

A compact neuromorphic system for ultra-energy-efficient, on-device robot localization

Adam D Hines^{1*}, Michael Milford¹, Tobias Fischer¹

¹QUT Centre for Robotics, Queensland University of Technology, 2 George St, Brisbane, QLD 4001, Australia

*Corresponding author. Email: adam.hines@qut.edu.au

Neuromorphic computing offers a transformative pathway to overcome the computational and energy challenges faced in deploying robotic localization and navigation systems at the edge. Visual place recognition, a critical component for navigation, is often hampered by the high resource demands of conventional systems, making them unsuitable for small-scale robotic platforms which still require accurate long-endurance localization. Although neuromorphic approaches offer potential for greater efficiency, real-time edge deployment remains constrained by the complexity of bio-realistic networks. In order to overcome this challenge, fusion of hardware and algorithms is critical to employ this specialized computing paradigm. Here, we demonstrate a neuromorphic localization system that performs competitive place recognition in up to 8 kilometers of traversal using models as small as 180 kilobytes with 44,000 parameters, while consuming less than 8% of the energy required by conventional methods. Our Locational Encoding with Neuromorphic Systems (LENS) integrates spiking neural networks, an event-based dynamic vision sensor, and a neuromorphic processor within a single SynSense Speck™ chip, enabling real-time, energy-efficient localization on a hexapod robot. When compared to a benchmark place recognition method, Sum-of-Absolute-Differences (SAD), LENS performs comparably in overall precision. LENS represents an accurate fully neuromorphic localization system capable of

large-scale, on-device deployment for energy efficient robotic place recognition.
Neuromorphic computing enables resource-constrained robots to perform energy efficient, accurate localization.

Introduction

Robot localization is a critical component of many autonomous navigation systems, enabling robots to determine their location whilst supporting the ability to understand and interact with their surroundings. A central challenge in robot localization is visual place recognition (VPR), which requires robots to identify and classify previously visited locations under varying conditions (1–3). Conventional VPR strategies often rely on deep convolutional neural networks or transformer-based architectures to robustly extract features from the environment (4–7). For resource-constrained robotic platforms that nonetheless require the ability to keep track of where they are located over vast distances, using such conventional VPR methods is often not feasible due to their computational demands. An open challenge for real-world deployment of VPR is to find models that are computationally and energy efficient, and can be deployed at the edge.

Neuromorphic computing, which takes inspiration by the brain, has emerged as a promising solution for addressing the energy and computational challenges associated with robot localization and navigation (8–12). The human brain’s ability to learn and navigate complex environments efficiently (13–16) has inspired roboticists to create more computationally efficient localization systems (17–21). In particular, Spiking Neural Networks (SNNs) have been widely explored due to their ability to perform tasks through bio-realistic simulations of neuron activity (22, 23). Although several SNN models have been proposed for robotic localization (10, 12, 24–32), they ultimately have not yet fully lived up to the promise of computationally efficient deployment due to the complexity of continually modeling neurons in real-time (19, 24, 26).

To advance the practicality of neuromorphic computing in VPR, the focus must shift toward developing models that can be effectively deployed on Size, Weight, and Power (SWaP) constrained robotic platforms. One approach to overcoming deployment barriers is to trade the intricate bio-realism of traditional SNNs for simpler, more efficient networks that still deliver robust performance (33). However, small-scale systems can often be limited in their ability to map

large environments, which restricts the downstream use-cases (11). To improve efficiency, SNNs full potential for robotic deployment can be realized by integrating them with neuromorphic hardware (10–12, 19, 34–37) such as Intel®’s Loihi 2, TrueNorth, and Tianjic (8, 38, 39), which draw inspiration from neuroscience for their design, enabling them to transmit and receive physical spikes from neurons within the processor cores (40). Neuromorphic computing has been successfully utilized in tasks other than localization, such as control systems for drone flight (41), route following (11), autonomous driving controllers (42), and motor controllers (43). In contrast, non-spiking algorithms that are brain-inspired are widely used in localization tasks but have limited ability to be deployed on neuromorphic hardware. Although several neuromorphic localization systems have been developed and tested, they are limited by environment scale (11), require complex multi-modal fusion of large network models (12), or have restricted real-time capability (10).

Event cameras, including Dynamic Vision Sensors (DVS), offer further computational advantages when paired with neuromorphic processors as they only transmit information based on pixel-wise changes in light intensity exceeding a threshold, thereby reducing unnecessary data processing (35). In the context of SNNs, spiking activity can easily be triggered and propagated based on incoming input event streams and are well suited to process them. Fusing neuromorphic algorithms, sensors, and processors to perform robotic localization tasks therefore provides a promising avenue to overcome the computational limitations inherent in SNN-based VPR (10, 12).

Here, we introduce a neuromorphic pipeline for robotic localization that is compact, real-time capable, and able to map environments up to 8 km in length. Our system, Locational Encoding with Neuromorphic Systems (LENS), is designed to deliver high accuracy and efficiency with a model size of less than 180 KB and 44 K parameters, capable of processing up to 8 km of traversal data while consuming less than 8% of the energy required by conventional platforms. We train a SNN using a temporal time to first spike encoding scheme designed to perform VPR (27) and deploy our model on the SynSense Speck™ (44), which combines a DVS and system-on-chip neuromorphic processor. The LENS neuromorphic algorithm was developed in tandem with neuromorphic hardware deployment to deliver ultra energy-efficient place recognition for robotic localization. By deploying on Speck™, we forgo the need for any external sensory modalities or computational resources as all localization is performed on-chip. By abstracting complex bio-realism in favor of enhanced performance, LENS represents a notable advancement in neuromorphic

localization, as a fully event-driven platform designed for VPR. We validate LENS on a HiWonder JetHexa hexapod robot (Fig. 1), demonstrating its effectiveness in both indoor and outdoor small-scale traversals (≈ 25 -40 m, 50-80 places) and large-scale datasets (≈ 8 km, 600 places) (45), showcasing its potential for online VPR.

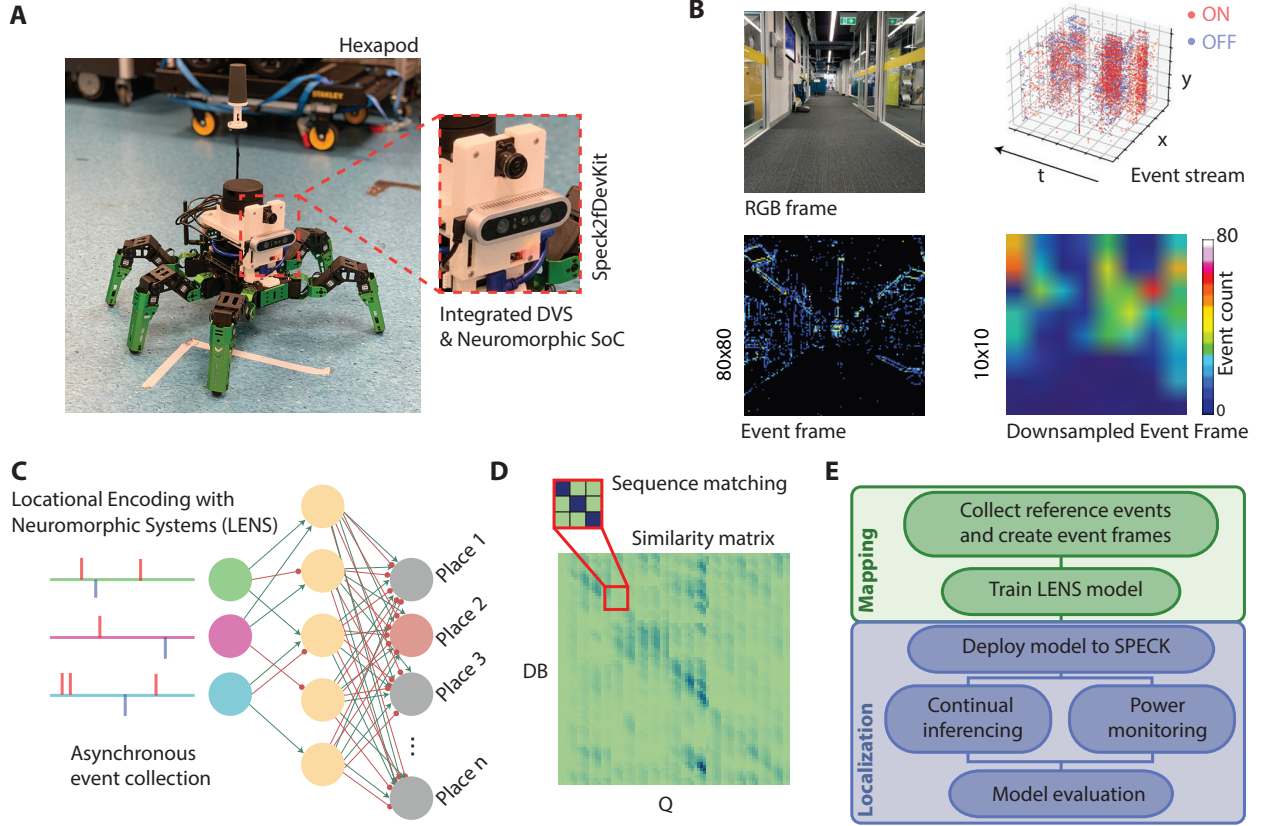


Figure 1: Neuromorphic place recognition system on a hexapod robot. (A) Our system was deployed on a HiWonder Jetahexa hexapod robotic platform equipped with a custom 3D-printed mount housing a SynSense Speck2fDevKit. Event streams captured by the onboard DVS are processed into event frames by counting ON and OFF events during the hexapod’s traversal (B). The initial 128×128 input is cropped to a 80×80 region of interest which is then downsampled to 10×10 by selecting central pixels via a 2D convolutional layer (see Materials and Methods: Event processing). (C) The spiking neural network, Locational Encoding with Neuromorphic Systems (LENS), learns reference event frames for on-chip deployment, enabling real-time localization using asynchronously collected events. (D) Sequence matching techniques enhance the similarity between reference and query inputs, improving overall precision. (E) Schematic of the fully neuromorphic visual place recognition pipeline.

Results

Locational encoding using neuromorphic systems An overview of our Locational Encoding using Neuromorphic Systems (LENS) network is presented in Figure 2. We used event streams from the Speck™ DAVIS128 sensor in two different ways for the mapping and localization phases of the robot navigation. A place was represented as one second of movement, translating to a distance of ≈ 0.15 m (Fig. 2A). For the mapping phase, the number of events were counted per pixel and then flattened into a 2D-temporal representation image of the place (Fig. 2B). Using a 2D-convolutional kernel with the center weight set to 1 and all other weights set to 0 (see Materials and Methods: Model training), we selected the center pixel from the 2D image to downsample to 10×10 pixels to fit in our LENS system (Fig. 2B). Following Hines et al. (27), we trained our model by first normalizing pixel intensities between $[0, 1]$ to use a temporally coded spike scheme such that higher pixel intensities result in earlier spiking with lower intensities activating later (Fig. 2C, see Materials and Methods: Model training). Connection weights were trained via spike-timing dependent plasticity learning, with enhanced connectivity when post-synaptic activity occurred after pre-synaptic spikes and diminished connectivity if post-synaptic neurons spike beforehand (Fig. 2C, see Materials and Methods: Model training). For our three layer architecture, we used unsupervised learning between the input and feature layers and a supervised delta learning rule between the feature and output layer (see Materials and Methods: Model training).

