 ms (equivalent to 50 s in the biological time frame) in which the agent is forced on a random walk. This simulates a flight trajectory that the insect might take on its search for a food source. During this time, the network receives sensory input from the head orientation and the optical flow across its eyes, but is disconnected from the motor units and does, thus, not control the insect’s motion.

The neuronal representation of head orientation is modeled after the behavior of neurons in the *central complex*, a part of the insect brain that is critically involved in naviga-

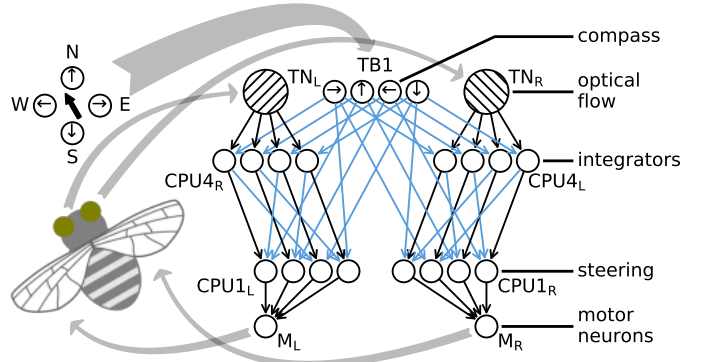


Figure 2: Network architecture. The black and blue arrows correspond to excitatory and inhibitory connections, respectively. The *columnar protocerebral bridge/upper division of the central body, type 1* (CPU1) and *columnar protocerebral bridge/upper division of the central body, type 4* (CPU4) populations are each divided into a left and right subpopulation, resulting in two logical hemispheres. The network topology relies on a physiological model proposed in [13].

tion. In particular, it is known that the *tangential protocerebral bridge, type 1* (TB1) group of neurons contained therein directly encodes the current head orientation like an internal biological compass. Each TB1 neuron corresponds to a specific absolute direction and its activity is proportional to the angular proximity of the current head orientation with respect to this direction [19, 66, 67]. For example, when the insect flies straight north, the neuron encoding north will be most active, whereas the neuron encoding south will be least active (see also eq. (1)). While the insect’s nervous system typically operates with a number of eight to nine such neurons, the resource constraints of the small BrainScaleS-2 prototype require reducing this number to four. Each of the modeled TB1 neurons therefore corresponds to one of the four cardinal directions (see fig. 2).

Odometric information, i.e. information on the agent’s speed, is derived from the optical flow across the left and right eye, respectively. Also in this case, the neuronal encoding is modeled after physiological observations. In particular, two *tangential noduli* (TN) neurons, which penetrate the *central complex*, coming from the *lateral accessory lobes* [13] spike with a rate proportional to the optical flow across the respective eye in one particular direction. For example, if the insect flies forward with maximum speed, the firing rate of these cells increases to a maximum. If it flies backward with sufficient speed, these neurons stop spiking. A detailed mathematical description of all sensors is given in section 4.1.

The output of both sensor populations is then processed by integrator neurons that mimic the CPU4 population found in the insect brain. Those cells are grouped into two symmetrical subpopulations, each containing eight to nine neurons which are, like in TB1, associated with the aforementioned azimuthal directions. Again, this number is reduced to four cardinal directions due to resource con-

²On the full-scale BrainScaleS-2 chip, this processor is clocked at 250 MHz.

straints on BrainScaleS-2, resulting in a total of eight emulated CPU4 neurons. In [13], each CPU4 cell implements a heuristically defined differential equation that integrates the presynaptic activities of TN and TB1 over time and outputs a numerical value in proportion to the accumulated value (see eq. (8) in section 4.3). As spiking LIF neurons are not capable of integrating or storing synaptic input signals in such a manner, we based the integration mechanism on axo-axonic synapses [68]. This synapse type plays an important role in the gill and siphon withdrawal reflex of the sea snail *Aplysia* [69] but is commonly found in insect brains as well [70]. Unlike in the more commonly found axodendritic synapses, the presynaptic neuron does not attach to the postsynaptic dendritic tree but to a postsynaptic axon terminal. Here, its activity can modulate the weight of the following synapse [71, 72], effectively implementing an activity-driven synaptic integration mechanism (see eq. (9) in section 4.3). Based hereon, the CPU4 population as a whole implements a working memory, in which the direction and distance to the home location is stored and continuously updated. Specifically, the vector connecting the current position to the origin is equal to the vector encoded by the CPU4 activities.

The vector pointing toward the home location, as encoded in the output of the CPU4 population, is then compared to the current movement direction provided by the TB1 cells. On BrainScaleS-2, this computation is performed by a symmetrically subdivided population of eight neurons in total modeled after biological CPU1 cells. Each CPU1 subpopulation receives excitatory one-to-one input from CPU4 with identical source and target index, but also inhibitory one-to-one input where the index assignment is rotated by 180° (see fig. 2). This differential mapping helps suppress the common average activity of the entire CPU4 population while amplifying the activity differences between the individual pairs of opposing cells, emphasizing the stored direction. Furthermore, the current direction of motion indicated by TB1 acts inhibitorily and rotated by 90° or -90° on the left or right CPU1 subpopulation. Consequently, the summed activity of the left and right CPU1 subpopulation is inversely proportional to the alignment between the home vector and the current heading direction tilted to the right and left, respectively. Hence, steering signals can be derived from CPU1 that guide the insect back to its nest. The corresponding summations are carried out by a left and a right motor neuron $M_{L/R}$. These neurons were not derived from observations of the bee brain but introduced to transfer computational load from the digital processor on BrainScaleS-2 to the accelerated neuromorphic substrate, improving the efficiency of the entire emulation.

In the second part of the experiment, the return phase, the insect’s motion is no longer externally imposed but derived from the outputs of these two motor neurons. They influence the insect’s motion by providing propulsion on the left and right hand side, similar to a tank drive.

All experiments run for 200 ms and the insect is set to return after $t_{\text{return}} = 50$ ms. Translated to the biological time domain, this corresponds to a total duration of around 3.3 min while the outbound journey lasts for 50 s.

Note that this duration is not intrinsic to model or hardware but can be set arbitrarily. Here, we adjusted it to provide quick experimental throughput, while keeping the time scales of the neural processes and the time scales of the foraging in a biologically realistic relation to each other. Specifically, the neuron dynamics take place within $\mathcal{O}(\text{ms})$ ($\mathcal{O}(\mu\text{s})$ on hardware) while the foraging happens within $\mathcal{O}(\text{min})$ ($\mathcal{O}(100 \text{ ms})$ on hardware).

Moreover, as the memory resources on the BrainScaleS-2 prototype are limited, the chosen duration provides a decent tradeoff between time resolution and length of the recorded movements and neural activities.

By bringing together all the mentioned components and settings, the agent can be put into operation.

2.3 Experiment execution

Figure 3 shows the trajectory and network activity for a single journey. While $t < t_{\text{return}}$, the insect performs a random walk through its environment. During this phase, it continuously integrates its current course, indicated by the compass neurons in TB1 and the odometric sensors TN, within the left and right CPU4 populations. While the activity among these neurons is initially homogeneous (at $t = 0$ s), the individual firing rates diverge as the insect moves away from its home location. The firing rates of the CPU4 neurons collectively encode the current vector from the insect to the home location.

