Fully Autonomous Neuromorphic Navigation and Dynamic Obstacle Avoidance (Supplementary Material)

Xiaochen Shang[1], Pengwei Luo[1], Xinning Wang[1], Jiayue Zhao[1], Huilin Ge[2], Bo Dong[3], Xin Yang[1,*]
[1]Key Laboratory of Social Computing and Cognitive Intelligence (Dalian University of Technology), Minstry of Education,
[2]Jiangsu University of Science and Technology, [3]Cephia AI
xcshang0614@gmail.com
xinyang@dult.edu.cn

Contents

Supplementary Note 1: Spiking Convolutional Neural Networks **3**
Spiking Neural Network (SNN) 3
Spiking Convolutional Neural Network 4

Supplementary Note 2: Mathematical Proofs for Event RF Model **4**

Supplementary Note 3: Biological Concepts **5**
Visual-homing 5
Frog-eye Visual Mechanism 6

Supplementary Note 4: Additional Details on Dynamic Obstacle Avoidance Accuracy Calculation **7**
Event Camera Processing 7
Error Metrics 8
Centroid Detection Accuracy Across Modalities and Obstacle Sizes 8

Supplementary Note 5: Obstacles in Real-world Experiment **9**
Different Obstacles 9
Background Dynamic Obstacles 11

Supplementary Note 6: Training Details of Calibration Spiking Convolutional Neural Network **11**
Network Structure 11
Implementation Details 11

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Evaluation Metrics 11
Outdoor Scenarios Performance 12

Supplementary Note 7: Additional Details on Simulation Experiments **13**
Experimental Setup 13
Navigation Experiments 13
Combined Task Simulation 13

Supplementary Note 8: Analysis of IMU Error **15**

Supplementary Note 9: Additional Details on Experiments Under Different Light Conditions **17**

Supplementary Note 10: Energy Consumption **18**

Supplementary Note 11: Computational Cost **19**

Supplementary Note 12: Analysis of RF Model **20**
Mathematical Analysis 20
Parameter Sensitivity Analysis 21

Supplementary Note 1: Spiking Convolutional Neural Networks

Spiking Neural Network (SNN)

Various models for spiking neurons mathematically describe the properties of a cell in the nervous system with varying degrees of detail. Normally, three conditions are considered by these models: resting, depolarization, and hyperpolarization. When a neuron is resting, it maintains a constant membrane potential. The change in membrane potential can be either a decrease or an increase relative to the resting potential. An increase in the membrane potential is called depolarization, which enhances a cell's ability to generate an action potential; it is excitatory. In contrast, hyperpolarization describes a reduction in the membrane potential, which makes the associated cell less likely to generate an action potential, and, as such, is inhibitory. All inputs and outputs of a spiking neuron model are sequences of spikes.

A sequence of spikes is called a spike train and defined as $s(t) = \Sigma_{t(f)\in F}\delta(t - t^{(f)})$, where F represents the set of times at which the individual spikes occur [1]. Typical spiking neuron models set the resting potential to 0. However, existing models achieve depolarization and hyperpolarization in substantially different ways. In the following, we review two commonly used models: the spike response model (SRM) [2] and leaky integrate-and-fire (LIF) model [3].

Spike Response Model (SRM) An SRM first converts an incoming spike train si(t) into a spike response signal as $(\varepsilon * s_i)(t)$, where $\varepsilon(\cdot)$ is a spike response kernel. Then, the generated spike response signal is scaled by a synaptic weight w_i. Depolarization is achieved by summing all the scaled spike response signals: $\Sigma_i w_i(\varepsilon * s_i)(t)$. When incoming spike trains trigger a spike $s(t)$, the SRM models hyperpolarization by defining a refractory potential as $(\zeta * s)(t)$, where $\zeta(\cdot)$ is a refractory kernel. With an SRM, a feedforward SNN architecture with n_l layers can be defined. Given N^l incoming spike trains at layer l, $s_i^l(t)$, the forward propagation process of the network is mathematically defined as follows [1, 2]:

$$v_i^{l+1}(t) = \sum_{j=1}^{N^l} w_{ij}(\varepsilon * s_j^l)(t) + (\zeta * s_i^{l+1})(t-1), \tag{1}$$

$$s_i^{l+1}(t) = f_s(v_i^{l+1}(t)), \tag{2}$$

$$f_s(v) : v \to s, s(t) := s(t) + \delta(t - t^{(f+1)}), \tag{3}$$

$$t^{f+1} = min\{t : v(t = \Theta, t > t^{(f)}\} \tag{4}$$

where $f_s(\cdot)$ is a spike function and Θ is the membrane potential threshold, which is static and the same for all neurons in the network. This static threshold is the one that we replace with the proposed dynamic threshold.

Leaky Integrate-and-Fire (LIF) An LIF model is a simplified variant of an SRM. This scheme directly processes incoming spike trains and ignores the spike response kernel. Hyperpolarization is achieved by a simple decay function $f_d(\cdot)$. The forward propagation process of the network can be defined as:

$$v_i^{l+1}(t) = \sum_{j=1}^{N^l} w_{ij}s_j^l(t) + v_i^{l+1}(t-1)f_d(s_i^{l+1}(t-1)) + b_i^{l+1}, \tag{5}$$

$$s_i^{l+1}(t) = f_s(v_i^{l+1}(t)), \tag{6}$$

$$f_d(s(t)) = \begin{cases} D & s(t) = 0 \\ 0 & s(t) = 1 \end{cases} \tag{7}$$

where b_i^{l+1} is an adjustable bias that is learned to mimic a dynamic threshold behavior. However, the biases of this model are static during forwarding propagation. In contrast, the proposed dynamic threshold is dynamic and automatically adapts to membrane potentials.