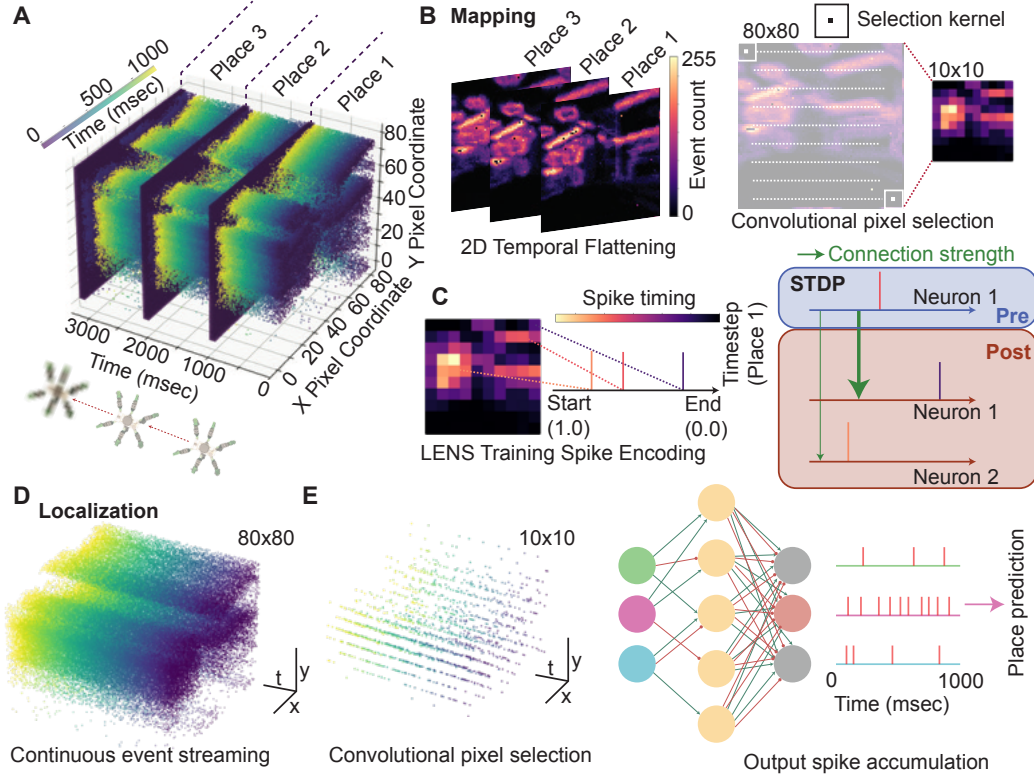


Figure 2: Event stream processing using LENS. (A) Event streams are collected from the Spec2fDevKit board from an 80×80 region of interest using a DAVIS128 sensor as our hexapod robot moves through the environment. One place represents one second or ≈ 0.15 m of movement. The event stream is then used in two unique ways during mapping and localization phases. (B) During mapping, events over the one second time period are counted and flattened into a 2D-temporal representation of the place. We use a convolutional selection kernel to pick the center pixel from the 2D image (see Materials and Methods: Model training), resulting in a 10×10 down sampled image. (C) Down sampled images are normalized in the range of $[0, 1]$ with pixel intensity determining spike timing in our LENS training regime, following (27). Using spike-timing dependent plasticity (STDP) across the network layers, we strengthen or weaken both excitatory and inhibitory connections if the spike occurs after or before pre-synaptic activity, respectively. (D) During localization, we forgo the temporal representation of places and use the raw event streams to continuously activate and bias the LENS network on-chip. (E) We apply the same convolutional pixel selection as the first layer of our neural network, deployed on the Speck™ directly. Over the time collection period of one second, we accumulate output spikes from our model and make a place prediction based on the neuron with the highest spiking activity.

During the localization phase, we used the raw event streams collected from the Speck™ directly into our model instead of 2D-temporal representations of places (Fig. 2D, see Materials and Methods: Deployment time). The first layer of our LENS system was the convolutional pixel selection filter applied earlier to reduce the input dimensionality to 10×10 pixels (Fig. 2E, see Materials and Methods: Deployment time). We performed place recognition with asynchronous events over one second collection windows and picked the location with the highest spiking activity for the neuron in the output layer (Fig. 2E, see Materials and Methods: Online place matching).

Power and energy efficiency Neuromorphic computing platforms offer an advantage in terms of power and energy consumption when compared to traditional von Neumann hardware, particularly in tasks like Visual Place Recognition (VPR) where energy efficiency is crucial for long-term operation on battery-powered robots. We measured the power consumed by various components of the Speck™ neuromorphic processor while performing VPR (Fig. 3A). Notably, when the robot was stationary, the DVS generated very few events which resulted in further reductions in power consumption. To quantify the energy savings of LENS deployed on the Speck™ processor relative to von Neumann hardware, we performed VPR using the Sum-of-absolute-differences (SAD) place matching method (46) on an Intel® i7-9700K CPU and a NVIDIA Jetson Nano, which ran the hexapod (see Materials and Methods: Power measurements). After subtracting baseline CPU power use, we found that the Speck™ consumed an average of 2.7 mW, which is just 8.8% and 0.5% of the power required by the Jetson Nano and Intel® CPU to run SAD, respectively (Fig. 3B).

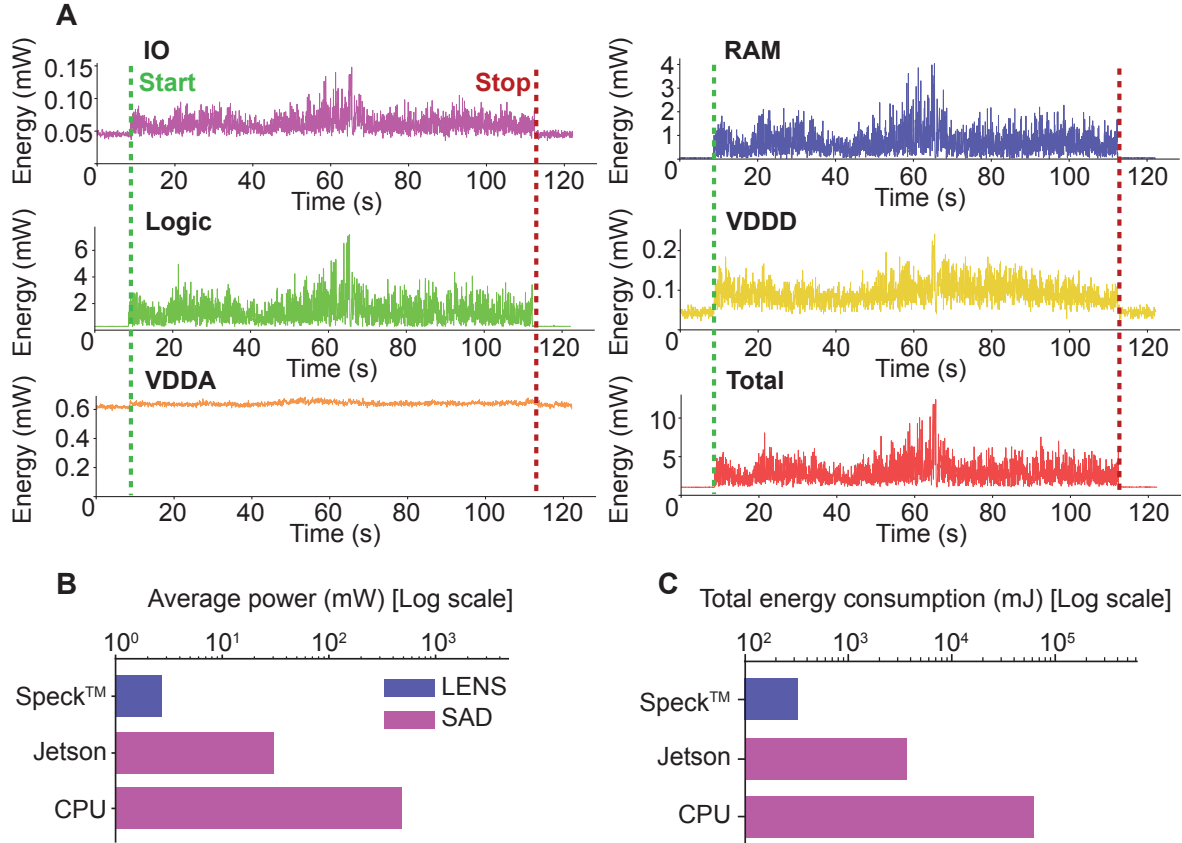


Figure 3: Energy efficiency of Real-time VPR on a neuromorphic processor vs. von Neumann hardware. (A) Power consumption was recorded across 5 components of the Speck™ neuromorphic processor: input/output (IO), memory (RAM), logic, digital components (VDDD), analog components (VDDA), and the total. Start indicates when the robot began to move until the navigation task has finished, indicated by the Stop. (B). The Speck™ processor consumed substantially less total energy compared to the Jetson Nano and Intel® CPU (C). The mW and mJ values are presented on a logarithmic scale for display purposes.

The overall energy consumption (the integral of power over time) of the Speck™ was 327 mJ, with the Jetson using 2,968 mJ and the Intel® CPU requiring 61,427 mJ, meaning overall our system required just 8.9% and 0.5% of the energy needed to perform the same task on von Neumann hardware (Fig. 3C). Fewer pixel representations of places still resulted in energy consumption far greater than what was required for LENS on Speck™ (see Table S2). We also observed a prominent reduction in power and energy consumption deploying to Speck™ when running our LENS system

on von Neumann hardware (Fig. S1).

Large-scale place recognition with compact models To assess our system’s suitability for real-world scenarios, we evaluated its performance in large scale environments using the Brisbane Event VPR dataset containing event streams from an ≈ 8 km route in Brisbane, Australia (45) (Fig. 4). This dataset was representative of a stable, four wheeled robotic platform that our model could be deployed on. We generated 641 event frames of unique places by sampling over one second intervals from the dataset (Fig. 4B). To fit the model onto the Speck™ chip, we used a network architecture of 49 input neurons, 63 feature layer neurons, and 641 output neurons (Fig. 4B). This was the largest and highest performing model architecture we could use to run our model on-chip, balancing the number of input pixels and feature representation (Fig. S6). We used the “sunset2” traverse for mapping (reference dataset), and the “sunset1” traverse for localization (query dataset). We also employed the Sum-of-absolute-difference (SAD method) (46, 47) as a baseline comparison method, calculating pixel-wise similarity between mapping (training) and localization (test) data.