Simultaneously, the CPU1 populations compute the differences between this vector and the current left and right-tilted heading direction while the motor neurons $M_{L/R}$ aggregate their activities.

After a fixed amount of time has passed (at $t = t_{\text{return}}$), the motor neurons $M_{L/R}$ assume control over the insect’s motion and steer it back home. A discrepancy in activity between M_L and M_R induces a change in direction for the insect, while similar activities set it on a straight path, autonomously regulating its course as long as the motor neurons remain in control. As the home vector shrinks, the firing rates of CPU4 are brought back into equilibrium.

In some instances, the insect first approaches an orthogonal axis (e.g the positive vertical axis in fig. 3) before taking a straight line back to its home. This artifact becomes more pronounced as the maximum traveled distance increases. It occurs when the input stimuli of CPU4 cells become high enough to saturate their output activities towards the maximum spike rate. As this happens, the home vector, as communicated to CPU1, gets compressed in the most dominant direction and therefore rotationally distorted, resulting in the observed behavior. However, as the distortion only emerges in the output activities of CPU4, not in the stored values, it does not influence agent’s accuracy or reliability.

After reaching its home location at $t \approx 100$ ms the insect begins swarming around it for another 100 ms. The motor neurons, under the influence of CPU1’s alternating decisions, trigger frequent turns, thus maintaining the agent in close proximity. As it thereby repeatedly approaches and departs from the origin, the home vector in CPU4 dynamically contracts, expands, and rotates.

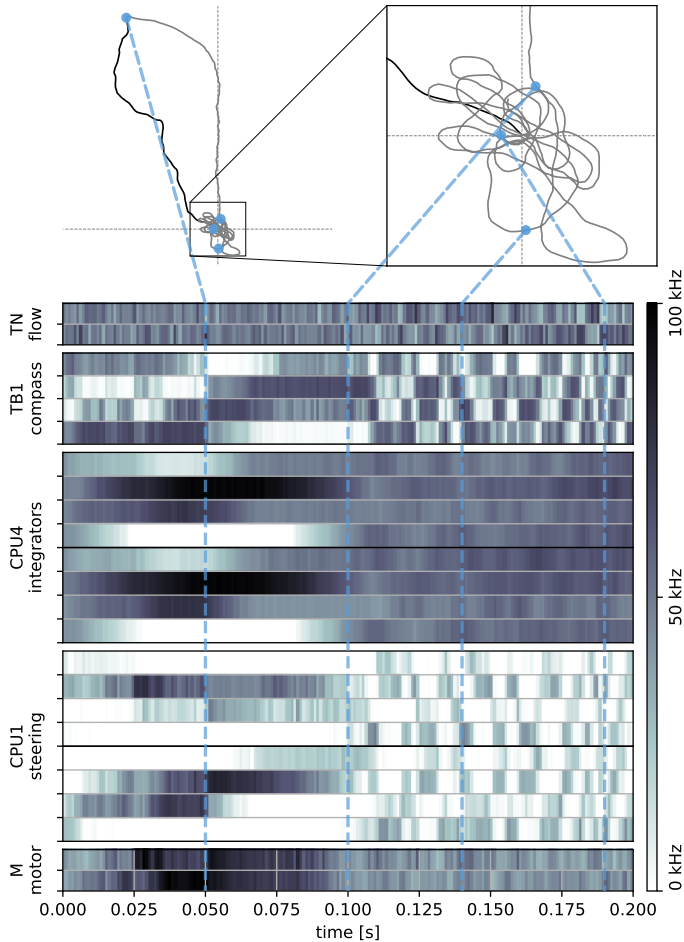


Figure 3: Network activity and trajectory. The top left plot shows the entire trajectory consisting of the outbound (black) and return phase (gray). The top right plot shows a zoom into a region measuring 5000×5000 steps around the origin where the agent loops around after his return. Four particular moments are highlighted as blue dots and connected to the network activity plot below: the moment of return, reaching the home location, and two situations during the looping phase. The neuron activities are measured as output spike counts in 1 ms intervals (corresponding to 1 s intervals in the biological time domain) and are displayed color-coded over time. The maximum count is approximately 100 which corresponds to an instantaneous spike rate of 100 kHz (equivalent to 100 Hz when translated into biological time scales).

The results, as depicted in fig. 3, validate that path integration-based navigation can be realized with accelerated neuromorphic circuits. They also show that the recorded movements and neural activities closely resemble those observed in biology [13, 15]. The experiment demonstrates that neural structures derived from insect brains can be mapped onto neuromorphic hardware to effectively control a virtual agent, thereby realizing a biomimetic cybernetic steering system.

2.4 Statistical performance and optimization

Furthermore, we performed statistical analyses to explore the properties and performance of the emulated insect. A series of 1000 independent experiments reveals that the agent always returns into the correct direction and typically stops at the correct distance. On average, the return location (computed as the mean position over 100 ms during looping) deviates from the actual home location by 842 steps. This falls within a circle around the origin, accounting for 4.9% of the median traveled outbound distance, or merely 0.24% of the area covered in by the median outbound journeys. The histogram over the trajectories during looping, i.e. after $t = 2 \cdot t_{\text{return}}$, is shown in fig. 4 A.

The distances given in integer unit lengths can be visualized more intuitively by relating them to physical quantities. Assuming typical traveling velocities of honeybees in the field (roughly 7 m s^{-1} according to [73]), the median journey radius corresponds to 618 m (17 309 steps) and the average deviation from the origin to approximately 23.6 m. Please note, however, that this might differ substantially when more accurate data for *Megalopta genalis* or other species is taken into account, or when the circumstances are modelled more accurately (considering varying flight speeds for outbound and return trips, obstacles, wind, etc.). The metric distances should therefore be understood solely as an illustrative aid.

To further improve the performance of the neuromorphic agent we optimized the synaptic weights based on an evolutionary strategy [74]. The optimization yielded a almost five-fold improvement in the average deviation of the return location, reducing it from 4.9% to 1.0%. Using the traveling velocities given above, this corresponds to a reduction from 23.6 m to 4.7 m. We also found that the insect’s motion around the return location becomes more narrow, reducing the average looping radius by 22%. The results are listed in more detail in table 1 and illustrated in fig. 4.

In line with a number of previous studies and experiments [33, 51, 55, 59, 60, 75], this confirms that device mismatches and inaccuracies intrinsic to analog neuromorphic hardware can be compensated by parameter optimization.

3 Discussion

This paper describes how a neural network model faithfully derived from physiological observations of insect brains can perform reliable and accelerated two-dimensional path integration on analog neuromorphic hardware. The navi-

	μ_x/μ_y	σ_x/σ_y	overlapping	within $r = 1000$
primitive	-313/-774	1419/1398	69.3 %	91.1 %
optimized	-75/150	1125/1066	88.7 %	98.9 %
relative improvement	-76%/-81%	-21%/-24%	+28 %	+8.6 %

Table 1: Mean displacement and standard deviation before and after evolutionary optimization. In the primitive state 69.3 % of the return trajectories overlap the home location directly and 91.1 % are within an $r = 1000$ radius around it. After evolutionary optimization this increases to 88.7 % and 98.9 %, respectively.