Spiking Convolutional Neural Network

Spiking Convolutional Neural Networks combine the feature extraction capabilities of CNNs with the temporal processing of spiking neurons. The network processes input spike trains through convolutional layers while maintaining biological plausibility. Each neuron's state evolves according to the membrane potential dynamics:

$$\tau_m \frac{dV_{xy}^l(t)}{dt} = -(V_{xy}^l(t) - V_{\text{rest}}) + I_{xy}^l(t) \tag{8}$$

where $V_{xy}^l(t)$ represents the membrane potential of neuron at position (x, y) in layer l, τ_m is the membrane time constant, and $I_{xy}^l(t)$ is the synaptic current from presynaptic spikes:

$$I_{xy}^l(t) = \sum_{i,j,k} W_{ijk}^l \cdot S_{x+i,y+j,k}^{l-1}(t) + b^l \tag{9}$$

The network employs a dynamic threshold mechanism that adapts based on recent firing activity:

$$\Theta_{xy}^l(t) = \Theta_0 + \alpha \sum_{t'=t-\Delta t}^{t} S_{xy}^l(t') \tag{10}$$

When $V_{xy}^l(t) \geq \Theta_{xy}^l(t)$, the neuron emits a spike and enters a refractory period:

$$S_{xy}^l(t) = \begin{cases} 1 & \text{if } V_{xy}^l(t) \geq \Theta_{xy}^l(t) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The SCNN architecture typically consists of:

- Convolutional layers with spike-triggered synaptic currents
- Temporal max-pooling layers that operate on spike counts
- Fully-connected layers for final classification

The forward propagation can be summarized as:

$$V^{l+1}(t) = W^l * S^l(t) + b^l + V^{l+1}(t-1) \odot (1 - S^{l+1}(t-1)) \tag{12}$$

$$S^{l+1}(t) = H(V^{l+1}(t) - \Theta^{l+1}(t)) \tag{13}$$

where $*$ denotes convolution, $\odot$ is element-wise multiplication, and $H(\cdot)$ is the Heaviside step function. The network's temporal dynamics enable efficient processing of spatio-temporal patterns while maintaining the energy efficiency characteristic of spiking neural networks.

Supplementary Note 2: Mathematical Proofs for Event RF Model

Lemma 1 (Static Event Suppression)**.** *For static events e_s with $\|\nabla I(x_s, y_s)\| = 0$, when the parameters satisfy:*

$$\frac{A_1}{A_2} = e^{-\Delta t/\tau} \quad \textit{and} \quad E_{th} = I_{th} \tag{14}$$

the Event RF model output is bounded by:

$$F(e_s) \leq \epsilon \quad \textit{where} \quad \epsilon = \sup_{t>0} |A_1 - A_2 e^{\Delta t/\tau}| G_{max} \tag{15}$$

Proof. For static events, the temporal kernels exhibit symmetry:

$$\begin{aligned} K(t, \tau_e) &= K(t - \Delta t, \tau_i) \quad \text{when} \quad \tau_e = \tau_i = \tau \\ F(e_s) &= \min(A_1 K(t, \tau) G_s, E_{th}) - \min(A_2 K(t - \Delta t, \tau) G_s, I_{th}) \\ &= \min(c_1, E_{th}) - \min(c_2, I_{th}) \quad (\text{where } c_i = A_i K(\cdot) G_s) \end{aligned}$$

Under the parameter conditions:

$$c_1 - c_2 = A_1 e^{-t/\tau} G_s - A_2 e^{-(t-\Delta t)/\tau} G_s = (A_1 - A_2 e^{\Delta t/\tau}) e^{-t/\tau} G_s = 0$$

Threshold clipping ensures $F(e_s) \leq \epsilon$ even with small asymmetries. □

Lemma 2 (Dynamic Event Enhancement)**.** *For dynamic events e_d with $\frac{d}{dt}\|\nabla I(x_d, y_d)\| > 0$, there exists $\delta > 0$ such that:*

$$F(e_d) \geq G_d \cdot \left(A_1 e^{-t/\tau_e} - A_2 e^{-(t-\Delta t)/\tau_i} \right) \geq \delta \tag{16}$$

when $\tau_e < \tau_i$ (faster ERF decay than IRF).

Proof. The motion-induced temporal variation breaks the symmetry:

$$\begin{aligned} \frac{d}{dt} K(t, \tau_e) &= -\frac{1}{\tau_e} e^{-t/\tau_e} \\ \frac{d}{dt} K(t - \Delta t, \tau_i) &= -\frac{1}{\tau_i} e^{-(t-\Delta t)/\tau_i} \end{aligned}$$

For $\tau_e < \tau_i$, the ERF term decays faster, creating a temporal window where:

$$A_1 e^{-t/\tau_e} - A_2 e^{-(t-\Delta t)/\tau_i} > 0 \quad \text{for} \quad t \in (t_0, t_0 + \Delta t)$$

Integrating over this window yields the lower bound δ. □

Theorem 1 (Dynamic Obstacle Selectivity)**.** *The Event RF model satisfies:*

$$\mathbb{E}[F(e_s)] \leq \epsilon \quad \textit{(static suppression)} \tag{17}$$

$$\mathbb{E}[F(e_d)] \geq \delta \quad \textit{(dynamic enhancement)} \tag{18}$$

with signal-to-noise ratio improvement:

$$\gamma = \frac{\mathbb{E}[F(D)]}{\mathbb{E}[F(S)]} \geq \frac{\delta}{\epsilon} \gg 1 \tag{19}$$

Proof. Combining Lemmas 1 and 2, the SNR gain follows from:

$$\gamma = \frac{\int_{t_0}^{t_0+\Delta t} F(e_d) dt}{\int_0^T F(e_s) dt} \geq \frac{\delta \cdot \Delta t}{\epsilon \cdot T} \quad \text{for observation window } T$$

□

Supplementary Note 3: Biological Concepts

Visual-homing

Visual homing in insects such as ants, bees, and wasps involves the integration of visual in formation with compass-based orientation. These insects employ a strategy of storing multiple views of their nest and surrounding landmarks from different vantage points during departure [4, 5, 6]. The "turn back and look" behavior suggests they actively acquire visual snapshots that are later used for navigation. During homing, insects compare current visual input with memorized views to determine movement directions, though the exact matching mechanism remains unclear. Two distinct visual processing modes are observed: at long distances, insects rely on panoramic landmark configurations (e.g., trees or rocks in specific directions), while near the nest, they focus on precise visual features directly associated with the goal location. The view-matching process appears orientation-specific,

requiring objects to be perceived at the same retinal angle as during learning, potentially achieved through body rotations that shift the retinal image. Interestingly, insects demonstrate the capacity to associate specific movements with sensory contexts, enabling complex maze navigation, and may employ top-down attentional processes during visual processing.