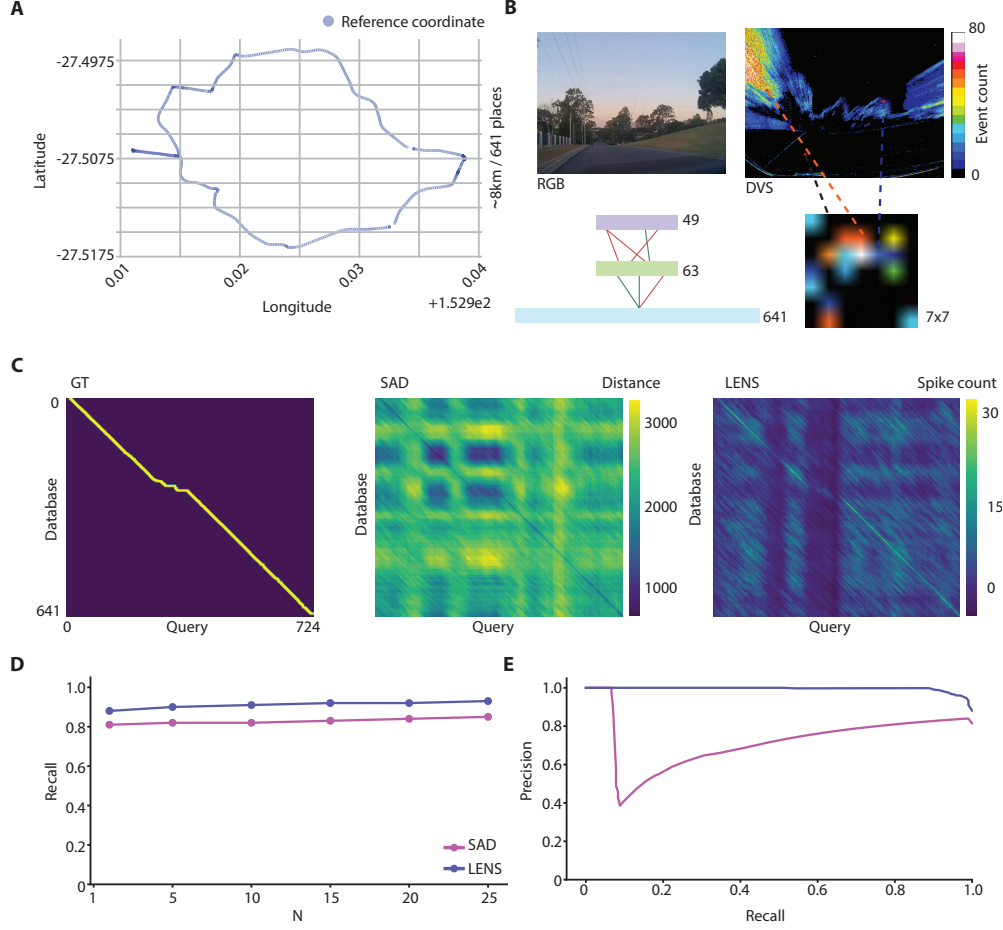


Figure 4: Large scale place recognition using a compact neuromorphic ecosystem. (A) Coordinate map of the Brisbane Event VPR dataset route (45) which represents an ≈ 8 km traversal and 641 unique place locations around the suburbs of Brisbane, Australia. (B) Event frames are generated in the same way as on robot experiments (Fig. 5). Pixels were selected randomly rather than by applying a 2D convolutional filter due to a larger event camera resolution used in (45). We selected pixels to generate event frames that are 7×7 in size. Our model architecture was modified to $49 \times 63 \times 641$ to allow for on-chip localization. (C) Similarity matrices for both SAD and LENS, showing how the similarity for both methods aligns closely with the ground truth (GT). (D) Recall@N and (E) Precision-Recall curves for our method when compared to SAD. LENS performed best overall compared to SAD, and also achieved better precision at lower recall indicating our system is more confident at the places it selected. The energy efficiency and compact model size of just 753 neurons with 44K parameters highlights the benefit of our system.

We used spikes from the model’s output layer and the Euclidean distance calculated in SAD to generate similarity matrices and used sequence matching which preserves the distances of sequences matches while nullifying cross-distances, improving localization precision during deployment time (Fig. 4C, see Materials and Methods: Online place matching). The similarity matrix was used to compute Recall@N (the proportion of true place matches retrieved within the top N predicted matches) and precision and recall metrics, which evaluated how well the robot is correctly identifying its location as it moves through the environment. Although other methods for event based place recognition exist, they require the reconstruction of images from event streams for deep feature extraction (48, 49). As such, we elected to not compare our system to these methods since they require higher complexity and computational time than what is feasible to run on our robotic platform.

Both LENS and SAD generated similarity matrices that closely aligned with the ground truth (Fig. 4C). LENS performed best overall for Recall@N, achieving a Recall@1 of 0.88 compared to SAD’s 0.81 (Fig. 4D). LENS also demonstrated better precision at lower recall, indicating higher confidence in its place recognition (Fig. 4E). The non-monotonic behavior in some precision-recall curves, where precision temporarily improves with increasing recall, occurred when the rate of true positive matches increased faster than false positives at certain threshold ranges. A spiking neural network to achieve this degree of accuracy with an architecture of just 753 neurons across three layers shows promise for large scale place recognition applications. We required the same amount of storage space capacity to store our model and the down sampled images for SAD, which was ≈ 179 KB.

Localization performance during a navigation task To evaluate our system’s real-time performance at the edge, we performed VPR on our hexapod in both indoor and outdoor environments (Fig. 5). The model architecture was designed with $100 \times 200 \times 75$ for the input, feature, and output neurons, respectively. As illustrated in Figs. 5A & F, the generated event frames corresponded to the environment being traversed, particularly by detecting edges in the scene. The robot dealt with viewpoint and timing differences between the mapping and localization phases, as the route paths differed due to imperfect teleoperation (Figs. 5B & G). In comparison to the Brisbane Event Dataset, the hexapod represented a more unstable and challenging platform to perform place recognition.

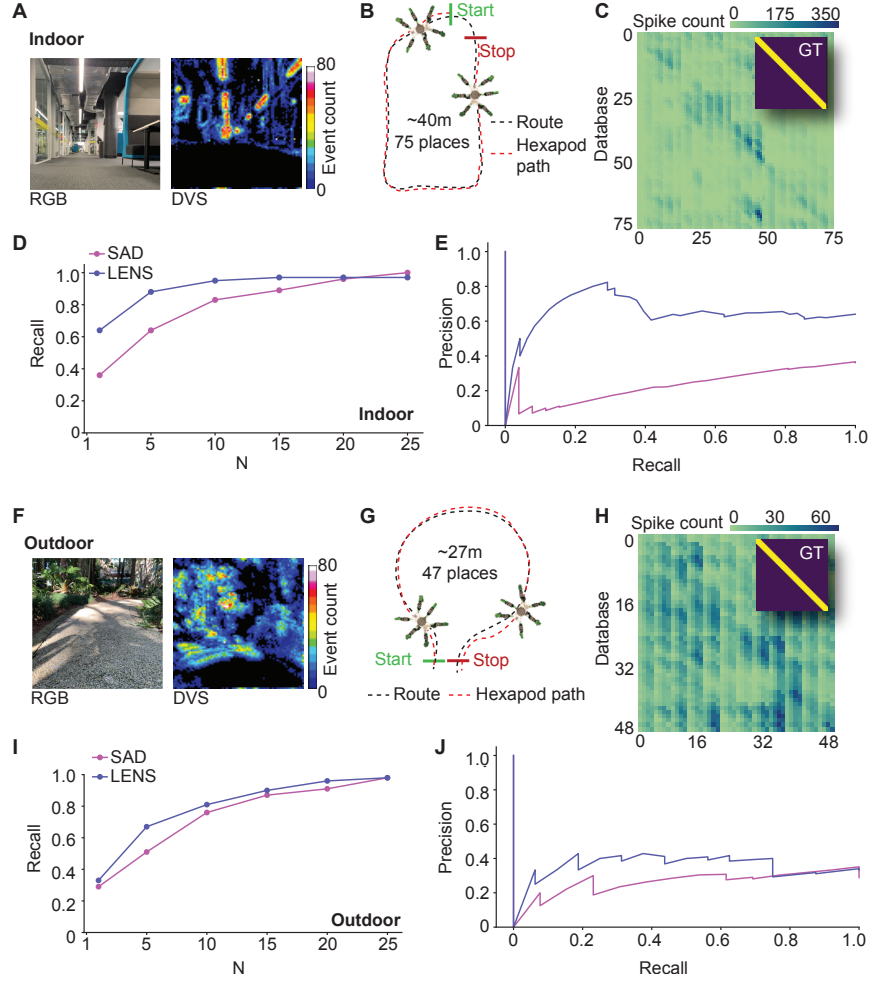


Figure 5: Precision of a real-time neuromorphic robotic localization platform. We deployed our system on a Hexapod robotic platform for on-device localization. (A) First, we tested an indoor environment by mapping and localizing along a predetermined route (B). As the Hexapod was teleoperated, there are slight differences in viewpoint between the mapping and localization phases. (C) To analyze the precision and recall of our indoor environment, we generated sequentially matched similarity matrices of the output spikes from all output neurons in our model while localizing. (D) Our compact system generally performed better than the sum of absolute differences (SAD) method for Recall@N, measuring accuracy in the top N matches, and Precision-Recall (E). We then deployed our Hexapod in an outdoor environment (F, G). The similarity matrix for the outdoor environment had more aliased incorrect matches than the indoor one (H), most likely due to the presence of multiple turns in the route. (I, J) In the outdoor task, LENS performed comparably to the SAD method in both the Recall@N and Precision-Recall evaluation.

For the indoor task, LENS achieved a Recall@1 of 0.64, outperforming SAD’s performance at 0.36 (Fig. 5C & D). The precision-recall curve (PR-curve) indicated that LENS achieved higher precision at lower recall values compared to SAD (Fig. 5E), suggesting that LENS was highly accurate in its identification of places. In the outdoor task, LENS performed comparably to SAD with a Recall@1 of 0.33 and 0.29, respectively (Fig. 5H & I). LENS also achieved a higher precision at lower recall, shown in the PR-curve (Fig. 5J). The reduced performance in outdoor environments was likely due to the increased amount of turns in the trajectory, which caused widespread neuronal activation across the field of view. As indicated by the similarity matrix (Fig. 5H), LENS showed a higher amount of activity across multiple output neurons. Fig. S8 suggests that increased turning in the navigation route introduced noise, distributing spikes across more neurons.

Overall, our compact spiking neural network model (130-150 KB in size) demonstrated comparable performance for VPR tasks in multiple environments, with key advantages of real-time deployability whilst being highly power and energy efficient. By contrast, the SAD method required 388 KB to store the images, ≈ 2.6 times the storage space needed for our model. Our system offered substantial benefits in terms of model size, making it feasible for deployment on computationally resource-constrained platforms.

Interestingly, both SAD and LENS performed better on the Brisbane Event VPR dataset than during the on-robot VPR task (Fig. 5). The Brisbane Event Dataset utilized a 346×240 sensor (45), compared to the 128×128 sensor (50) on our robot. Additionally, the more stable platform (i.e., a car) used for dataset collection could reduce errors from viewpoint variance or swaying movement observed with the hexapod (see Supplementary Movie 1). These results indicate that our proposed model can be effectively deployed for both short and long-range place recognition tasks for robotic localization.

Discussion

This work demonstrated a fully neuromorphic visual place recognition system capable of real-time localization, which was successfully deployed on a hexapod robotic platform without any external sensing or computational resources. The LENS spiking neural network model, characterized by its simplicity and compactness, performed accurately across multiple environments and scales,

demonstrating its versatility for a range of navigation platforms and tasks. Our system integrated with the SpeckTM chip operated with 1-2 orders of magnitude lower energy than conventional von Neumann hardware, underscoring its viability for deployment on resource-constrained platforms such as unmanned aerial vehicles (51), underwater (52), and locomotive robotics (53).

Notably, our models were less than 180 KB in size, with the largest containing just 44 K parameters. Accurate localization with our compact model enables portable deployment on neuromorphic processors with higher memory capability for larger models to improve overall performance (Fig. S6). Previous research has shown that accurate place recognition with event cameras can be achieved with as few as 25 pixels (47). Our system capitalized on this by balancing the number of pixels with the size of the feature layer to achieve the optimal performance (Fig. S6). A limiting factor in this work was the memory capacity of the neurocores for SpeckTM, which cap at 64 KB per core. Overcoming this limitation with neuromorphic hardware offering higher memory capabilities could further enhance the performance of large-scale place recognition tasks (Fig. S6). Nonetheless, we have shown comparable performance for place recognition in highly compact spiking network models across multiple environments.