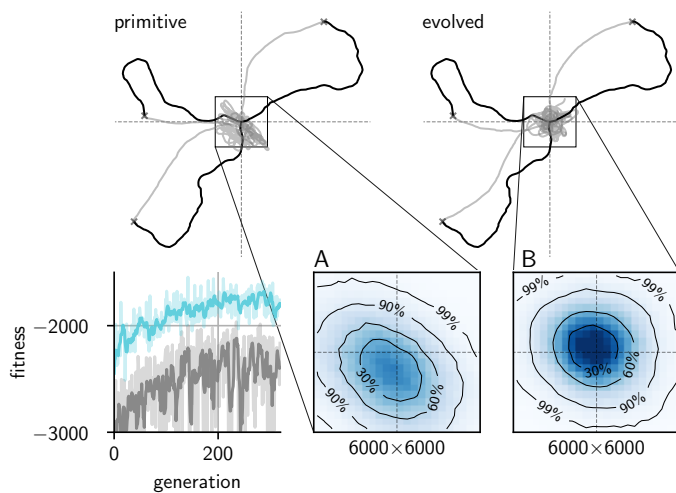


Figure 4: Statistical performance of the evolutionary optimization. Three sample trajectories generated by the primitive (left) and the evolved network (right). Below is a 6000×6000 zoom into the center region that shows a histogram over the data points of the looping phase of 1000 trajectories. The primitive network’s center of looping is shifted to the lower left with a broad and elliptically deformed looping area. The evolved network is more centered and exhibits tighter and more symmetric looping behavior. The lower left plot shows the population fitness (gray) and the fitness of the best three individuals (blue). For each individual, the faint line is the actual fitness whereas the strong line provides a moving average for better visibility. The optimization converges after ~ 200 steps.

gating agent thereby consists of the physically emulated neuromorphic brain connected to a virtual body via mixed-signal sensors and motor units. All these components are implemented and autonomously executed on a single neuromorphic chip.

To realize biologically viable integrators, we introduced a single-cell spike-based short-term memory mechanism based on axo-axonic synapses. This mechanism has not been implemented on neuromorphic hardware before and we are not aware of any studies in the computational neurosciences that have yet employed such a neural construct for similar purposes. As it is application-agnostic, it can potentially be utilized in various other scenarios beyond path integration, where flexible and network-interactive spike-based short-term memory cells might play a role.

Here, these memory cells provide the key elements for implementing the spike-based neuronal navigation circuit. Primarily through them, the underlying network model, which initially relied entirely on heuristically described differential equations [13], could be translated into a network of spiking LIF neurons. To our knowledge, this constitutes the first publication of a purely spike-based path integration model.

To comply with the limitations in network size of the BrainScaleS-2 prototype, we reduced the angular resolution of the neural compound from eight to four divisions without compromising the navigational functions. In addition, we omitted a population of *pontine* neurons which map input from the excitatory CPU4 populations inhibitory onto CPU1 [13]. This is possible because BrainScaleS-2 does not obey Dale’s principle [76], allowing the neuromorphic CPU4 populations to both excite and inhibit postsynaptic partners. Therefore, they can fully absorb the function of the *pontine* populations. With respect to the source model [13], the number of neurons in the path integration circuit (consisting of TB1, CPU4, CPU1, and optionally the *pontine* population) was, hence, reduced from 56 down to 20. As the head direction neurons in TB1 are implemented virtually on BrainScaleS-2, and since two additional motor neurons were introduced to offload numerical summations, the total number of neurons on hardware is only 18. The remarkable functional richness that the insect brain generates with so few neurons could thus be replicated and even further condensed on neuromorphic hardware.

While the primitive network configuration was already able to successfully navigate, its performance could be im-

proved with an evolution strategy-based optimization acting on the synaptic weights. The evolution of 15 individuals, each performing an entire spread-out and homing journey, through 320 generations took only 32 minutes on a single neuromorphic core. Had this evolution not been carried out on BrainScaleS-2 with its highly accelerated neurons but in the same configuration on a real-time system, it would instead have taken more than 11 days. Even on a multi-core real-time system with enough compute resources to fully parallelize all individuals, it would still have taken 17.8 hours to breed 320 consecutive generations. As each generation depends on its predecessor, this execution time can intrinsically not be further reduced on a real-time emulator.

Moreover, as the emulation speed on BrainScaleS-2 does not depend on the network size, the rapid experimental throughput can also be obtained for larger networks, and the longer the simulation time is, the more significantly this advantage manifests. Neuroplastic developments in mammals, for example, can extend over many weeks, months or even years. In order to simulate such processes and iteratively adapt their model parameters to biology, accelerated emulators are, therefore, absolutely indispensable.

In scenarios where the accelerated neural network needs to interact with external entities, like sensor organs, motor units, the entire body, or its environment, those entities have to be simulated at a higher speed as well. Otherwise, the various components would act on different time scales and the biological phenomena could, hence, not be reproduced with plausibility and accuracy. For the presented agent, the entities belonging to the extraneural domain were therefore modelled and implemented with the same acceleration factor on the digital on-chip co-processor to guarantee consistency and synchronicity between the brain and the body.

Thus, we demonstrated that the $1000\times$ acceleration factor enables biologically detailed neurobotic experiments at an unprecedented throughput, cutting down emulation times by three orders of magnitude as compared to similar studies carried out in biological real-time. Since the extent of such experiments is not physically limited and the technical capabilities of BrainScaleS-2 are by no means exhausted, we are confident that further neurobotic closed-loop experiments will take advantage of the considerable benefits of accelerated neuromorphic hardware in the future. This kind of experiments can substantially contribute to a better understanding of the functional interaction between neuronal structures and their environments.

4 Methods

4.1 Sensory inputs

The neural network receives input from two classes of sensors, the TB1 compass and the TN flow field sensors. Both use rate encoding to translate the sensory input into spike trains that can be processed by the LIF neurons at a maximum rate of $r_{\max} = 100$ kHz corresponding to 100 Hz in the biological time equivalent.

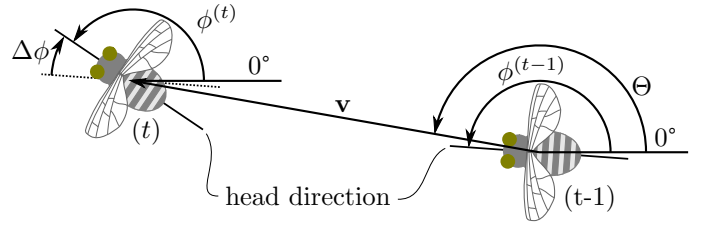


Figure 5: Coordinate system of the insectoid agent. Θ is the direction of flight and $\phi^{(t)}$ and $\phi^{(t-1)}$ the head direction of the current and the previous time step, respectively. $\Delta\phi$ is the difference between both and indicates the head rotation. As the time interval is discrete and defined to be 1, the traveled distance is equal to the velocity vector \mathbf{v} .