Mammals including birds, rats, and primates exhibit similar visual homing capabilities but with enhanced processing mechanisms. While they share the insect-like vulnerability to landmark displacement errors, mammals possess additional neural strategies such as ocular saccades and attentional spotlighting that allow sequential analysis of scene components [7]. This enables both global scene recognition (possibly through low spatial frequency processing) and detailed local feature analysis. Neurobiological studies reveal that mammalian navigation involves specialized hippocampal circuits, with place cells in CA3/CA1 regions encoding specific spatial locations, as shown in Fig 1 [8] Hippocampal lesions impair spatial learning in novel environments while preserving familiar routes, mirroring the declarative memory deficits seen in human hippocampal damage cases [9]. These findings suggest that while insects and mammals share fundamental sensory-motor learning principles for visual homing, mammals have evolved more sophisticated neural architectures that generalize across both spatial and non-spatial memory domains, with the hippocampus serving as a critical hub for spatiotemporal information integration in cognitive processes.

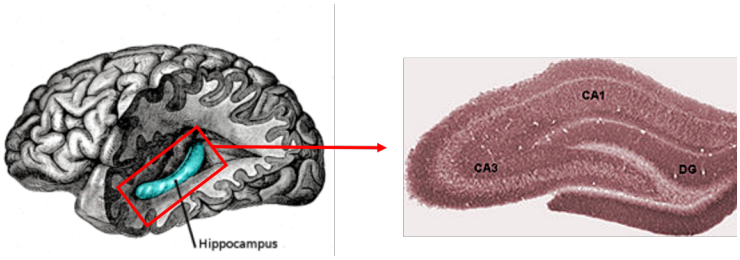


Figure 1: **Hippocampus and CA3/CA1 regions**

Frog-eye Visual Mechanism

The visual system's fundamental processing unit, the receptive field, comprises all photoreceptors that directly or indirectly stimulate a specific neural cell. As the basic feature extraction unit throughout visual processing hierarchies, receptive fields exhibit distinct properties at different levels of the visual pathway, progressively filtering out non-essential information to extract key features for perception and recognition tasks. Receptive fields are classified into classical and non-classical types based on their spatial organization and functional characteristics, with mutual modulation occurring between these two categories. The unique "seeing-moving-but-not-stationary" visual behavior of frogs has inspired extensive research into their neurobiological mechanisms and computational modeling. Early investigations focused primarily on the neurophysiological aspects of frog visual systems, while post-1980s research incorporated neural network approaches to simulate frog visual behaviors. Notable contributions include Lee's retina-based neural network model[10], Nishio's two-dimensional electronic circuits for target detection[11]. Recent advances have seen limited but significant attempts to develop frog vision-inspired computational models for complex environments. However, the field still lacks comprehensive theoretical descriptions and mathematical formulations of frog visual properties in computer vision, particularly regarding motion target analysis models.

The dynamic nature of receptive fields represents a paradigm shift from viewing them as fixed neural structures to understanding them as temporally adaptive systems. Receptive field properties including size, center position, modulation intensity, and spatial frequency exhibit stimulus-dependent variations. Recent studies have increasingly emphasized these dynamic characteristics. Such biologically inspired receptive field models have found applications across image processing domains, including Ekvall's target detection method[12] and Perez's pattern recognition-enhancing neural networks[13]. Despite these advances, research specifically addressing motion vision's receptive field characteristics remains scarce compared to general visual system studies.

Frog retinas demonstrate remarkably efficient visual information processing and filtering capabilities. Retinal neurons exhibit distinct responses to ON (light onset) and OFF (light offset) stimuli through hyperpolarization and depolarization channels respectively. The T5 neurons primarily handle prey-

predator discrimination, with R3 ganglion cells serving as their major input source, as shown in Fig 2 Extensive research on R3 cells has revealed their ON-OFF characteristics, responding exclusively to transient light changes while remaining insensitive to static stimuli[14]. Some R3 cells additionally demonstrate selectivity for motion direction, convex edges, and contrast. The spatial-temporal properties of R3 cells' receptive fields explain their motion response patterns, typically featuring concentric excitatory (ERF) and inhibitory (IRF) regions. Findings by Hoshino have identified asymmetric R3 receptive field organizations where ERF and IRF centers are spatially offset, creating directionally selective response patterns. This asymmetric anisotropy makes R3 cells particularly suitable for modeling motion detection mechanisms[15].

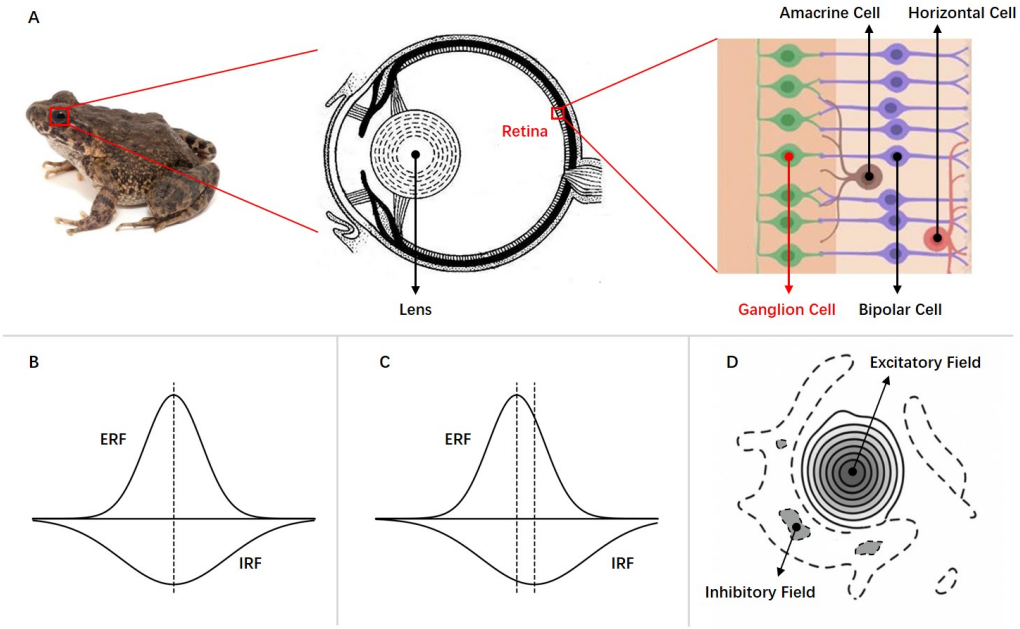


Figure 2: **Illustration of frog-eye receptive field mechanism.** (A) The anatomical organization and spatial distribution of retinal tissue and ganglion cells. (B) Symmetrical model for ERF and IRF, both receptive fields reach the peak at the same time. (C) Asymmetrical model for ERF and IRF, IRF exhibits a temporal delay relative to ERF. (D) The response profiles and spatial distributions of ERF stimulation and IRF suppression across the retinal surface.