Our neuromorphic place recognition system employs real event streams directly for localization in a unique way, establishing a new methodology for event-driven place recognition. This work highlights the potential for substantial efficiency gains when using event-based DVS over conventional cameras, especially in scenarios where energy-efficient processing is critical. Unlike frame based cameras, which operate continuously, the event-driven DVS captures only essential changes, reducing data redundancy and energy consumption. Our experiments encompass both a conventional benchmark dataset widely used in the robotics community (45) and a more challenging legged robot navigation task, ensuring that the localization accuracy of our method is evaluated across a diverse range of environments and baseline performance levels. The accuracy achieved by LENS demonstrates its effectiveness within broader systems such as Simultaneous Localization and Mapping (SLAM), where geometric constraints and filtering methods can mitigate occasional inaccuracies (54–56). By utilizing a more stable platform, such as a four wheeled robot, our network is able to achieve high precision comparable to a benchmark technique. This demonstrates that our compact spiking network models perform comparably to conventional non-event-driven methods, highlighting the potential for further improvements with future developments in neuromorphic

hardware and algorithmic enhancements.

As observed in the similarity matrices for the localization tasks (Fig. 5), introducing many turns into shorter trajectories proved challenging for LENS in the outdoor hexapod experiment. Dynamically adjusting these biases and parameters of DVS event streams to better handle these sudden motions could enhance accuracy across challenging environments (57). Continued development of this system will focus on detecting and handling turning motions to improve localization performance.

Another limitation of our system is the use of static, temporal representations of places generated from event streams using time-window binning for model training (Fig. 2). Training our model directly on DVS event streams could help overcome the localization challenges observed in dynamic environments. Developing this would also facilitate online learning capabilities, where event streams are used directly to train network models in real-time. This would enable the system to adapt to new environments on-the-go, potentially incorporating introspective capabilities that allow the robot to recognize when it is uncertain about its location and learn accordingly (58–61). Additionally, using more information from the DVS sensor such as including convolutional layers and self-supervised learning could be useful to extract more features for enhanced performance (62).

Integration of our neuromorphic system with other methods for navigation could be used to enhance and improve robotic localization performance. Dupeyroux *et al.* developed a bio-inspired path integration system for a hexapod robotic platform that would serve as a complementary method for VPR (63). Additionally, other insect inspired systems employ methods such as goal approaching and collision avoidance which allows for autonomous navigation in novel environments, typically for homing tasks (11, 63, 64). Integration of these mixed-model paradigms present themselves as a promising method to be used in more comprehensive navigation systems such as SLAM, of which there are many neuroscience inspired systems available (18, 65–67).

In conclusion, we present a fully end-to-end neuromorphic VPR system that performs accurate, low-power localization on robotic platforms. Our work has the potential to enable robots in the field to navigate further and conduct longer missions by reducing the computational and energy demands of current vision systems (68). An important aspect of this work was to ensure that neuromorphic localization could be achieved on multiple robotic platforms, including for resource-constrained scenarios. Our approach moves VPR toward efficient, versatile neuromorphic localization, unlock-

ing new possibilities for autonomous robotic navigation at the edge.

Materials and Methods

Model training Our network models were trained on CPU hardware or a Jetson Nano using static temporal representations of event streams with pixel intensity values relating to the number of events collected over a one second time-window, subsequently normalized in the range $[0, 1]$ (see Fig. 2). The network was trained using spike timing from an abstracted spiking method, where the pixel intensity value represents temporal activity (27, 69, 70). The networks three layer architecture (input, feature, and output) used unsupervised learning between the input and feature layers, and a supervised delta learning rule for the feature to output layer.

Unsupervised learning applied spike-timing dependent plasticity (STDP) to update weights W , encouraging or pruning connections between neurons. Initial weights and connections were randomly generated from predetermined probabilities, with connections either being positive (*excitatory connection*) or negative (*inhibitory connection*). From the input to feature layer there was a 35% and 75% excitatory and inhibitory connection probability, respectively. The feature to output layer was fully connected with an equal distribution of excitatory and inhibitory connections. As spikes were propagated through the network, the layer weights were updated in the following way:

$$\Delta W_{ji}^{nm}(t) = \frac{\eta_{\text{STDP}}(t)}{f_j^n} \cdot \left[\Theta(x_i^m(t-1)) \cdot \Theta(x_j^n(t)) \cdot (0.5 - x_j^n(t)) \right], \quad (1)$$

where W_{ij}^{nm} is the connection weight between neuron j in layer n to neuron i in layer m , η_{STDP} is the STDP learning rate, f_j^n is the target firing rate of neuron j in layer n , $\Theta(\cdot)$ is the Heaviside step function, x_i^m and x_j^n are the neuron states x in the network layer i and j with connections m and n , and t is the timestep. During weight updates throughout training, we did not specify any rule that says weights must remain excitatory or inhibitory, however we did not observe weights switching sign. That is, initially seeded connection types remained robust throughout training.

For supervised learning between the feature and output layer, we employed a delta learning rule to force each output neuron to learn the feature representation of a single reference place (27, 69):

$$\Delta W_{ji}^{nm}(t) = \eta_{\text{STDP}}(t) / f_j^n \cdot [x_i^m(t-1)(x_{\text{force } j}^n(t) - x_j^n(t))], \quad (2)$$

where x_{force} was the forced spike in the selected output neuron during training.

The hyperparameters used for our models can be found in Table S1 which were obtained by performing a hyperparameter sweep.

Deployment time The Locational Encoding with Neuromorphic Systems (LENS) framework leveraged spiking neural networks (SNN) on neuromorphic processors for computationally and energy efficient place recognition. We deployed our pre-trained SNN model on a SynSense Speck2fDevKit, which incorporates the Speck™ neuromorphic processor with a 128×128 dynamic vision sensor (DVS) (44) for event driven, real-time localization. This was in contrast to the training methodology where event frames were generated by counting the number of events over a specified time-window. Our localization system used truly asynchronous event streams to continually bias and activate neurons as the robot navigated through the environment.

To optimize input size and remove low-activity pixels, we first selected an 80×80 region of interest (ROI) in the top-center of the sensor. Events from this ROI were processed using a 2D convolution layer with a kernel size and stride of 8, which acted as the first processing layer of our SNN prior to the pre-trained input layer and was deployed on chip. We set the center weight of the kernel to 1, with all other weights at 0, effectively selecting the center pixel from each convolution to reduce the input to 100 neurons. This selection approach followed a previously established method for event pixel selection (47).

The input neurons were sparsely connected to a linear feature layer of 200 neurons, with excitatory and inhibitory connection probabilities set at 35% and 75% respectively. The feature layer was fully connected to the output layer, with the number of output neurons corresponding to the number of learned places (see Materials and Methods: Model training). Events were asynchronously collected and passed into our model, where Constant-leak Integrate and Fire (IAF) neurons propagated spikes across the network layers:

$$\tau \dot{v} = -v_{\text{leak}} + R \cdot (I_{\text{syn}} + I_{\text{bias}}), \quad (3)$$

where τ is the membrane time constant, v is the membrane potential, R is the constant resistance value used to match units between membrane potential and currents, I_{syn} is the weighted sum of all input synaptic contributions, and I_{bias} is a constant bias.

Online place matching Place matching was performed using real-time sequence matching (46, 71) to enhance the precision of our VPR model on the SpeckTM. Spike counts from the output layer were collected in one second time bins, then averaged over four time bins (four seconds), representing ≈ 0.5 m of robot movement (average speed of $13.5 \text{ cm} \cdot \text{s}^{-1}$). All place and sequence matching was performed off-chip on a Jetson Nano during on-robot deployment or a CPU for simulated experiments. Once four sets of output spike counts were gathered, representing about 2 m of travel, they were compiled into a distance matrix M_{dist} (Fig. 1D), which was convolved (*) with an identity matrix \mathbb{I}_L to generate a sequence-based distance matrix M_{seq} , as previously described (71):

$$M_{\text{seq}}(i, j) = \mathbb{I}_L * M_{\text{dist}}(i, j) / L, \quad (4)$$

where i and j are the row and column of the matrix, respectively, and L is the sequence length, which we set to 4 matching the number of output spike counts. For the Brisbane Event Dataset, we set the sequence length to 30. The choice of sequence length matters for both computational efficiency and evaluation performance, with shorter lengths favoring the former and longer lengths benefiting the latter (Fig. S7).

After generating the sequence-based distance matrix M_{seq} , we performed place matching on the processed spike outputs. The matched place \hat{p} was identified as the place i corresponding to the neuron with the highest spike count x_i for each query in the M_{seq} :

$$\hat{p}_k = \arg \max_i M_{\text{seq}}(k, i) \quad \text{for each } k \in 1, 2, 3, 4. \quad (5)$$

where \hat{p}_k is the matched place for the k -th sequence, i ranged over all database reference places, $M_{\text{seq}}(k, i)$ represents the sequence-based distance between the k -th spike count set and the i -th reference place.

Off-chip simulation of event streams For power and energy consumption comparisons using LENS on neuromorphic and von Neumann hardware, we simulated event-driven localization using event frames from the SpeckTM energy experiments (see Materials and Methods: Power and energy measurements). Integrate And Fire neuron spike rates were generated using a time-based rate code:

$$I_{t,h,w} = \begin{cases} 1 & \text{if } R_{t,h,w} < X_{h,w} \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where $I_{t,h,w}$ is a boolean tensor with spatial dimensions h and w over t time-windows, where 1 = spike and 0 = no spike. $R_{t,h,w} \in [0, 1]$ is a uniformly random tensor, and $X_{h,w}$ is the original input. Localization was simulated over 1000 time windows with $dt = 0.001s$ to represent 1s of output spike collection, similar to the online place matching (see Materials and Methods: Online place matching).

Robotic deployment We used the commercially available JetHexa hexapod (Hiwonder, Shenzhen China) customized with a 3D-printed mount to house the Speck2fDevKit (see Fig. 1). We also simulated a four wheeled robotic deployment by evaluating our system on the Brisbane Event Dataset (45), to highlight that our energy-efficiency performance is robotic platform agnostic. Training data was collected by teleoperating the robot for a reference traversal. Real-time localization was performed on the SpeckTM neuromorphic processor with the pre-trained model closely following the reference traversal. Indoor and outdoor datasets represented approximately 40 m and 27 m of traversal, encoding 75 and 48 unique places, respectively. To ensure accurate ground truth alignment between the mapping and localization phases, experiments were consistently started and ended at the same location.

Power and energy measurements Intel[®] CPU power was measured using SoC Watch, and NVIDIA Jetson Nano power was recorded with the jetson-stats Python package, with baseline idle power subtracted in both cases to only measure energy consumed for the VPR algorithms. SpeckTM power tracks (IO, RAM, memory, VDDA, and VDDD) were recorded at 20 Hz using SynSense’s samna package during testing (see Supplementary Movie 1 and Fig. 3).

Statistical analysis We evaluated our system using Recall@N and precision-recall curves, two standard metrics for VPR studies (1). All statistical analysis was performed on CPU hardware. Our implementation of precision and recall matched methods used in other place recognition systems (12) (Fig. S2). Recall@N measured the accuracy of the system such that for $N = 1$ only the highest match was considered across all reference places, whereas for $N = 5$ the top 5 matches were considered when calculating the accuracy, up to a maximum of 25 potential places. Precision-recall curves were utilized to assess performance by illustrating the relationship between precision and recall across different threshold levels for positive matches, calculated as follows:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{GTP} \quad (7)$$

where TP is the number of true positives, FP is the number of false positives, and GTP is the number of ground truth positives (1). Initially precision and recall is undefined due to there being no predictions, however we initialize precision at 1 and recall at 0 for visualization purposes.

We used the Sum of Absolute Differences (SAD) method for comparison, which computed pixel-wise similarity across reference and query images and has been used previously for place recognition using event streams (46, 47). Pseudo ground-truth data was collected following (47) using GPS correspondence between reference and query images. Pseudo-GPS and manually confirmed ground-truth showed high correspondence, with the majority of places being ± 1 place label away (Fig. S3). Further analysis of the accuracy of the pseudo ground-truth and applied tolerances were confirmed by near identical results (Fig. S4). We allowed a ground-truth tolerance of ≈ 1 m, which is equivalent to ± 2 reference places, and a tolerance of ≈ 42 m for the Brisbane Event VPR dataset, which is equivalent to ± 3 reference places. A visualization of the ground truth, TP, and FP matches for each similarity matrix can be found in Fig. S5.