The TB1 population comprises four neurons $TB1_j$ with $j = \{0, 1, 2, 3\}$ that encode the head orientation ϕ in their output activity:

$$r_{TB1_j} = \frac{r_{\max}}{2} \cdot \left[1 + \sin\left(\phi + \frac{j \cdot \pi}{2}\right) \right]. \quad (1)$$

One TN flow field sensor measures a scalar velocity into one selected direction of its own movement relative to the environment. Its output depends on the head rotation ϕ relative to the insect's velocity \mathbf{v} , the static rotational separation ϕ_{TN} of the bilateral sensor relative to ϕ , and the head rotation $\Delta\phi^{(t)} = \phi^{(t)} - \phi^{(t-1)}$ between the last and the current time step (see fig. 5):

$$r_{TN_{L/R}} = \mp |\mathbf{v}| \cdot \sin(\Theta - \phi \pm \phi_{TN}) \pm \rho \cdot \Delta\phi. \quad (2)$$

Θ is the direction of the insect's velocity

$$\mathbf{v} = v \cdot \begin{bmatrix} \cos(\Theta) \\ \sin(\Theta) \end{bmatrix} \quad (3)$$

and ρ is a constant scaling factor for the head rotation that is proportional to the interocular distance between the two flow field sensors. For the experiment, we select $\rho = 2$ and restrict ourselves to $\Theta = \phi$. Stone et al. [13] demonstrated that the model can tolerate a significant constant offset $\Theta \neq \phi$.

Equation (2) can therefore be reduced to

$$r_{TN_{L/R}} = |\mathbf{v}| \cdot \sin(\phi_{TN}) \pm \rho \cdot \Delta\phi. \quad (4)$$

4.2 Motor outputs

Following Stone et al., the average activity of the left and right CPU1 populations determines the insect's direction of movement. We connect all neurons of a CPU1_{L/R} population to one motor neuron $M_{L/R}$ in order to compute the average. On BrainScaleS-2 these connections are realized by a single synapse that receives input from all four CPU1 neurons per hemisphere.

The motor neuron's spike rate is then translated into a directional velocity update

$$\Delta\phi = \Delta\Theta = \mu \cdot (r_{M_L} - r_{M_R}) \quad (5)$$

where $\mu = 1.6$ is a heuristically set scaling constant. The absolute velocity $|\mathbf{v}|$ stays constant.

4.3 Neuron model

The circuit proposed in [13] is based on a firing-rate neuron model with a synaptic input current

$$I_j = \sum_i w_{ij} r_i \quad (6)$$

of the j^{th} neuron. The output of the j^{th} neuron is obtained by applying a logistic function

$$r_j = \frac{1}{1 + e^{a_j \cdot I_j - b_j}} \quad (7)$$

to the internal state I_j , where a_j and b_j are two real-valued model parameters that are similar for all neurons. This transfer function is a commonly used choice to approximate the rate output of spiking neuron models under stochastic influence [77–82]. Our empirical measurements of the analog neurons on the chip have revealed similar transfer curves (see fig. 6B).

Equations (6) and (7) apply to the neurons in the three subcircuits TN, TB1, and CPU1. The neurons in CPU4, which are responsible for integrating over the traveled path, behave differently. Their internal state is given by

$$I_{\text{CPU4}}^{(t)} = I_{\text{CPU4}}^{(t-1)} + h \cdot (r_{\text{TN}}^{(t)} - r_{\text{TB1}}^{(t)} - k) \quad (8)$$

with a discrete time t , coupling strength h and decay constant k . It therefore depends on its own previous internal state and the output rates of TN on TB1. The time constant of this dynamical process is roughly on the order of the duration of the entire journey, i.e. around 200s in biological time.

Since the synaptic time constant of typical neurons is on the order of 1ms to 100ms and the membrane potential is reset after each firing event, eq. (8) cannot not solely emerge from the dynamics of a single LIF unit. Also, a single-cell feedback mechanism based on self-recurrent synapses would not allow for an integration over these time scales while maintaining the necessary precision and stability. Since each *glomerulus* [15] in the respective compartments of the *central complex* contains exactly one CPU4 neuron and only a small number of other neurons ($\mathcal{O}(1-10)$), it is also unlikely that the time scales required by eq. (8) arise from recurrent connections within small clusters of LIF-like cells.

Neural adaptations (as modeled by AdEx-equations [83] for instance) neither act on suitable time scales, as they typically last for no more than several seconds (usually several hundred ms) [84]. The same holds true for short-term synaptic facilitation and depression [85].

Among the remaining options are long-term synaptic plasticity and synaptic modulations, in particular, presynaptic facilitation and depression [71, 72]. As described in section 2.2, this type of synaptic modulation involves axo-axonic synapses which are formed between the axons of facilitating or depressing presynaptic neurons and axon terminals of postsynaptic neurons. While the axo-axonic synapse itself remains constant over time, the activity it projects on the postsynaptic axon terminal modulates the

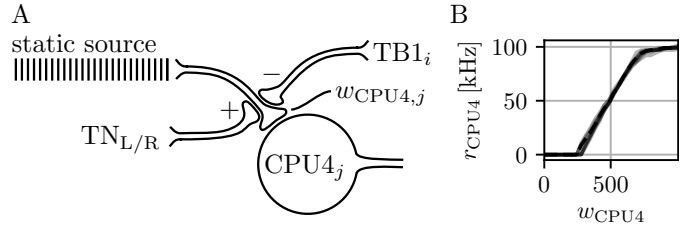


Figure 6: Axo-axonic depression and facilitation. A) A synapse that forms an excitatory connection to a background source with a constant spike rate is modulated by two axo-axonic synapses. Activity coming from the $\text{TN}_{\text{L/R}}$ axons facilitates the synaptic weight $w_{\text{CPU4},j}$, while activity from TB1_i inhibits it. The neuron responds with an output rate $r_{\text{CPU4},j}$. B) Output rate vs. weight. The plot shows an overlay of the transfer functions of eight CPU4 neurons on BrainScaleS-2.

weight of the following synapse between the targeted terminal and the corresponding postsynaptic neuron. Presynaptic modulations can occur on short-term memory time scales ($\mathcal{O}(10 \text{ min})$) [71] and, therefore, comply with the demands of eq. (8).

We implement eq. (8) through axo-axonic weight modulations of excitatory synapses coming from background spike sources that are firing periodically at a constant rate (see fig. 6). The strengths of those synapses are reduced or increased on short-term memory time scales by presynaptic facilitation or depression [71, 72]. In our case, these neurons correspond to the TN and TB1 subpopulations. Thus, in order to reproduce the behavior of eq. (8) on BrainScaleS-2, one background spike source with a constant average rate is connected to each integrating neuron through an excitatory synapse with a variable weight $w_{\text{CPU4},j}$. The integrating neuron therefore responds with an average output spike rate r that is proportional to this weight. To that end, the synapse realizes the short-term memory dynamics which are necessary for the distance integration. In analogy to eq. (8), we have

$$w_{\text{CPU4}}^{(t)} = w_{\text{CPU4}}^{(t-1)} + h \cdot (r_{\text{TN}}^{(t)} - r_{\text{TB1}}^{(t)} - k) \quad (9)$$

A parameter sweep revealed an optimal parameter combination of $k = 2$ and $h = 0.0336$.

We would like to add that the background spike sources fire periodically due to the tight timing demands of the code execution on the BrainScaleS-2 prototype. Spike sources with Poisson-distributed numbers of spikes per time interval would be favorable because they would introduce a more stochastic behavior, blurring out artifacts, and appear biologically more plausible. However, in simulation we found that the relevant statistical performance is not measurably affected by the strictly periodic sources and in hardware, jitter and intrinsic analog noise additionally mitigate possibly introduced artifacts to some degree. Moreover, the current full-scale version of BrainScaleS-2 offers configurable on-chip Poisson spike generators and, hence, renders the circumvention by periodic sources in future experiments obsolete.