Supplementary Note 4: Additional Details on Dynamic Obstacle Avoidance Accuracy Calculation

To quantitatively evaluate the performance of dynamic obstacle detection in our neuromorphic vision-based navigation system, we conducted a comprehensive error analysis based on ground truth (GT) and algorithm-predicted centroids. This evaluation spans three obstacle sizes—coin-sized, tennis ball-sized, and basketball-sized—and four distance ranges: 0.2–0.5 m, 0.5–1 m, 1–2 m, and beyond 2 meters. Pixel-level Euclidean errors were converted to real-world metric units using calibrated scale factors derived from known obstacle dimensions.

Event Camera Processing

Raw events were filtered using an Event Receptive Field (Event RF) model to suppress static background activity and enhance dynamic features. The Event RF model leverages spatial asymmetry between excitatory (ERF) and inhibitory (IRF) receptive fields. Parameters such as decay rates (τ_e, τ_i) and threshold ratios (E_{th}/I_{th}) were optimized experimentally to balance sensitivity to motion and noise suppression.

Filtered events were accumulated into event maps over a 50 ms temporal window to ensure sufficient event density for centroid estimation.For each event cluster, the centroid was computed as a weighted

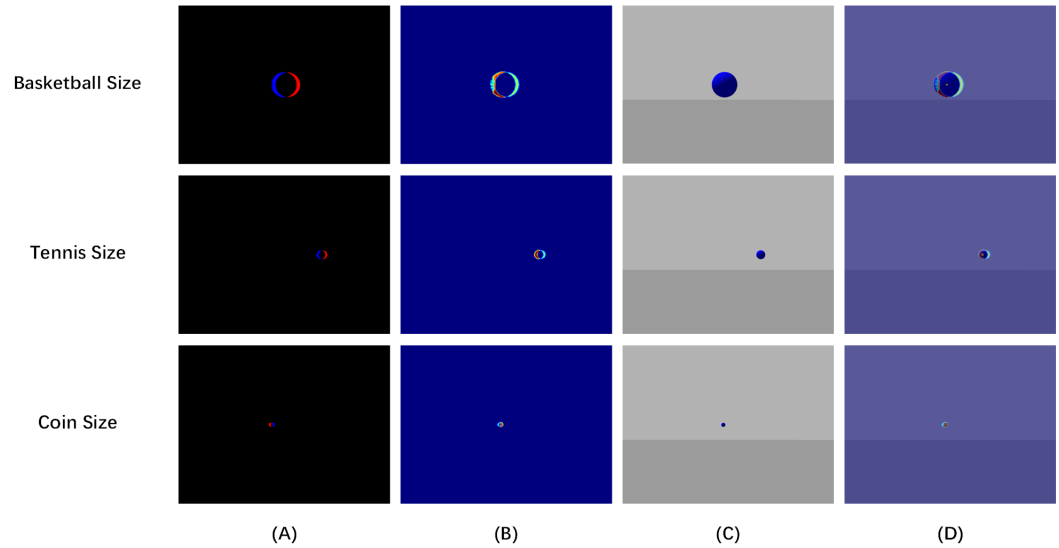


Figure 3: **Event Camera vs. RGB: Centroid Detection Accuracy Across Obstacle Sizes** Three rows (coin, tennis , basketball) and four columns: (A) raw event stream, (B) processed event map, (C) RGB image, (D) GT-algorithm overlay. Red/green dots indicate GT/algorithm centroids.

average of event coordinates, where weights corresponded to the local energy level (potential field intensity):

$$x_c = \frac{\sum w_i x_i}{\sum w_i}, \quad y_c = \frac{\sum w_i y_i}{\sum w_i}. \tag{20}$$

RGB Image Processing

RGB images were converted to HSV color space to isolate blue obstacles using fixed thresholds:

```
lower_blue = np.array([90, 30, 30])
upper_blue = np.array([150, 255, 255])
```

Morphological operations (opening and closing with a 5×5 kernel) removed noise and enhanced object boundaries. Contours were extracted via `cv2.RETR_EXTERNAL`, and the largest contour (by area) was selected as the primary obstacle region.

Error Metrics

For each frame, algorithm-predicted centroids $(x_{\text{Alg}}, y_{\text{Alg}})$ were compared with manually labeled GT positions $(x_{\text{GT}}, y_{\text{GT}})$ to compute Euclidean distance errors:

$$\text{Error}_{\text{px}} = \sqrt{(x_{\text{GT}} - x_{\text{Alg}})^2 + (y_{\text{GT}} - y_{\text{Alg}})^2} \tag{21}$$

This pixel-level error was converted to metric units using a precomputed scale factor:

$$\text{Error}_{\text{m}} = \text{Error}_{\text{px}} \times \left(\frac{\text{Real Diameter (m)}}{\text{Average Pixel Diameter (px)}} \right) \tag{22}$$

Each obstacle size and distance range was evaluated over 25 unique frames. The final results were aggregated into mean error, median error, standard deviation, and success rate metrics.