Data and code availability All data needed to support the conclusions of this manuscript are included in the main text, Supplementary Materials, and are also available alongside the code at <https://doi.org/10.5281/zenodo.15392412>.

References and Notes

1. S. Schubert, P. Neubert, S. Garg, M. Milford, T. Fischer, Visual Place Recognition: A Tutorial. *IEEE Robotics and Automation Magazine* **31**, 139–153 (2024).
2. X. Zhang, L. Wang, Y. Su, Visual place recognition: A survey from deep learning perspective. *Pattern Recognition* **113**, 107760 (2021), doi:<https://doi.org/10.1016/j.patcog.2020.107760>.
3. C. Masone, B. Caputo, A Survey on Deep Visual Place Recognition. *IEEE Access* **9**, 19516–19547 (2021), doi:[10.1109/ACCESS.2021.3054937](https://doi.org/10.1109/ACCESS.2021.3054937).
4. G. Berton, C. Masone, B. Caputo, Rethinking Visual Geo-Localization for Large-Scale Applications, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), p. 4878–4888.
5. G. Neuhold, T. Ollmann, S. R. Bulò, P. Kotschieder, The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes, in *IEEE International Conference on Computer Vision* (2017), p. 5000–5009, doi:[10.1109/ICCV.2017.534](https://doi.org/10.1109/ICCV.2017.534).
6. M. Cordts, *et al.*, The Cityscapes Dataset for Semantic Urban Scene Understanding, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2016), p. 3213–3223.
7. S. Izquierdo, J. Civera, Optimal Transport Aggregation for Visual Place Recognition, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), p. 17658–17668.
8. G. Orchard, *et al.*, Efficient Neuromorphic Signal Processing with Loihi 2, in *IEEE Workshop on Signal Processing Systems* (2021), p. 254–259.
9. E. Painkras, *et al.*, SpiNNaker: A multi-core System-on-Chip for massively-parallel neural net simulation, in *IEEE Custom Integrated Circuits Conference* (2012), p. 1–4, doi:[10.1109/cicc.2012.6330636](https://doi.org/10.1109/cicc.2012.6330636).
10. L. Zhu, M. Mangan, B. Webb, Neuromorphic sequence learning with an event camera on routes through vegetation. *Science Robotics* **8** (82), 1–13 (2023), doi:[10.1126/scirobotics.adg3679](https://doi.org/10.1126/scirobotics.adg3679).

11. T. van Dijk, C. D. Wagter, G. C. H. E. de Croon, Visual route following for tiny autonomous robots. *Science Robotics* **9** (92), 1–13 (2024), doi:10.1126/scirobotics.adk0310.
12. F. Yu, *et al.*, Brain-inspired multimodal hybrid neural network for robot place recognition. *Science Robotics* **8** (78), 1–12 (2023), doi:10.1126/scirobotics.abm6996.
13. K. Friston, The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience* **11** (2), 127–138 (2010).
14. D. Chen, N. Axmacher, L. Wang, Grid codes underlie multiple cognitive maps in the human brain. *Progress in Neurobiology* **233**, 102569 (2024), doi:https://doi.org/10.1016/j.pneurobio.2024.102569.
15. J. E. Lisman, O. Jensen, The Theta-Gamma Neural Code. *Neuron* **77**, 1002–1016 (2013).
16. C. Liu, R. Todorova, W. Tang, A. Oliva, A. Fernandez-Ruiz, Associative and predictive hippocampal codes support memory-guided behaviors. *Science* **382** (6668) (2023), doi:10.1126/science.adi8237.
17. M. J. Milford, G. F. Wyeth, Mapping a suburb with a single camera using a biologically inspired SLAM system. *IEEE Transactions on Robotics* **24** (5), 1038–1053 (2008).
18. F. Yu, J. Shang, Y. Hu, M. Milford, NeuroSLAM: a brain-inspired SLAM system for 3D environments. *Biological Cybernetics* **113** (5), 515–545 (2019).
19. J. Wang, S. Lin, A. Liu, Bioinspired Perception and Navigation of Service Robots in Indoor Environments: A Review. *Biomimetics* **8** (4) (2023), doi:10.3390/biomimetics8040350.
20. L. Wang, S. X. Yang, M. Biglarbegian, Bio-inspired Navigation of Mobile Robots, in *Autonomous and Intelligent Systems* (2012), p. 59–68.
21. T. Joseph, T. Fischer, M. Milford, Trajectory Tracking via Multiscale Continuous Attractor Networks, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2023), p. 368–375, doi:10.1109/IROS55552.2023.10341938.

22. A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, A. Maida, Deep learning in spiking neural networks. *Neural Networks* **111**, 47–63 (2019), doi:<https://doi.org/10.1016/j.neunet.2018.12.002>.
23. J. K. Eshraghian, *et al.*, Training Spiking Neural Networks Using Lessons From Deep Learning. *Proceedings of the IEEE* **111** (9), 1016–1054 (2023), doi:[10.1109/jproc.2023.3308088](https://doi.org/10.1109/jproc.2023.3308088).
24. S. Hussaini, M. Milford, T. Fischer, Ensembles of Compact, Region-specific & Regularized Spiking Neural Networks for Scalable Place Recognition, in *IEEE International Conference on Robotics and Automation* (2023), p. 4200–4207, doi:[10.1109/icra48891.2023.10160749](https://doi.org/10.1109/icra48891.2023.10160749).
25. S. Hussaini, M. J. Milford, T. Fischer, Spiking Neural Networks for Visual Place Recognition via Weighted Neuronal Assignments. *IEEE Robotics and Automation Letters* **7** (2), 4094–4101 (2022).
26. S. Hussaini, M. Milford, T. Fischer, Applications of Spiking Neural Networks in Visual Place Recognition. *IEEE Transactions on Robotics* **41**, 518–537 (2025).
27. A. D. Hines, P. G. Stratton, M. Milford, T. Fischer, VPRTempo: A Fast Temporally Encoded Spiking Neural Network for Visual Place Recognition, in *IEEE International Conference on Robotics and Automation* (2024), p. 10200–10207, doi:<https://doi.org/10.48550/arXiv.2309.10225>.
28. N. S. Giraldo, S. Isaza, R. A. Velásquez, Sailboat navigation control system based on spiking neural networks. *Control Theory and Technology* pp. 489–504 (2023), doi:[10.1007/s11768-023-00150-1](https://doi.org/10.1007/s11768-023-00150-1), <https://doi.org/10.1007/s11768-023-00150-1>.
29. J. Liu, H. Lu, Y. Luo, S. Yang, Spiking neural network-based multi-task autonomous learning for mobile robots. *Engineering Applications of Artificial Intelligence* **104** (2021), doi:<https://doi.org/10.1016/j.engappai.2021.104362>, <https://www.sciencedirect.com/science/article/pii/S0952197621002104>.
30. G. Tang, A. Shah, K. P. Michmizos, Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2019), p. 4176–4181, doi:[10.1109/iros40897.2019.8967864](https://doi.org/10.1109/iros40897.2019.8967864).

31. F. Galluppi, *et al.*, Live Demo: Spiking ratSLAM: Rat hippocampus cells in spiking neural hardware. *IEEE Biomedical Circuits and Systems Conference* p. 91–91 (2012), doi:10.1109/BioCAS.2012.6418493.
32. G. qiang Bi, M. ming Poo, Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *Journal of Neuroscience* **18** (24), 10464–10472 (1998), doi:10.1523/jneurosci.18-24-10464.1998.
33. M. Chancán, L. Hernandez-Nunez, A. Narendra, A. B. Barron, M. Milford, A Hybrid Compact Neural Architecture for Visual Place Recognition. *IEEE Robotics and Automation Letters* **5** (2), 993–1000 (2020), doi:10.1109/LRA.2020.2967324.
34. C. Bartolozzi, G. Indiveri, E. Donati, Embodied neuromorphic intelligence. *Nature Communications* **13** (1) (2022), doi:10.1038/s41467-022-28487-2.
35. G. Gallego, *et al.*, Event-Based Vision: A Survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **44** (01), 154–180 (2022), doi:10.1109/TPAMI.2020.3008413.
36. N. Russo, H. Huang, E. Donati, T. Madsen, K. Nikolic, An Interface Platform for Robotic Neuromorphic Systems. *Chips* **2** (1), 20–30 (2023), doi:10.3390/chips2010002.
37. Y. Yang, C. Bartolozzi, H. H. Zhang, R. A. Nawrocki, Neuromorphic electronics for robotic perception, navigation and control: A survey. *Engineering Applications of Artificial Intelligence* **126** (2023), doi:https://doi.org/10.1016/j.engappai.2023.106838.
38. F. Akopyan, *et al.*, TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **34** (10), 1537–1557 (2015), doi:10.1109/tcad.2015.2474396.
39. J. Pei, *et al.*, Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* **572** (7767), 106–111 (2019).
40. B. J. Shastri, *et al.*, Principles of Neuromorphic Photonics. *Encyclopedia of Complexity and Systems Science* p. 83–118 (2018).

41. F. Paredes-Vallés, *et al.*, Fully neuromorphic vision and control for autonomous drone flight. *Science Robotics* **9** (90) (2024), doi:10.1126/scirobotics.adi0591, <https://www.science.org/doi/abs/10.1126/scirobotics.adi0591>.
42. R. Halaly, E. Ezra Tsur, Autonomous driving controllers with neuromorphic spiking neural networks. *Frontiers in Neurorobotics* **17** (2023), doi:10.3389/fnbot.2023.1234962, <https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2023.1234962>.
43. A. Linares-Barranco, F. Perez-Peña, A. Jimenez-Fernandez, E. Chicca, ED-BioRob: A Neuromorphic Robotic Arm With FPGA-Based Infrastructure for Bio-Inspired Spiking Motor Controllers. *Frontiers in Neurorobotics* **14** (2020), doi:10.3389/fnbot.2020.590163, <https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2020.590163>.
44. M. Yao, *et al.*, Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications* **15** (1) (2024).
45. T. Fischer, M. Milford, Event-Based Visual Place Recognition With Ensembles of Temporal Windows. *IEEE Robotics and Automation Letters* **5** (4), 6924–6931 (2020).
46. M. J. Milford, G. F. Wyeth, SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights, in *IEEE International Conference on Robotics and Automation* (2012), p. 1643–1649, doi:10.1109/icra.2012.6224623.
47. T. Fischer, M. Milford, How Many Events do You Need? Event-based Visual Place Recognition Using Sparse But Varying Pixels. *IEEE Robotics and Automation Letters* **7** (4), 12275–12282 (2022), doi:10.1109/LRA.2022.3216226.
48. A. J. Lee, A. Kim, EventVLAD: Visual Place Recognition with Reconstructed Edges from Event Cameras, in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2021), pp. 2247–2252, doi:10.1109/IROS51168.2021.9635907.