4.4 Experiment scheduling

In order to run the environment and agent simulation in temporal coherence with the dynamics of the neural network, code execution has to be carefully scheduled. Every experimental run starts with transferring a set of parameters from the host computer to the BrainScaleS-2 system. This includes a random seed, the experiment run time t_{stop} , the time at which the agent is to return t_{return} , the CPU4 decay k , the CPU4 update scaling h , and the weights of all but the CPU4 input synapses. Subsequently, the code execution is started on the PPU.

During runtime, the PPU executes the simulation and implements the sensory input spike sources. One agent/environment update is conducted every $\Delta t = 100 \mu\text{s}$, which corresponds to a movement update rate of 10 Hz in the biological time equivalent. To achieve a maximum biological spike rate of 100 Hz, i.e., 100 kHz on the chip, the spike sources have to be able to send a spike every $10 \mu\text{s}$. As the code execution time for one full agent/environment update exceeds the time between two consecutive spikes, it is divided into three subcycles that are called between spike sending routines:

- iterate the insect state and transmit the sensory input to the TN and TB1 spike generators,
- read the CPU4 weights and calculate their updates,
- write back the updated CPU4 weights, process the motor neuron output and update the agent velocity accordingly, update the simulation state, and store the trajectory data.

At $t = t_{\text{return}}$, the agent switches from random to model-driven movement. The simulation stops at $t = t_{\text{stop}}$. At that point, the trajectory data and spikes are read back from the host computer.

The BrainScaleS-2 prototype chip can record all post-synaptic spikes using the connected FPGA, while the available on-chip memory for arbitrary data storage is limited to 4 kB. The latter is used for saving the trajectory from which the velocity and the optical flow can be reconstructed. Both the x and y coordinate are 16 bit signed integer values. Therefore, a total of 1000 locations can be stored. With $n = t_{\text{stop}}/\Delta t = 2000$ time steps in one run, every second step is written to memory. Figure 7 gives an overview of the scheduling.

4.5 Discretization of synaptic weights

While in [13], the internal state of the integrator neurons have floating point precision and a corresponding dynamic range, the 6 bit discretization on BrainScaleS-2 limits the number of possible weights per synapse to 64. In order to extend this range, we additively join 16 synapses together into one supersynapse. One presynaptic partner is connected to one postsynaptic neuron over 16 synapses, forming one supersynapse with a possible weight of 0 to $16 \cdot 63 = 1008$. The dynamic range is therefore extended from 6 bit to almost 10 bit.

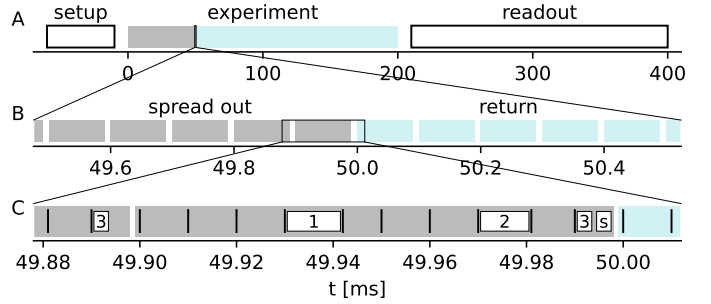


Figure 7: Experiment schedule. A) The experiment starts with a setup phase in which the parameters are transferred to the BrainScaleS-2 prototype and the synaptic weights are initialized. The actual experiment, starting with a spread out phase follows. After that, data is read back from the system memory to the host. This phase’s execution time is stochastic and depends on the amount of spikes produced in an experiment. B) Zoom into the point of return. Each block symbolizes an agent/environment update, the blue color indicates the return phase. C) Zoom into one update. The black bars mark the times at which the spike sending routine is called. The three update phases are represented by the white blocks 1, 2, and 3. Position recording happens only in every second update cycle in block s . Note that the variable duration of the update phase executions can introduce jitter of the following spike.

4.6 Calibration

Due to their analog design, the neuronal circuits on BrainScaleS-2 are subject to temporal and fixed-pattern noise. The latter is caused by manufacturing variations and results in instance-to-instance variations of the neuron and synapse parameters. In order to mitigate this effect, we calibrate the neuronal firing rates on each chip and store the results in a static data bank. For this experiment, two calibration routines are particularly important:

The CPU1 and motor neurons are calibrated such that their output rate is proportional to $r_{\text{out}} = r_{\text{exc}} \cdot (1 - r_{\text{inh}})$, where r_{exc} and r_{inh} is the excitatory and inhibitory input rate, respectively.

The CPU4 neurons, on the other hand, are calibrated to respond with an output rate that is proportional to the weight $w_{\text{CPU4},j}$ (see fig. 6B). All rates are normalized to a maximum of $r_{\text{max}} = 100 \text{ kHz}$ and a minimum of $r_{\text{min}} = 0 \text{ Hz}$. Functionally, it is relevant that the neuron parameters match well around the operating point $r = 0.5 \cdot r_{\text{max}}$, while the behavior at the range margins is of minor importance.

4.7 Evolutionary optimization

To mitigate fixed-pattern noise effects and further optimize network performance, we employ an evolution strategy with covariance matrix adaptation [74]. The optimization parameters are the 26 synaptic weights \mathbf{w} that are connected to the CPU1 population: TB1 \rightarrow CPU1,

CPU4 \rightarrow CPU1 (inhibitory and excitatory), and CPU1 \rightarrow M. To optimize for tight and precise looping around the home position, the fitness f_i of an individual i is derived from its trajectory $\mathbf{x}_i(t)_{t=[2 \cdot t_{\text{return}}, t_{\text{stop}}]}$ during looping:

$$f_i = - \left\langle |\langle \mathbf{x}_i \rangle_t| + \left| \sqrt{\langle (\mathbf{x}_i - \langle \mathbf{x}_i \rangle)^2 \rangle_t} \right| \right\rangle_{\text{runs}} \quad (10)$$

The first term is the time-averaged radial distance to the home location and the second term the time-averaged looping diameter. This sum is then additionally averaged over three runs with different outbound trajectories.

Instead of selecting a fixed number of best individuals of each generation, we take a weighted sum over the parameters of all individuals to derive the new mean parameter vector for the offspring population:

$$\boldsymbol{\mu} = \sum_i p_i \mathbf{w}_i \quad (11)$$

p_i is obtained as

$$p_i = \frac{\tilde{p}_i}{\sum_i \tilde{p}_i}; \tilde{p}_i = f_i^{-8} \quad (12)$$

The power of eight increases the impact of high-fitness genomes on the seed genome for the next generation, while the impact of low-fitness genomes is scaled down. Thus, weak individuals are not completely sorted out but still contribute to the next generation to a low degree. The exponent therefore implements a soft selection. Note that p_i is positive due to the even exponential index. Moreover, small absolute values of f_i translate to a high p_i and vice versa.

The new population consists of 15 weight vectors drawn from a multivariate Gaussian distribution:

$$\mathbf{w}_i \sim \mathcal{N}(\boldsymbol{\mu}, \sigma \cdot \boldsymbol{\Sigma}) \quad (13)$$

Here, $\sigma = 0.3$ is a heuristically chosen step size and $\boldsymbol{\Sigma}$ is a covariance matrix derived from the previous generation

$$\boldsymbol{\Sigma} = \sum_i p_i \mathbf{d}_i \otimes \mathbf{d}_i \quad (14)$$

with $\mathbf{d}_i = \mathbf{w}_i - \boldsymbol{\mu}$. In this way, the variance is increased into the direction of successful mutations. Typically, the optimization converges after approximately 200 generations.