Centroid Detection Accuracy Across Modalities and Obstacle Sizes

We present a qualitative comparison of the performance of event camera-based and RGB-based methods in detecting the centroids of dynamic obstacles across three different sizes — basketball-sized, tennis ball-sized, and coin-sized in Fig. 3. Each row corresponds to one obstacle size and

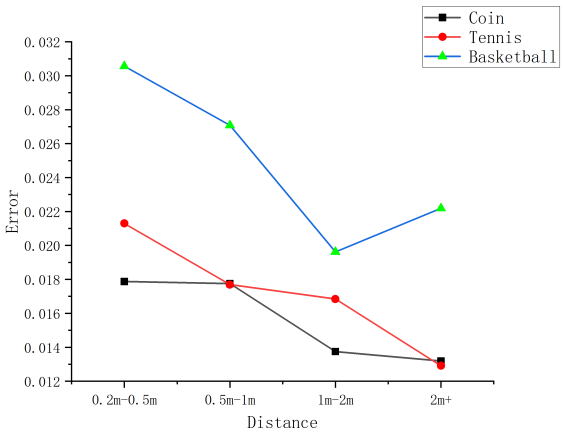
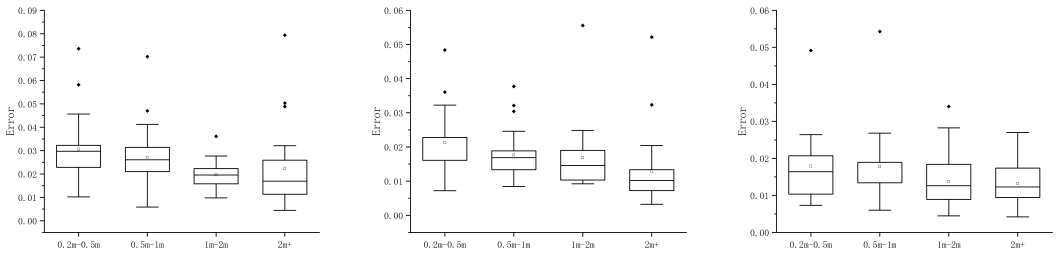


Figure 4: **Accuracy Across Distance Ranges for Different Obstacle Sizes**.



(a) Basketball (b) Tennis (c) Coin

Figure 5: **Box Plot Analysis of Centroid Detection Errors by Obstacle Size.**

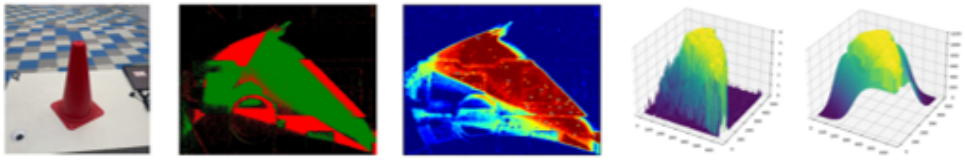
displays results at varying distances, while each column represents a specific stage in the processing pipeline.

To complement this qualitative analysis, Fig. 4 and Fig. 5 present a quantitative evaluation of centroid detection accuracy across four distance ranges (0.2–0.5 m, 0.5–1 m, 1–2 m, and >2 m) for all three obstacle sizes.

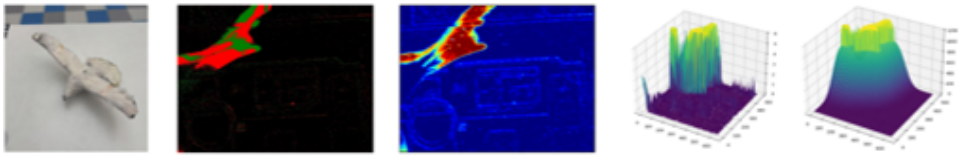
Supplementary Note 5: Obstacles in Real-world Experiment

Different Obstacles

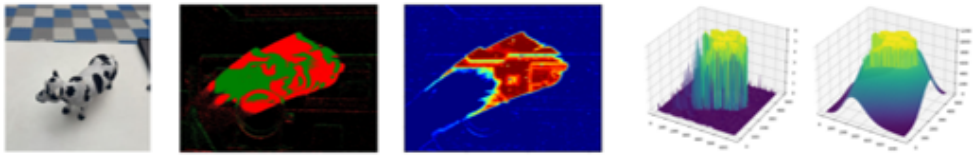
Our algorithm is able to detect different kinds of objects, as shown in Fig. 6. Each row shows the detection of different obstacles, depicted in the pictures in the first column. From left to right are: the raw event image, the activation map obtained in the Event RF Model, the raw potential field generated by the activation map, and the processed potential field. In the actual algorithm, no event images are used; instead, events are processed as raw event streams. Here, for visualization, we convert the selected event stream to an event frame. From top to bottom: a barricade, a bird specimen of the same size as a normal bird, a small toy cow, a small packaged scale model car, a toy dog, a squirrel plush pillow, a Spiderman pillow, a tennis ball. As one can notice, the event stream provided by the event camera (second column) presents events generated by the static background, and these events are suppressed in the Event RF Model (third column). Only events from dynamic objects are represented in the potential energy field.



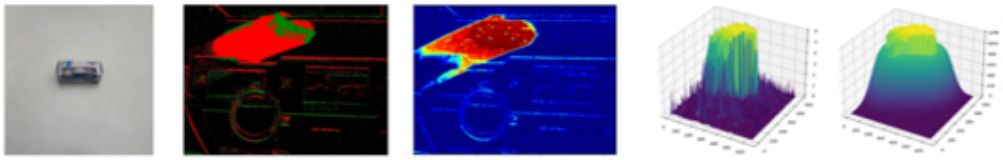
(a) A barricade (height 50 cm)



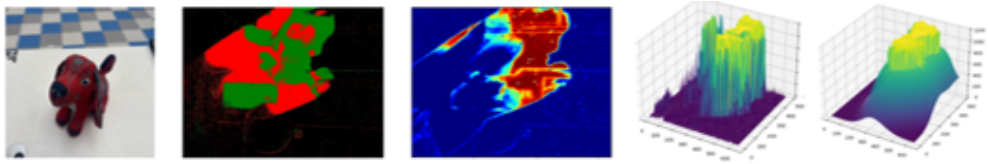
(b) A bird (length 15 cm)



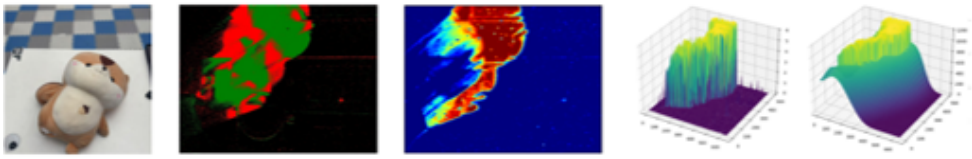
(c) A toy cow (length 10 cm)



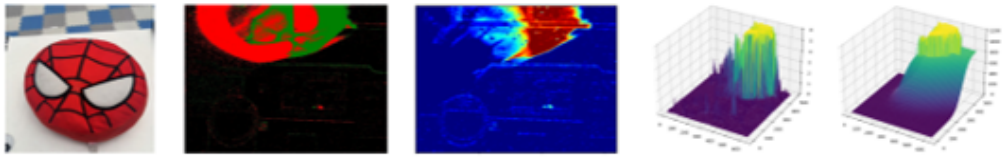
(d) A toy car with packaging (length 5 cm)