49. H. Lee, H. Hwang, Ev-ReconNet: Visual Place Recognition Using Event Camera With Spiking Neural Networks. *IEEE Sensors Journal* **23** (17), 20390–20399 (2023), doi:10.1109/JSEN.2023.3298828.
50. P. Lichtsteiner, C. Posch, T. Delbruck, A 128× 128 120 dB 15 μ s Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits* **43** (2), 566–576 (2008), doi:10.1109/JSSC.2007.914337.
51. S. Bian, *et al.*, ColibriUAV: An Ultra-Fast, Energy-Efficient Neuromorphic Edge Processing UAV-Platform with Event-Based and Frame-Based Cameras, in *IEEE International Workshop on Advances in Sensors and Interfaces* (2023), p. 287–292.
52. E. Kelasidi, K. Y. Pettersen, J. T. Gravdahl, Energy efficiency of underwater robots. *IFAC Conference on Manoeuvring and Control of Marine Craft* **48** (16), 152–159 (2015), doi: <https://doi.org/10.1016/j.ifacol.2015.10.273>.
53. N. Kashiri, *et al.*, An Overview on Principles for Energy Efficient Robot Locomotion. *Frontiers in Robotics and AI* **5** (2018), doi:10.3389/frobt.2018.00129, <https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2018.00129>.
54. R. Islam, H. Habibullah, A Semantically Aware Place Recognition System for Loop Closure of a Visual SLAM System, in *2021 4th International Conference on Mechatronics, Robotics and Automation (ICMRA)* (2021), pp. 117–121, doi:10.1109/ICMRA53481.2021.9675715.
55. R. Mur-Artal, J. M. M. Montiel, J. D. Tard'os, ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics* **31** (5), 1147–1163 (2015), doi:10.1109/TRO.2015.2463671.
56. S. Yang, *et al.*, Visual SLAM Based on Semantic Segmentation and Geometric Constraints for Dynamic Indoor Environments. *IEEE Access* **10**, 69636–69649 (2022), doi:10.1109/ACCESS.2022.3185766.
57. G. B. Nair, M. Milford, T. Fischer, Enhancing Visual Place Recognition via Fast and Slow Adaptive Biasing in Event Cameras, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2024).

58. Y. Wu, *et al.*, Brain-inspired global-local learning incorporated with neuromorphic computing. *Nature Communications* **13** (1) (2022).
59. R. Zhu, *et al.*, Online dynamical learning and sequence memory with neuromorphic nanowire networks. *Nature Communications* **14** (1) (2023).
60. A. Rostami, B. Vogginger, Y. Yan, C. G. Mayr, E-prop on SpiNNaker 2: Exploring online learning in spiking RNNs on neuromorphic hardware. *Frontiers in Neuroscience* **16** (2022), doi:10.3389/fnins.2022.1018006.
61. S. Hwang, *et al.*, ReplaceNet: real-time replacement of a biological neural circuit with a hardware-assisted spiking neural network. *Frontiers in Neuroscience* **17** (2023), doi:10.3389/fnins.2023.1161592.
62. T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A Simple Framework for Contrastive Learning of Visual Representations, in *International Conference on Machine Learning*, vol. 119 (2020), p. 1597–1607, <https://proceedings.mlr.press/v119/chen20j.html>.
63. J. Dupeyroux, J. R. Serres, S. Viollet, AntBot: A six-legged walking robot able to home like desert ants in outdoor environments. *Science Robotics* **4** (27) (2019), doi:10.1126/scirobotics.aau0307, <https://www.science.org/doi/abs/10.1126/scirobotics.aau0307>.
64. X. Sun, Q. Fu, J. Peng, S. Yue, An insect-inspired model facilitating autonomous navigation by incorporating goal approaching and collision avoidance. *Neural Networks* **165**, 106–118 (2023), doi:<https://doi.org/10.1016/j.neunet.2023.05.033>, <https://www.sciencedirect.com/science/article/pii/S0893608023002782>.
65. M. Milford, G. Wyeth, D. Prasser, RatSLAM: a hippocampal model for simultaneous localization and mapping, in *IEEE International Conference on Robotics and Automation* (2004), pp. 403–408, doi:10.1109/ROBOT.2004.1307183.
66. N. S.-Y. Dumont, P. M. Furlong, J. Orchard, C. Eliasmith, Exploiting semantic information in a spiking neural SLAM system. *Frontiers in Neuroscience* **17** (2023), doi:10.3389/fnins.2023.1190515, <https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2023.1190515>.

67. T. Zeng, F. Tang, D. Ji, B. Si, NeuroBayesSLAM: Neurobiologically inspired Bayesian integration of multisensory information for robot navigation. *Neural Networks* **126**, 21–35 (2020), doi:<https://doi.org/10.1016/j.neunet.2020.02.023>, <https://www.sciencedirect.com/science/article/pii/S0893608020300770>.
68. L. Liu, J. Chen, M. Brocanelli, W. Shi, E2M: an energy-efficient middleware for computer vision applications on autonomous mobile robots, in *ACM/IEEE Symposium on Edge Computing* (2019), p. 59–73, doi:10.1145/3318216.3363302.
69. P. G. Stratton, A. Wabnitz, C. Essam, A. Cheung, T. J. Hamilton, Making a Spiking Net Work: Robust brain-like unsupervised machine learning. *arXiv 2208.01204* (2022).
70. W. Guo, M. E. Fouda, A. M. Eltawil, K. N. Salama, Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems. *Frontiers in Neuroscience* **15** (2021), doi:10.3389/fnins.2021.638474, <https://www.frontiersin.org/articles/10.3389/fnins.2021.638474>.
71. S. Garg, M. Vankadari, M. Milford, SeqMatchNet: Contrastive Learning with Sequence Matching for Place Recognition & Relocalization, in *Conference on Robot Learning* (2022).

Acknowledgments

The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. We wish to acknowledge the support of the Research Engineering Facility (REF) team at QUT for the provision of engineering support, expertise and research infrastructure in enablement of this project. We also would like to thank Dr Scarlett Raine and Dr Somayeh Hussaini for their valuable feedback during initial manuscript preparation. Finally, we would like to thank the organisers and participants of the 2022 Lifelong Learning at Scale topic area at the Telluride Neuromorphic Workshop for the insightful discussions and inspiring environment, which helped shape some of the ideas explored in this paper.

Funding: This work received funding from an ARC Laureate Fellowship FL210100156 to MM, AUSMURIB000001 associated with ONR MURI grant N00014191-2571 to MM and TF, and an

ARC Discovery Early Career Researcher Award DE240100149 to TF.

Author contributions: A.D.H. & T.F. conceptualized and designed the experiments; A.D.H. performed and analyzed all experiments; T.F. & M.M. provided general input into the research; T.F. & M.M. provided funding for the project; A.D.H. wrote the paper; A.D.H., M.M. & T.F. edited the paper.

Competing interests: The authors declare no competing interests.

Data and materials availability: All data and code used in this study is available at <https://doi.org/10.5281/zenodo.15392412>.

Supplementary materials

Supplementary Figures S1-S8

Supplementary Tables S1-S2

Supplementary Movie S1

Supplementary Materials for

A compact neuromorphic system for ultra-energy-efficient, on-device robot localization

Adam D Hines^{1*}, Michael Milford¹, Tobias Fischer¹

¹QUT Centre for Robotics, Queensland University of Technology, 2 George St, Brisbane, QLD 4001,
Australia

*Corresponding author. Email: adam.hines@qut.edu.au

This PDF file includes:

Supplementary Figures S1-S8

Supplementary Tables S1-S2

Other Supplementary Materials for this manuscript:

Supplementary Movie S1

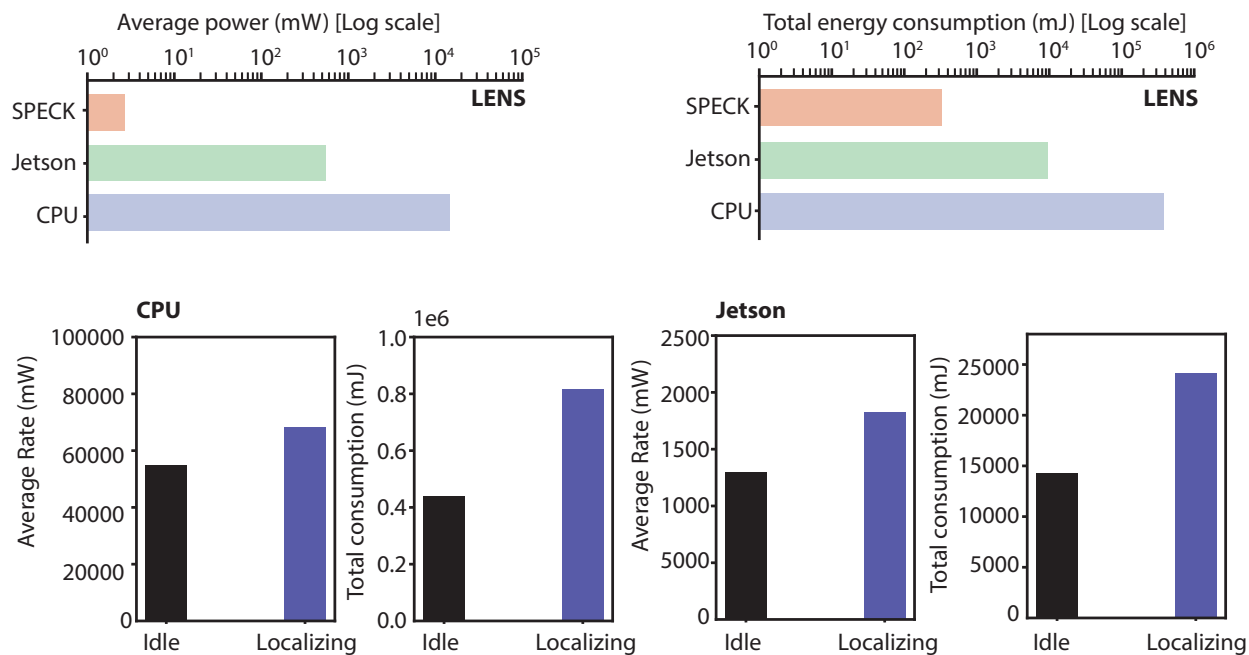


Figure S1: Power and energy measurements for running LENS on von Neumann hardware. The baseline power consumption and energy was subtracted from the localization measurements for our comparisons to other hardware.

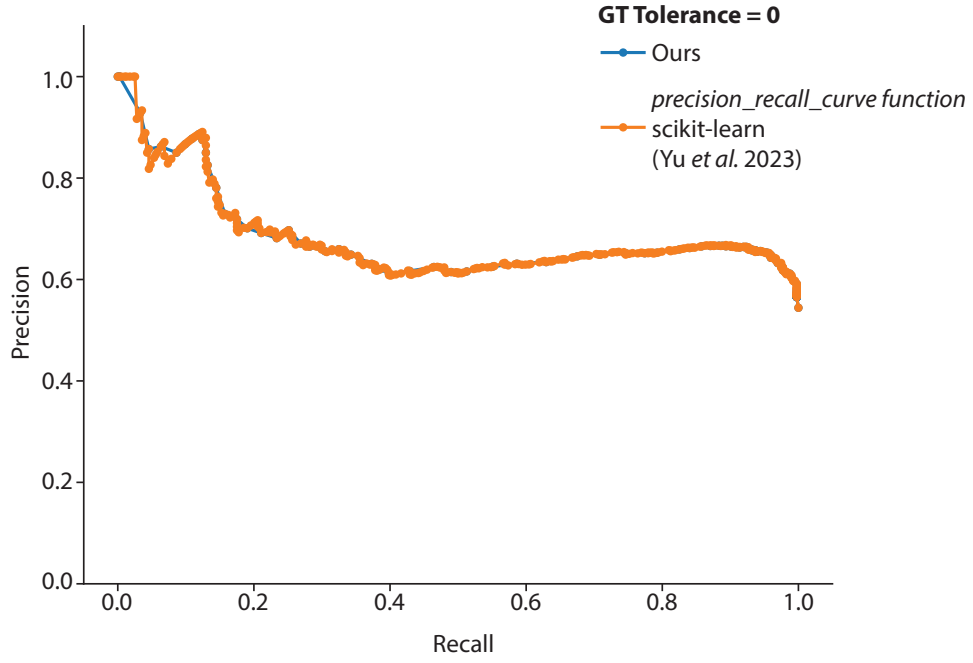


Figure S2: Comparison of Precision-Recall calculation without ground truth tolerance. PR curves for the Brisbane Event-VPR dataset generated by our pipeline (blue) align closely with those generated using the scikit-learn `precision_recall_curve` function (orange) employed by Yu et al. (12). The minor deviation is due to discretization artefacts, as our method uses 100 points while the approach taken by Yu et al. uses 398 points to compute the curves.