Author Contributions

KS conceived the project and conceptualized, implemented and evaluated all experiments. TW implemented a simulation of the experiment, verifying its feasibility. TW, EM, PS, and YS contributed to the development of the BrainScaleS-2 operating system. PS offered consulting and support concerning the PPU compiler. SB, BC and YS developed parts of the calibration routines used in this work. CP implemented the FPGA-based spike router. MAP contributed to the manuscript and to the development of the BrainScaleS-2 system. JS is the leading hardware architect

of the BrainScaleS-2 system and head of the Electronic Vision(s) research group at Heidelberg University. Karlheinz Meier was the head of the research group in the early stages of this work and provided the initial financial, infrastructural, and motivational foundation without which none of this research could have been realized. This manuscript is dedicated to him personally and adds yet another facet to his extensive scientific accomplishments.

References

- [1] Xiaoya Ma, Xianguang Hou, Gregory D Edgecombe, and Nicholas J Strausfeld. Complex brain and optic lobes in an early cambrian arthropod. *Nature*, 490(7419):258–261, 2012.
- [2] Thomas S Collett and Matthew Collett. Memory use in insect visual navigation. *Nature Reviews Neuroscience*, 3(7):542–552, 2002.
- [3] Jason W Chapman, Don R Reynolds, Alan D Smith, Joe R Riley, David E Pedgley, and Ian P Woiwod. High-altitude migration of the diamondback moth *Plutella xylostella* to the UK: a study using radar, aerial netting, and ground trapping. *Ecological Entomology*, 27(6):641–650, 2002.
- [4] Andrea Adden, Sara Wibrand, Keram Pfeiffer, Eric Warrant, and Stanley Heinze. The brain of a nocturnal migratory insect, the australian bogong moth. *Journal of Comparative Neurology*, 528(11):1942–1963, 2020.
- [5] Karl Von Frisch. *The dance language and orientation of bees*. Harvard University Press, 1967.
- [6] Vivek Nityananda, Peter Skorupski, and Lars Chittka. Can bees see at a glance? *Journal of Experimental Biology*, 217(11):1933–1939, 2014.
- [7] Malcolm Burrows. *The neurobiology of an insect brain*. Oxford University Press on Demand, 1996.
- [8] Randolf Menzel and Martin Giurfa. Cognitive architecture of a mini-brain: the honeybee. *Trends in cognitive sciences*, 5(2):62–71, 2001.
- [9] Ann-Shyn Chiang, Chih-Yung Lin, Chao-Chun Chuang, Hsiu-Ming Chang, Chang-Huain Hsieh, Chang-Wei Yeh, Chi-Tin Shih, Jian-Jheng Wu, Guo-Tzau Wang, Yung-Chang Chen, et al. Three-dimensional reconstruction of brain-wide wiring networks in drosophila at single-cell resolution. *Current Biology*, 21(1):1–11, 2011.
- [10] Shinya Takemura, Arjun Bharioke, Zhiyuan Lu, Aljoscha Nern, Shiv Vitaladevuni, Patricia K Rivlin, William T Katz, Donald J Olbris, Stephen M Plaza, Philip Winston, et al. A visual motion detection circuit suggested by drosophila connectomics. *Nature*, 500(7461):175, 2013.
- [11] Shinya Takemura, Yoshinori Aso, Toshihide Hige, Allan Wong, Zhiyuan Lu, C Shan Xu, Patricia K Rivlin,

- Harald Hess, Ting Zhao, Toufiq Parag, et al. A connectome of a learning and memory center in the adult drosophila brain. *Elife*, 6:e26975, 2017.
- [12] C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shinya Takemura, Kenneth Hayworth, Gary Huang, Kazunori Shinomiya, Jeremy Maitin-Shepard, David Ackerman, Stuart Berg, et al. A connectome of the adult drosophila central brain. *BioRxiv*, 2020.
- [13] Thomas Stone, Barbara Webb, Andrea Adden, Nicolai Ben Weddig, Anna Honkanen, Rachel Templin, William Wcislo, Luca Scimeca, Eric Warrant, and Stanley Heinze. An anatomically constrained model for path integration in the bee brain. *Current Biology*, 27(20):3069–3085, 2017.
- [14] Kuo-Hua Huang, Peter Rupperecht, Thomas Frank, Koichi Kawakami, Tewis Bouwmeester, and Rainer W Friedrich. A virtual reality system to analyze neural activity and behavior in adult zebrafish. *Nature Methods*, 17(3):343–351, 2020.
- [15] Daniel B Turner-Evans and Vivek Jayaraman. The insect central complex. *Current Biology*, 26(11):R453–R457, 2016.
- [16] Kirsa Neuser, Tilman Triphan, Markus Mronz, Burkhard Poeck, and Roland Strauss. Analysis of a spatial orientation memory in drosophila. *Nature*, 453(7199):1244, 2008.
- [17] Christine Grienberger and Arthur Konnerth. Imaging calcium in neurons. *Neuron*, 73(5):862–885, 2012.
- [18] Johannes D Seelig and Vivek Jayaraman. Neural dynamics for landmark orientation and angular path integration. *Nature*, 521(7551):186–191, 2015.
- [19] Jonathan Green, Atsuko Adachi, Kunal K Shah, Jonathan D Hirokawa, Pablo S Magani, and Gaby Maimon. A neural circuit architecture for angular integration in drosophila. *Nature*, 546(7656):101–106, 2017.
- [20] Efstathios Kagioulis, Andrew Philippides, Paul Graham, James C Knight, and Thomas Nowotny. Insect inspired view based navigation exploiting temporal information. In *Conference on Biomimetic and Biohybrid Systems*, pages 204–216. Springer, 2020.
- [21] Chang Zhao, Yves F. Widmer, Soeren Diegelmann, Mihai A. Petrovici, Simon G. Sprecher, and Walter Senn. Predictive olfactory learning in drosophila. *Scientific Reports*, 11(6795), 2021. doi: 10.1038/s41598-021-85841-y.
- [22] Egidio Falotico, Lorenzo Vannucci, Alessandro Ambrosano, Ugo Albanese, Stefan Ulbrich, Juan Camilo Vasquez Tieck, Georg Hinkel, Jacques Kaiser, Igor Peric, Oliver Denninger, et al. Connecting artificial brains to robots in a comprehensive simulation framework: the neurorobotics platform. *Frontiers in neurorobotics*, 11:2, 2017.
- [23] Benedikt Feldotto, Jochen Martin Eppler, Cristian Jimenez-Romero, Christopher Bignamini, Carlos Enrique Gutierrez, Ugo Albanese, Eloy Retamino, Viktor Vorobev, Vahid Zolfaghari, Alex Upton, et al. Deploying and optimizing embodied simulations of large-scale spiking neural networks on HPC infrastructure. *Frontiers in neuroinformatics*, 16:34, 2022.
- [24] Benedikt Feldotto, Fabrice O Morin, and Alois Knoll. The neurorobotics platform robot designer: modeling morphologies for embodied learning experiments. *Frontiers in Neurorobotics*, 16:856727, 2022.
- [25] James EM Bennett, Andrew Philippides, and Thomas Nowotny. Learning with reinforcement prediction errors in a model of the drosophila mushroom body. *Nature communications*, 12(1):2569, 2021.
- [26] Julien Dupeyroux, Julien R Serres, and Stéphane Viollet. Antbot: A six-legged walking robot able to home like desert ants in outdoor environments. *Science Robotics*, 4(27), 2019.
- [27] Phil Husbands, Yoonsik Shim, Michael Garvie, Alex Dewar, Norbert Domcsek, Paul Graham, James Knight, Thomas Nowotny, and Andrew Philippides. Recent advances in evolutionary and bio-inspired adaptive robotics: Exploiting embodied dynamics. *Applied Intelligence*, 51(9):6467–6496, 2021.
- [28] Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael W Reimann, Marwan Abdellah, Carlos Aguado Sanchez, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, et al. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492, 2015.
- [29] Jakob Jordan, Tammo Ippen, Moritz Helias, Itaru Kitayama, Mitsuhisa Sato, Jun Igarashi, Markus Diesmann, and Susanne Kunkel. Extremely scalable spiking neuronal network simulation code: from laptops to exascale computers. *Frontiers in neuroinformatics*, 12:2, 2018.
- [30] James C Knight and Thomas Nowotny. GPUs outperform current HPC and neuromorphic solutions in terms of speed and energy when simulating a highly-connected cortical model. *Frontiers in neuroscience*, 12:941, 2018.
- [31] Gopal P Sarma, Chee Wai Lee, Tom Portegys, Vahid Ghayoomie, Travis Jacobs, Bradly Alicea, Matteo Cantarelli, Michael Currie, Richard C Gerkin, Shane Gingell, et al. OpenWorm: overview and recent advances in integrative biological simulation of *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society B*, 373(1758):20170382, 2018.
- [32] Raphaela Kreiser, Alpha Renner, Yulia Sandamirskaya, and Panin Pienroj. Pose estimation and map formation with spiking neural networks: towards neuromorphic slam. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2159–2166. IEEE, 2018.