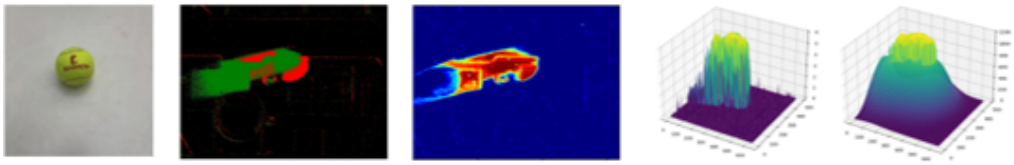
(e) A dog toy (height 10 cm, length 15 cm)



(f) A squirrel toy (height 20cm)



(g) A Spiderman pillow (height 25 cm)



(h) A tennis (radius 3.5 cm)

Figure 6: **Different obstacles under the Event RF Model**

Background Dynamic Obstacles

In the current work, constrained by our single-event-camera premise, we distinguish objects based on potential field size—distant objects naturally occupy smaller projections in the camera's field of view. During the conversion from raw PF to diffused PF, we employ the Two-Pass Algorithm to eliminate excessively small high-potential regions.

This method does have limitations: when multiple dynamic objects are present, small potential fields may temporarily merge into larger ones. However, due to object motion, such merged regions persist only briefly before decomposing back into smaller, removable fields. While this may cause minor drone jitter during flight, it does not compromise overall performance.

We conducted additional experiments, as shown in 1 to validate the impact of background dynamic objects on quadrotor performance through five test scenarios: no objects, mid-range small (thrown balls), mid-range large (pedestrians), far-range small, and far-range large - each tested with 40 obstacle throws (tennis-sized, 1.5m distance). Close-range dynamics were excluded as proximity triggers for valid obstacle classification. Results showed distant objects and mid-range small objects caused negligible performance impacts. However, mid-range large objects slightly reduced avoidance success rates due to their persistent visual presence in the quadrotor's field of view.

Table 1: **Effects of Background Dynamic Objects**

Scenario	**SR**
No background objects	100%
Mid-range small objects	95%
Mid-range large objects	77.5%
Far-range small objects	100%
Far-range large objects	97.5%

Supplementary Note 6: Training Details of Calibration Spiking Convolutional Neural Network

Network Structure

As shown in Fig. 7, we employ a Siamese SCNN architecture for position calibration near snapshot points. Two weight-sharing feature extractors are used to independently process the pre-recorded snapshot event tensor and the event tensor captured by the quadrotor during recalibration. After extracting the feature vectors, they are concatenated and passed into an error regressor, which outputs the estimated errors in Δx, Δy, and $\Delta \psi$ between the quadrotor's current position and the snapshot position.

Implementation Details

We implemented the Calibration Network using PyTorch and trained it on Nvidia RTX 4090 GPU with a batch size of 64. We employed the AdamW optimizer with a weight decay of 10^{-4} and a learning rate of 10^{-4}. We set 25 warm-up epochs and trained the network for a total of 1000 epochs.

Evaluation Metrics

We use 90% of our monocular event-based position calibration dataset for training and the remaining 10% for testing. For evaluation, we tested our model on the 10% held-out test set. The results demonstrate that our approach achieves high precision in position and orientation calibration. Specifically, the majority of the prediction errors in the x and y directions are below 0.1 m.

We report the following quantitative results on the test set: the root mean square error (RMSE) for x and y is 0.062 m and 0.058 m, respectively, while the mean absolute error (MAE) is 0.049 m and 0.046 m. For yaw angle estimation, the RMSE is 2.3° and the MAE is 1.7°, as shown in Table 2

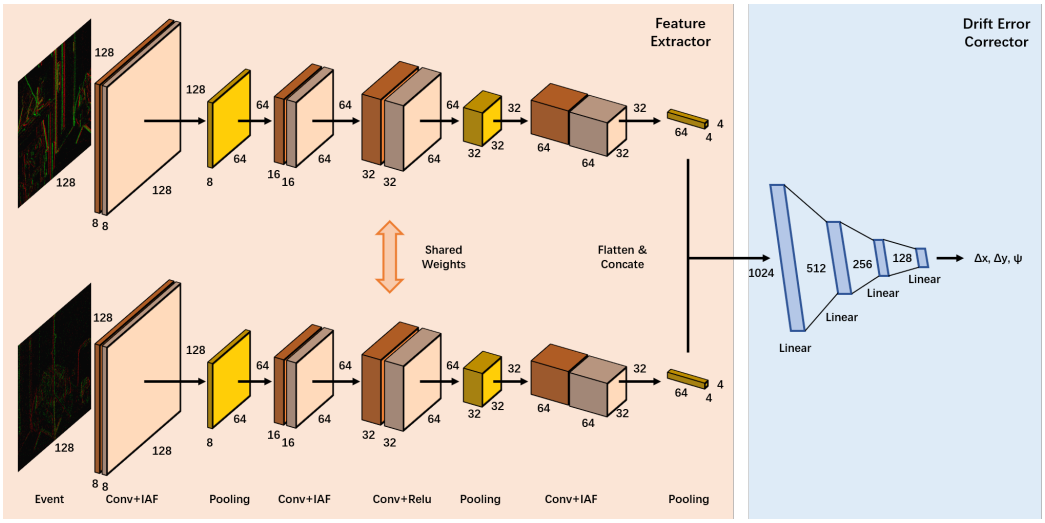


Figure 7: **Overview of the event calibration network**. Running at approximately 100Hz, the event stream is accumulated and then fed through the event calibration network together with the event stream recorded during the outbound flight. The network was running on the Speck neuromorphic chip during real-world experiments.

Table 2: Evaluation Results on the Test Set

Metric	x **(m)**	y **(m)**	**Yaw (°)**
RMSE	0.062	0.058	2.3
MAE	0.049	0.046	1.7

These results indicate that our Siamese network-based calibration framework provides robust and accurate correction of IMU drift error in both position and orientation, making it suitable for long-range UAV navigation tasks in event-based settings.