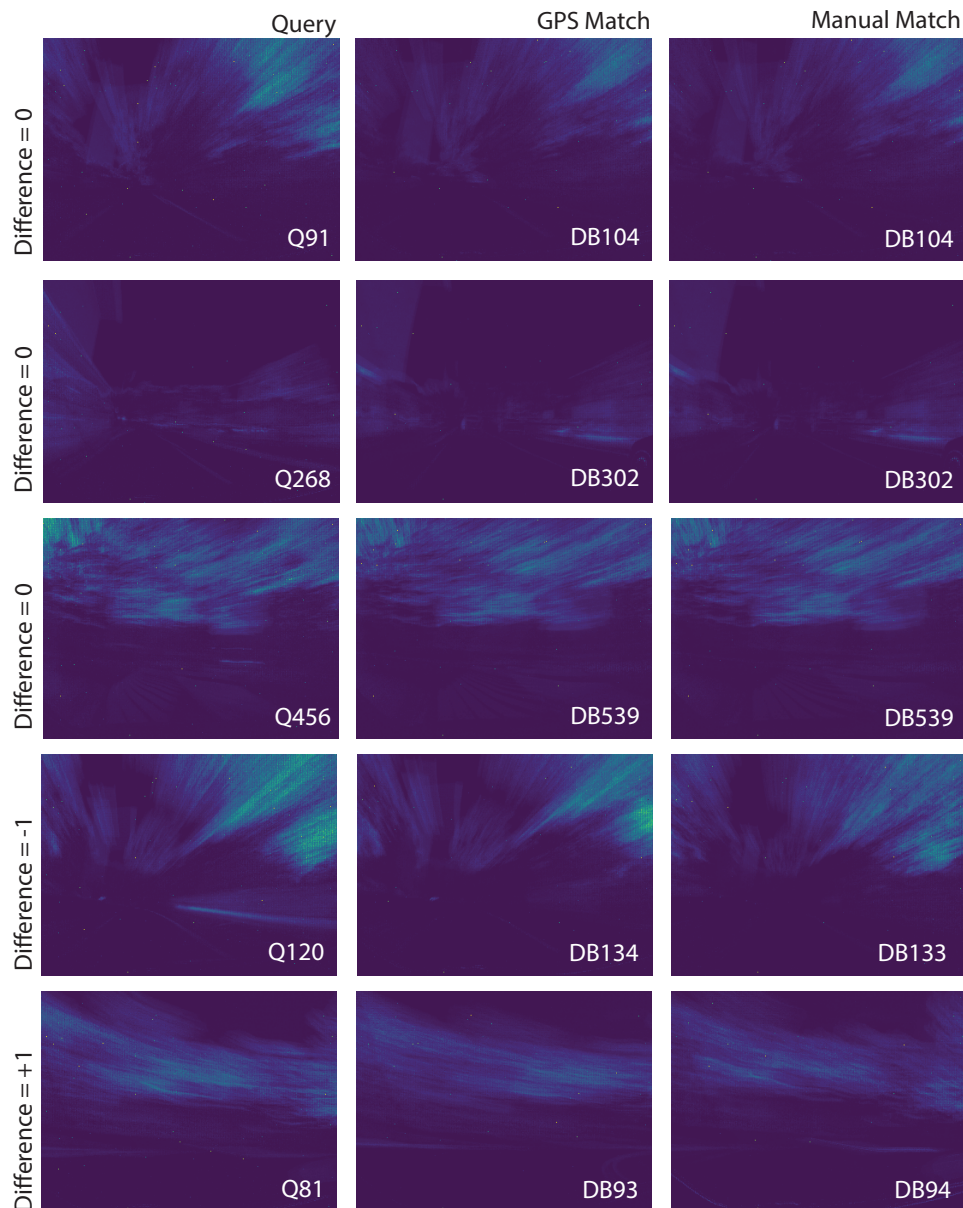


Figure S3: Comparison of GPS and human-labeled ground truth for queries and their matching references. (Left) Shows the query images with an inset of their place index for the Brisbane Event-VPR dataset. (Middle) The matching reference place using the GPS pseudo ground truth and the (Right) actual match confirmed by manual ground truth. In the cases where Difference = 0, the GPS pseudo ground truth matches exactly to the manually labeled correct image. In cases where Difference= ± 1 , the GPS match was offset by one image of the true match, which is equivalent to approximately 14 m. The purpose of the ground truth tolerances is to allow such neighboring images to count as correct matches, accounting for such slight misalignments.

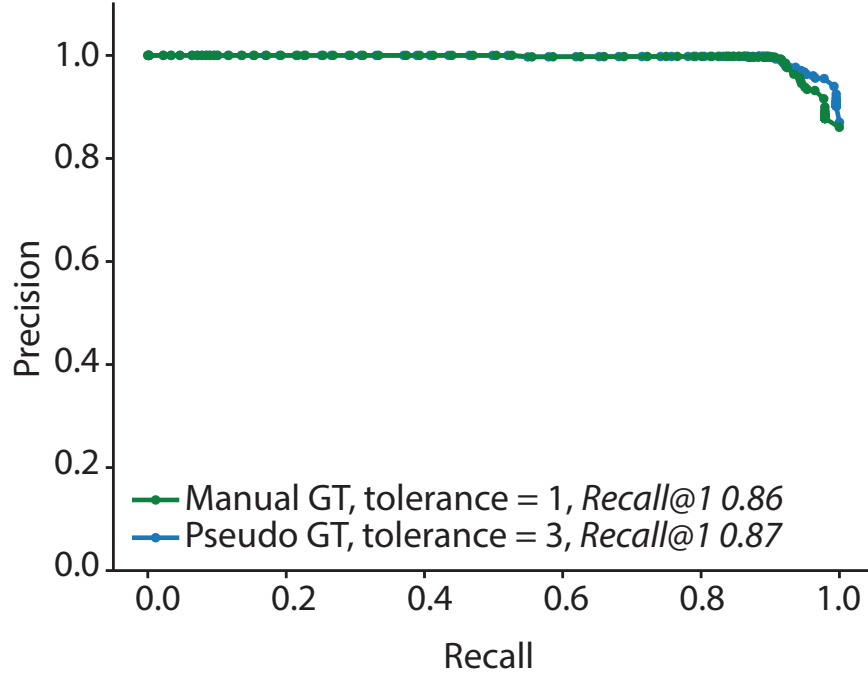


Figure S4: Precision-recall evaluation comparing GPS pseudo ground truth with manually aligned ground truth. For the Brisbane Event-VPR dataset, we compare the GPS pseudo ground truth (GT) with the manually annotated ground truth using tolerances of 1 and 3 places representing approximately 14 m and 42 m, respectively. Both performed very similarly with a Recall@1 of 0.86 and 0.87 for the manual and GPS pseudo ground truth, respectively. The slight improvement in performance observed with the pseudo ground truth can be attributed to the higher tolerance threshold of 42 m capturing 7 additional matches, which extends beyond the 14 m limit used for manual ground truth correction.

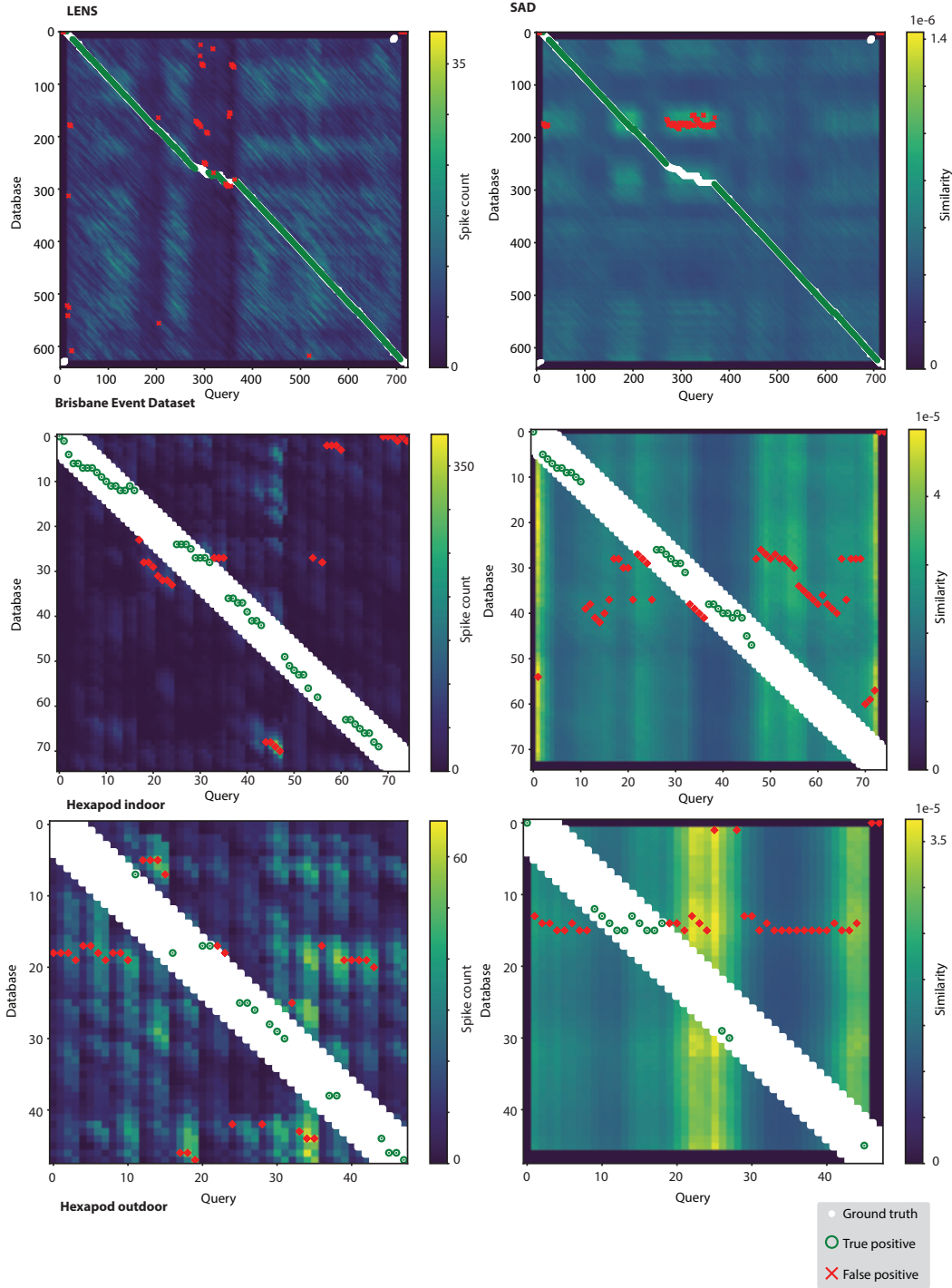


Figure S5: Distribution of true and false positives for similarity matrices. Similarity matrices for (left) LENS and (right) SAD for the 3 datasets evaluated from top to bottom - Brisbane Event Dataset, hexapod indoor, and hexapod outdoor. Matrices show the ground truth used (white dots) with true positives (green circles) and false negatives (red crosses).