- [33] Timo Wunderlich, Akos F. Kungl, Eric Müller, Andreas Hartel, Yannik Stradmann, Syed Ahmed Aamir, Andreas Grübl, Arthur Heimbrecht, Korbinian Schreiber, David Stöckel, Christian Pehle, Sebastian Billaudelle, Gerd Kiene, Christian Mauch, Johannes Schemmel, Karlheinz Meier, and Mihai A. Petrovici. Demonstrating Advantages of Neuromorphic Computation: A Pilot Study. *Frontiers in Neuroscience*, 2019. doi: 10.3389/fnins.2019.00260.
- [34] Raphaela Kreiser, Yulia Sandamirskaya, et al. Error-driven learning for self-calibration in a neuromorphic path integration system. In *Robust Artificial Intelligence for Neurorobotics (RAI-NR Workshop 2019)*, 2019.
- [35] Raphaela Kreiser, Gabriel Waibel, Yulia Sandamirskaya, et al. Self-calibration and learning on chip: towards neuromorphic robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019)*, 2019.
- [36] Raphaela Kreiser, Vanessa RC Leite, Baris Serhan, Chiara Bartolozzi, Arren Glover, Yulia Sandamirskaya, et al. An on-chip spiking neural network for estimation of the head pose of the icub robot. *Frontiers in Neuroscience*, 2020.
- [37] Elvin Hajizada, Patrick Berggold, Massimiliano Iacono, Arren Glover, and Yulia Sandamirskaya. Interactive continual learning for robots: a neuromorphic approach. In *Proceedings of the International Conference on Neuromorphic Systems 2022*, pages 1–10, 2022.
- [38] Yulia Sandamirskaya, Mohsen Kaboli, Jorg Conradt, and Tansu Celikel. Neuromorphic computing hardware and neural architectures for robotics. *Science Robotics*, 7(67):eabl8419, 2022.
- [39] Yannik Stradmann and Johannes Schemmel. Biomimetic control for high-speed robotic applications. In *7th HBP Student Conference on Interdisciplinary Brain Research*, pages 277–281. Frontiers Media SA, mar 2023.
- [40] Steve B Furber, Francesco Galluppi, Steve Temple, and Luis A Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014.
- [41] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Philipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.
- [42] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.
- [43] Charlotte Frenkel, Martin Lefebvre, Jean-Didier Legat, and David Bol. A 0.086-mm² 12.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos. *IEEE transactions on biomedical circuits and systems*, 13(1):145–158, 2018.
- [44] Charlotte Frenkel, Jean-Didier Legat, and David Bol. Morphic: A 65-nm 738k-synapse/mm² quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning. *IEEE Transactions on Biomedical Circuits and Systems*, 13(5):999–1010, 2019.
- [45] Garrick Orchard, E Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T Sommer, and Mike Davies. Efficient neuromorphic signal processing with loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259. IEEE, 2021.
- [46] Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R Chandrasekaran, Jean-Marie Bussat, Rodrigo Alvarez-Icaza, John V Arthur, Paul A Merolla, and Kwabena Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.
- [47] Ning Qiao, Hesham Mostafa, Federico Corradi, Marc Osswald, Fabio Stefanini, Dora Sumislawska, and Giacomo Indiveri. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Frontiers in neuroscience*, 9:141, 2015.
- [48] Saber Moradi, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs). *IEEE transactions on biomedical circuits and systems*, 12(1):106–122, 2017.
- [49] Alexander Neckar, Sam Fok, Ben V Benjamin, Terrence C Stewart, Nick N Oza, Aaron R Voelker, Chris Eliasmith, Rajit Manohar, and Kwabena Boahen. Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proceedings of the IEEE*, 107(1):144–164, 2018.
- [50] Johannes Schemmel, Laura Kriener, Paul Müller, and Karlheinz Meier. An accelerated analog neuromorphic hardware system emulating NMDA-and calcium-based non-linear dendrites. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2217–2226. IEEE, 2017.
- [51] C Pehle, S Billaudelle, B Cramer, J Kaiser, K Schreiber, Y Stradmann, J Weis, A Leibfried, Müller E, and Schemmel J. The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity. *Front. Neurosci.*, 16(795876), feb 2022. doi: 10.3389/fnins.2022.795876.