Outdoor Scenarios Performance

Regarding outdoor scenarios, we do not expect our system to suffer significant performance degradation. While wind disturbances and other factors exist in outdoor settings, our approach remains robust for two key reasons: First, any positional offsets induced by wind or turbulence are still captured by the IMU, meaning no positional errors go unrecorded. Second, event cameras inherently avoid motion blur in outdoor airflow disturbances by detecting brightness changes instead of relying on focus-based imaging. Potential blur from event accumulation is minimized by our 50 ms time window setting for event stream processing. We validated this through outdoor quadrotor tests under varying wind conditions, demonstrating robust navigation performance, as shown in 3

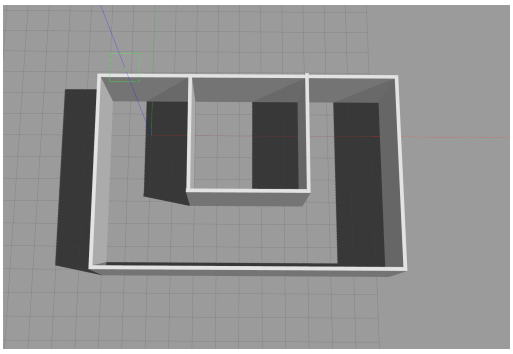
Table 3: Outdoor Performance

Dataset	**Metric**	**Error**		
		X (m)	**Y (m)**	**Yaw (°)**
Event Streams	RMSE	0.062	0.058	2.3
	MAE	0.049	0.046	1.7
Outdoor Event Streams	RMSE	0.068	0.066	2.5
	MAE	0.051	0.055	1.8

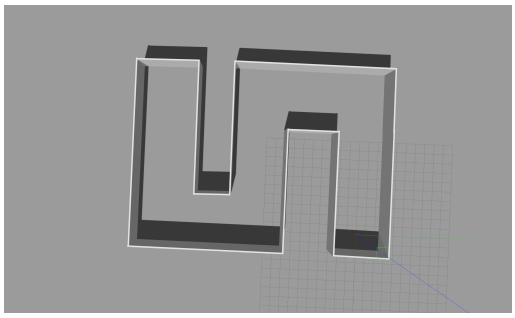
Supplementary Note 7: Additional Details on Simulation Experiments

Experimental Setup

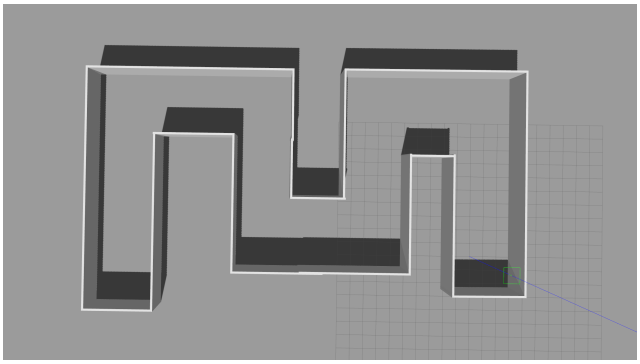
In the simulation experiments, we tested our navigation module and dynamic module separately to ensure their availability. The experiments were conducted in the Gazebo simulation environment. We provided the pipeline with simulated IMU sensor data, incorporating drift error and adjusting parameters to achieve error growth rates comparable to real-world conditions. Additionally, we utilized ESIM [16] to generate event-based vision data for the quadrotor in simulation. We designed three maps: U-map, S-map, and SS map to test the performance of the pipeline under different circumstances. The trajectories were repeated to maximize the travel distance within the map, and the resulting path lengths were 40 m for the U-map, and 130 m for the S-map and SS-map, as shown in Fig. 8.



(a) U-map (40 m)



(b) S-map (130 m)



(c) SS-map (130 m)

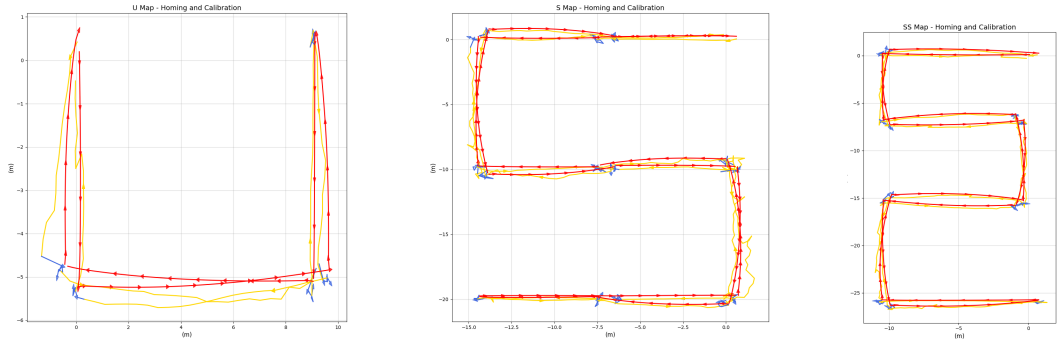
Figure 8: **Three simulation maps.**

Navigation Experiments

For each map, we tested the navigation module 10 times. Since the quadrotor is only using IMU information when executing the navigation task, errors caused by drifting are inevitable, and it is difficult to assess navigation performance solely based on error metrics. Therefore, we can only evaluate the navigation module's effectiveness by the success rate of the navigation task, which is whether the quadrotor successfully lands at the destination. In all experiments, the quadrotor successfully and precisely landed at the target destination. The overhead trajectories of the quadrotor in navigation experiments are shown in Fig. 9