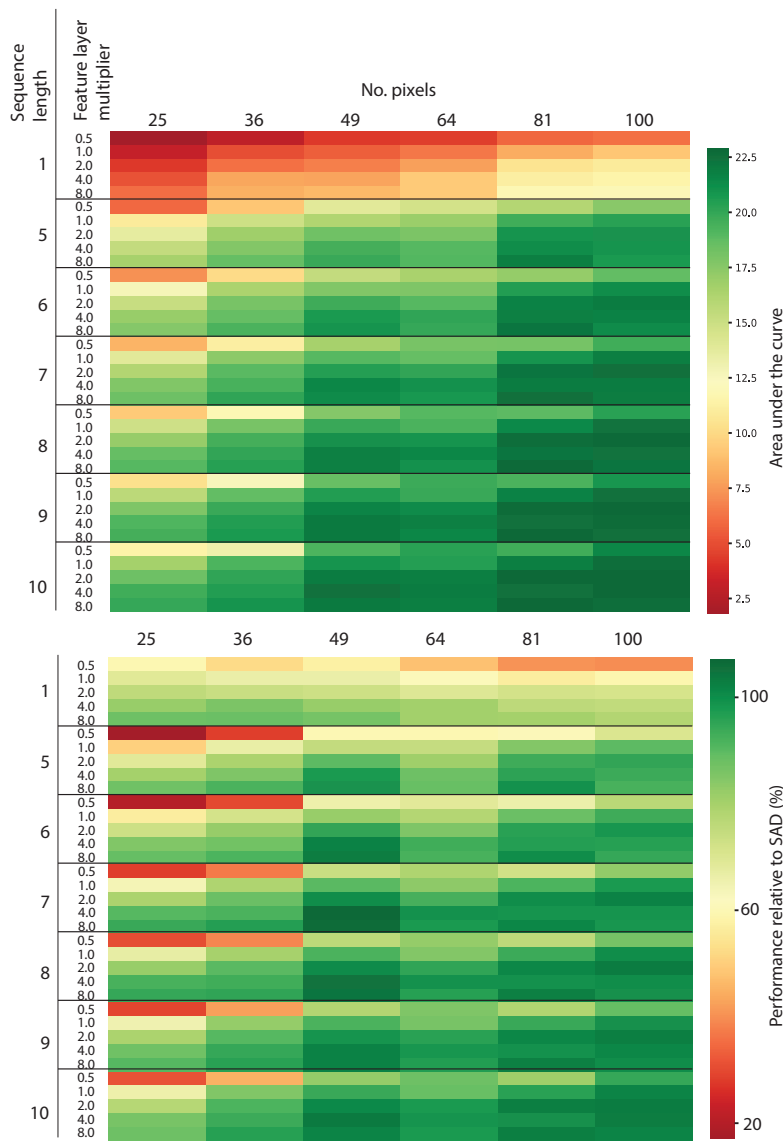


Figure S6: Effects of model size and sequence length on network performance. (Top) Heatmap of the area under the curve of the Recall@N results from training and testing the Brisbane Event VPR data (45) for different number of input pixels, feature layer size multiplier (relative to number of pixels), and sequence lengths. Our model with $49 \times 49 \times 641$ neurons with a sequence length of 10 was the highest performer whilst still being able to be deployed to Speck™. The best performing model had the architecture of $100 \times 200 \times 641$ neurons and a sequence length of 10, indicating that model accuracy heavily depends on the number of input events the ratio of the feature layer to the input. (Bottom) Performance of the various models relative to sum of absolute differences for the area under the curve. Smaller number of input pixels with larger feature layer sizes can perform as well as or better than a larger number of input pixels with the same feature layer size. The important consideration is that increasing the feature layer size greatly increases the model size, which may limit deployment on memory constrained hardware.

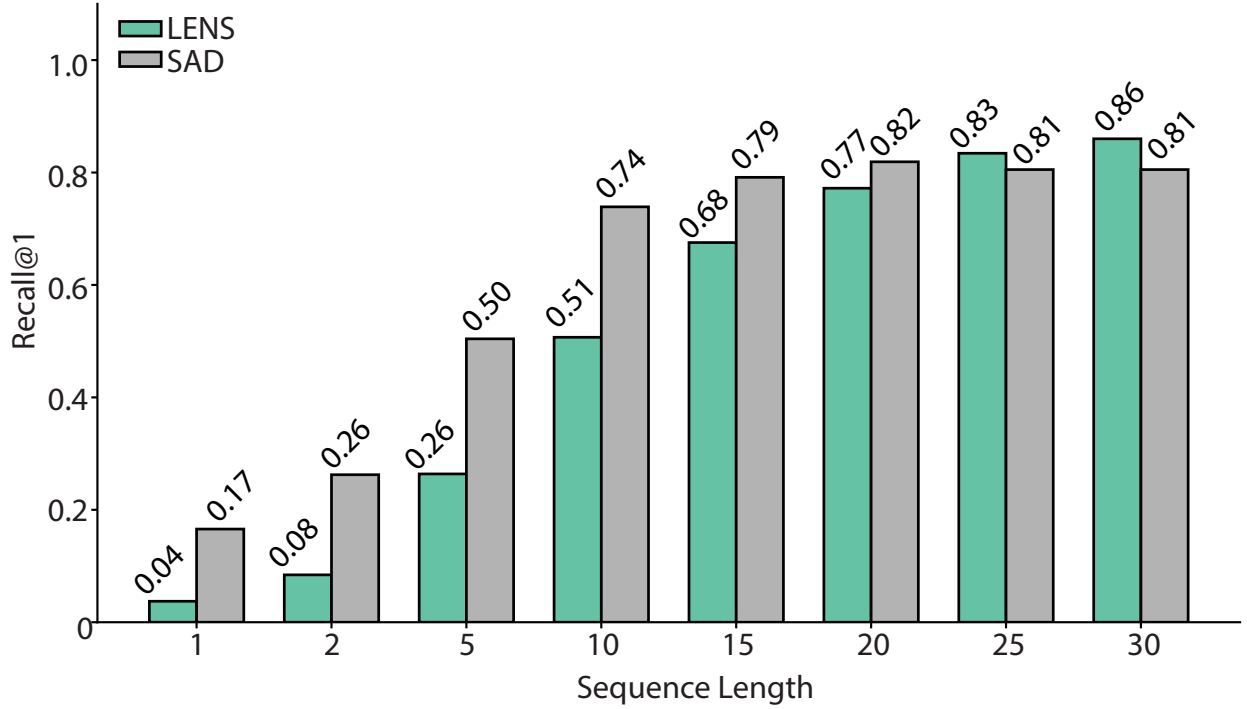


Figure S7: Effect of sequence length on Recall@1 performance. An ablation study was conducted on the Brisbane Event-VPR dataset observing the effects of sequence length for Recall@1 performance between our system (LENS) and the Sum-of-Absolute-Differences (SAD) baseline method. We used the manually generated ground truth with a tolerance of +/- 1 place (approximately 14 m) for this investigation.

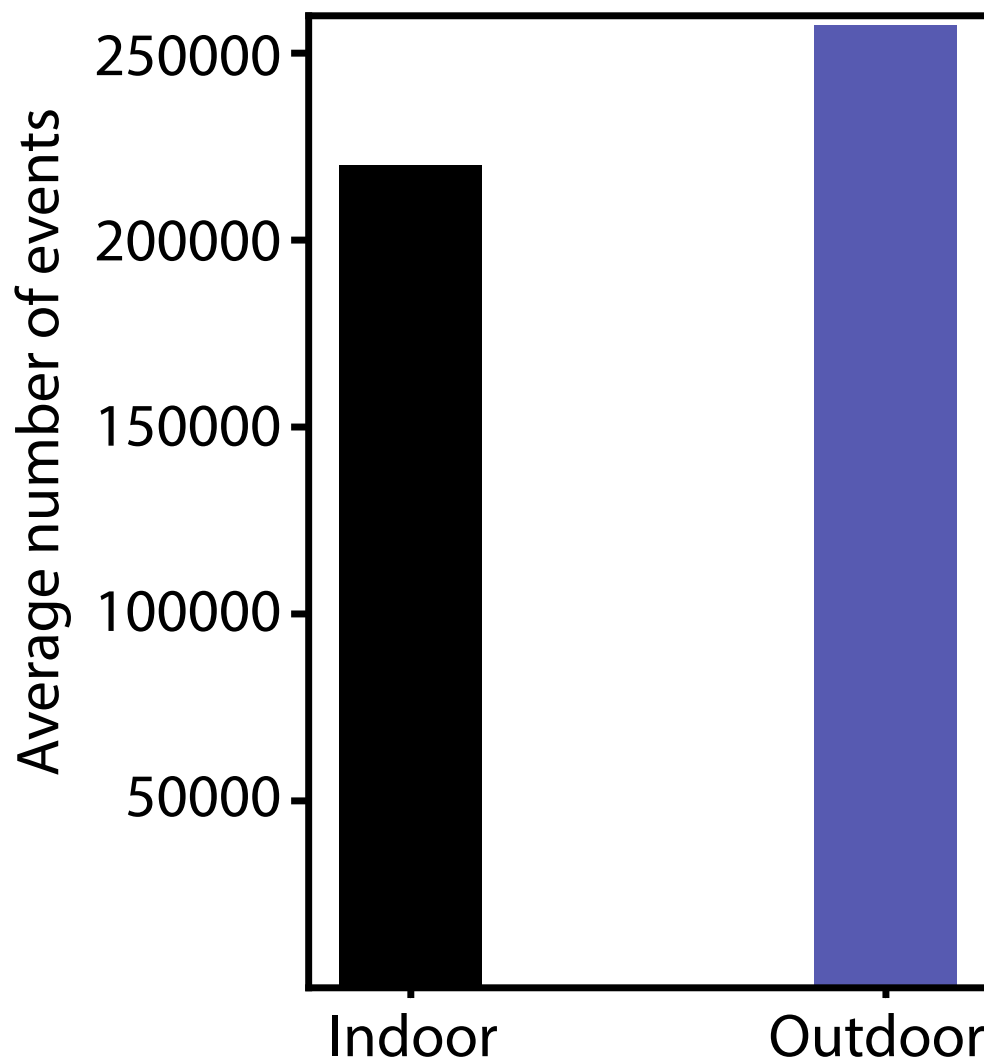


Figure S8: Average number of events between indoor and outdoor traversals.

Table S1: Hyperparameters for network training

Parameter	$\theta_{\max}^{I \rightarrow F}$	$\theta_{\max}^{F \rightarrow O}$	$\eta_{\text{STDP}}^{\text{init}}$	$\eta_{\text{ITP}}^{\text{init}}$	$f_{\min}^{I \rightarrow F}, f_{\max}^{I \rightarrow F}$	$f_{\min}^{F \rightarrow O}, f_{\max}^{F \rightarrow O}$
Values	0.75	0.5	0.01	0.02	[0.4, 0.6]	[0.5, 0.5]
Parameter	$p_{\text{exc}}^{I \rightarrow F}$	$p_{\text{inh}}^{I \rightarrow F}$	$p_{\text{exc}}^{F \rightarrow O}$	$p_{\text{inh}}^{F \rightarrow O}$	$\epsilon^{I \rightarrow F}$	$\epsilon^{F \rightarrow O}$
Values	0.35	0.75	1.0	1.0	64	128

Table S2: Energy consumption for SAD with fewer pixels

Pixel count	100	64	1
Energy CPU (mJ)	61,427.0	46943.18	25698.37
Energy Jetson (mJ)	2,968.0	531.81	474.88

Caption for Supplementary Movie S1. LENS display during localization. The left side shows the main display of LENS during online localization deployed on a robotic platform. The Visualizer shows the field of view of the on-board dynamic vision sensor with the power being actively consumed dynamically plotted underneath. The right side shows the output of the LENS matching system which returns the place matching result after collecting output spikes from the model over the specific time window.