- [52] Manuel Le Gallo, Riduan Khaddam-Aljameh, Milos Stanisavljevic, Athanasios Vasilopoulos, Benedikt Kersting, Martino Dazzi, Geethan Karunaratne, Matthias Brändli, Abhairaj Singh, Silvia M Mueller, et al. A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference. *Nature Electronics*, pages 1–14, 2023.
- [53] Timo C Wunderlich, Akos F Kungl, Eric Müller, Johannes Schemmel, and Mihai Petrovici. Brain-inspired hardware for artificial intelligence: Accelerated learning in a physical-model spiking neural network. *Artificial Neural Networks and Machine Learning - ICANN 2019: Theoretical Neural Computation*, pages 119–122, 2019. doi: 10.1007/978-3-030-30487-4_10.
- [54] Sebastian Billaudelle, Yannik Stradmann, Korbinian Schreiber, Benjamin Cramer, Andreas Baumbach, Dominik Dold, Julian Göltz, Akos F Kungl, Timo C Wunderlich, Andreas Hartel, et al. Versatile emulation of spiking neural networks on an accelerated neuromorphic substrate. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2020.
- [55] Sebastian Billaudelle, Benjamin Cramer, Mihai A Petrovici, Korbinian Schreiber, David Kappel, Johannes Schemmel, and Karlheinz Meier. Structural plasticity on an accelerated analog neuromorphic hardware system. *Neural Networks*, 133:11 – 20, 2021. doi: 10.1016/j.neunet.2020.09.024. 0893-6080.
- [56] Akos F Kungl, Sebastian Schmitt, Johann Klähn, Paul Müller, Andreas Baumbach, Dominik Dold, Alexander Kugele, Nico Gürtler, Eric Müller, Christoph Koke, et al. Accelerated physical emulation of bayesian inference in spiking neural networks. *Frontiers in Neuroscience*, 13:1201, 2019. doi: 10.3389/fnins.2019.01201.
- [57] Korbinian Schreiber, Timo C Wunderlich, Christian Pehle, Mihai A Petrovici, Johannes Schemmel, and Karlheinz Meier. Closed-loop experiments on the BrainScaleS-2 architecture. In *Proceedings of the 2020 Annual Neuro-Inspired Computational Elements Workshop*, pages 1–3, 2020.
- [58] Korbinian Schreiber. *Accelerated neuromorphic cybernetics*. PhD thesis, Universität Heidelberg, January 2021.
- [59] J Göltz, L Kriener, A Baumbach, S Billaudelle, O Brei-wieser, B Cramer, D Dold, AF Kungl, W Senn, J Schemmel, Karlheinz Meier, and Mihai A. Petrovici. Fast and energy-efficient neuromorphic deep learning with first-spike times. *Nature Machine Intelligence*, 3: 823–835, 2021. doi: 10.1038/s42256-021-00388-x.
- [60] Benjamin Cramer, Sebastian Billaudelle, Simeon Kanya, Aron Leibfried, Andreas Grübl, Vitali Karasenko, Christian Pehle, Korbinian Schreiber, Yannik Stradmann, Johannes Weis, Johannes Schemmel, and Friedemann Zenke. Surrogate gradients for analog neuromorphic computing. *Proceedings of the National Academy of Sciences*, 119(4), 2022. doi: 10.1073/pnas.2109194119.
- [61] Tarek M Massoud and Timothy K Horiuchi. A neuromorphic vlsi head direction cell system. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58 (1):150–163, 2010.
- [62] Raphaela Kreiser, Matteo Cartiglia, Julien NP Martel, Jörg Conrard, and Yulia Sandamirskaya. A neuromorphic approach to path integration: a head-direction spiking neural network with vision-driven reset. In *2018 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–5. IEEE, 2018.
- [63] Syed Ahmed Aamir, Yannik Stradmann, Paul Müller, Christian Pehle, Andreas Hartel, Andreas Grübl, Johannes Schemmel, and Karlheinz Meier. An accelerated lif neuronal network array for a large-scale mixed-signal neuromorphic architecture. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(12): 4299–4312, 2018.
- [64] Christian Mayr, Sebastian Hoepfner, and Steve Furber. Spinnaker 2: A 10 million core processor system for brain simulation and machine learning. *arXiv preprint arXiv:1911.02385*, 2019.
- [65] Simon Friedmann, Johannes Schemmel, Andreas Grübl, Andreas Hartel, Matthias Hock, and Karlheinz Meier. Demonstrating hybrid learning in a flexible neuromorphic hardware system. *IEEE transactions on biomedical circuits and systems*, 11(1):128–142, 2016.
- [66] Stanley Heinze and Uwe Homberg. Maplike representation of celestial e-vector orientations in the brain of an insect. *Science*, 315(5814):995–997, 2007.
- [67] Stanley Heinze and Steven M Reppert. Sun compass integration of skylight cues in migratory monarch butterflies. *Neuron*, 69(2):345–358, 2011.
- [68] Eric R Kandel, James H Schwartz, Thomas M Jessell, Department of Biochemistry, Molecular Biophysics Thomas Jessell, Steven Siegelbaum, and AJ Hudspeth. *Principles of neural science*, volume 5. McGraw-hill New York, 2013.
- [69] Eric Richard Kandel. *Cellular basis of behavior: An introduction to behavioral neurobiology*. W. H. Freeman, 1976.
- [70] Casey M Schneider-Mizell, Stephan Gerhard, Mark Longair, Tom Kazimiers, Feng Li, Maarten F Zwart, Andrew Champion, Frank M Midgley, Richard D Fetter, Stephan Saalfeld, et al. Quantitative neuroanatomy for connectomics in drosophila. *Elife*, 5: e12059, 2016.
- [71] John H Byrne and Eric R Kandel. Presynaptic facilitation revisited: state and time dependence. *Journal of Neuroscience*, 16(2):425–435, 1996.

- [72] Harvey Lodish, Arnold Berk, S Lawrence Zipursky, Paul Matsudaira, David Baltimore, and James Darnell. *Molecular Cell Biology*. W. H. Freeman, 4 edition, 2000.
- [73] A. M. Wenner. The flight speed of honeybees: a quantitative approach. *Journal of Apicultural Research*, 2(1):25–32, 1963.
- [74] Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation*, pages 312–317. IEEE, 1996.
- [75] Sebastian Schmitt, Johann Klähn, Guillaume Bellec, Andreas Grübl, Maurice Guettler, Andreas Hartel, Stephan Hartmann, Dan Husmann, Kai Husmann, Sebastian Jeltsch, et al. Neuromorphic hardware in the loop: Training a deep spiking network on the brain-scales wafer-scale system. In *2017 international joint conference on neural networks (IJCNN)*, pages 2227–2234. IEEE, 2017.
- [76] Piergiorgio Strata and Robin Harvey. Dale’s principle. *Brain research bulletin*, 50(5-6):349–350, 1999.
- [77] Charles F Stevens and Anthony M Zador. When is an integrate-and-fire neuron like a poisson neuron? In *Advances in neural information processing systems*, pages 103–109, 1996.
- [78] Nicolas Brunel and Simone Sergi. Firing frequency of leaky integrate-and-fire neurons with synaptic current dynamics. *Journal of theoretical Biology*, 195(1):87–95, 1998.
- [79] Peter Dayan and Laurence F Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT press, 2001.
- [80] Nicolas Fourcaud and Nicolas Brunel. Dynamics of the firing probability of noisy integrate-and-fire neurons. *Neural computation*, 14(9):2057–2110, 2002.
- [81] Rubén Moreno-Bote and Néstor Parga. Role of synaptic filtering on the firing response of simple model neurons. *Physical review letters*, 92(2):028102, 2004.
- [82] Mihai A Petrovici, Johannes Bill, Ilja Bytschok, Johannes Schemmel, and Karlheinz Meier. Stochastic inference with spiking neurons in the high-conductance state. *Physical Review E*, 94(4):042312, 2016.
- [83] Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology*, 94(5):3637–3642, 2005.
- [84] Richard Naud, Nicolas Marcille, Claudia Clopath, and Wulfram Gerstner. Firing patterns in the adaptive exponential integrate-and-fire model. *Biological cybernetics*, 99:335–347, 2008.
- [85] Misha Tsodyks, Klaus Pawelzik, and Henry Markram. Neural networks with dynamic synapses. *Neural computation*, 10(4):821–835, 1998.