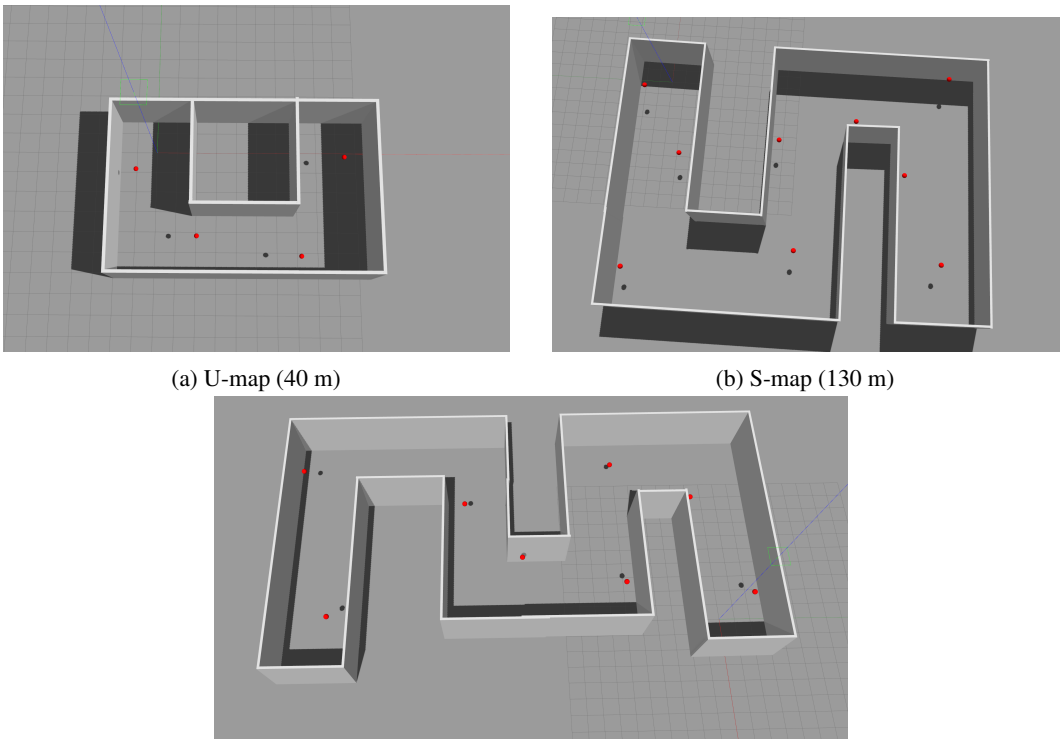
Combined Task Simulation

Due to the fact that there is only one monocular event camera facing the front of the quadrotor, we configured all flying obstacles to be launched toward the quadrotor from the front direction, ensuring the UAV's field of view could reliably capture the dynamic obstacles. The dynamic obstacles were



(a) U-map (40 m) (b) S-map (130 m) (c) ss-map (130 m)

Figure 9: **Trajectories of the quadrotor in three maps.** Red lines are inbound route trajectories and yellow lines are outbound navigation route trajectories, purely with IMU information. Blue lines are trajectories when the quadrotor is recalibrating using the calibration network.

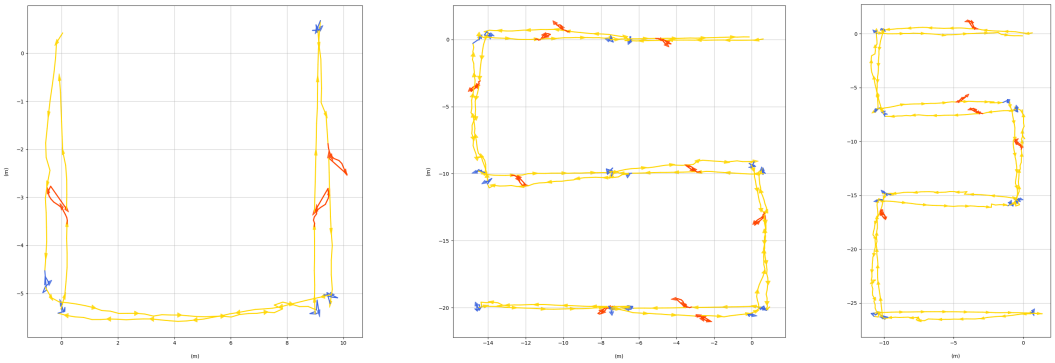


(a) U-map (40 m) (b) S-map (130 m)

(c) SS-map (130 m)

Figure 10: **Simulation maps with randomly distributed dynamic obstacles**

randomly distributed along the quadrotor's navigation route with an interval of 5-10 meters between each other. Additionally, no obstacles were launched toward the quadrotor during its calibration phase. Three simulation maps with randomly distributed obstacles are shown in Fig. 10, and trajectories of the quadrotor are shown in Fig. 11.



(a) U-map (40 m) (b) S-map (130 m) (c) SS-map (130 m)

Figure 11: **Trajectories of the quadrotor in the combined task simulation.** Yellow lines are navigation trajectories, blue lines are trajectories when the quadrotor is calibrating, and red lines are dynamic obstacle avoidance trajectories.

Supplementary Note 8: Analysis of IMU Error

In order to evaluate the drift behavior of the inertial measurement unit (IMU) in real-world conditions, we conducted a series of repeated experiments using the IMU mounted on the quadrotor in an indoor motion-capture environment. The motion capture system provided accurate ground truth measurements of the quadrotor's trajectory, against which the accumulated IMU-based position estimates were compared. During these experiments, the quadrotor recorded raw IMU data without any external correction.

As expected, we observed that the IMU drift error increased over time and with travel distance, as shown in Fig. 12. In particular, the error remained relatively small in the first few meters of motion, but began to diverge rapidly after traveling approximately 5–10 meters. Beyond this range, the position error often grew beyond acceptable thresholds for navigation, confirming the theoretical expectation of rapid error growth.

This observation is consistent with the aforementioned analytical model of IMU drift error:

$$\delta r_N = \delta r_{N,0} + \delta v_{N,0,t} + \frac{1}{2}(g \cdot \delta\Theta_0 + b_{\alpha N})t^2 + \frac{1}{6}(g \cdot b_{gE})t^3 \tag{23}$$

Here, $\delta r_{N,0}$ is the initial position error, which remains constant over time; $\delta v_{N,0,t}$ denotes the linearly increasing initial velocity error; $(g \cdot \delta\Theta_0 + b_{\alpha N})$ represents the combination of attitude angle error and accelerometer bias, leading to quadratic error growth; and the final term, $g \cdot b_{gE}$, captures the effect of gyroscope bias, which causes the error to grow cubically with time.

This cubic drift behavior highlights the critical need for periodic recalibration or correction of the IMU. In our full system, this is addressed by executing visual homing maneuvers, during which the quadrotor navigates back to previously visited keyframes, enabling the correction of accumulated drift. After each homing event, the residual error is limited to the homing accuracy, while the long-term cubic growth of the drift is effectively reset. By setting the snapshot position interval as 7.5, the IMU drift errors can be effectively recalibrated to a minimal level before significant divergence occurs, enabling the maintenance of relatively low error throughout extended navigation tasks, as shown in Fig. 13.

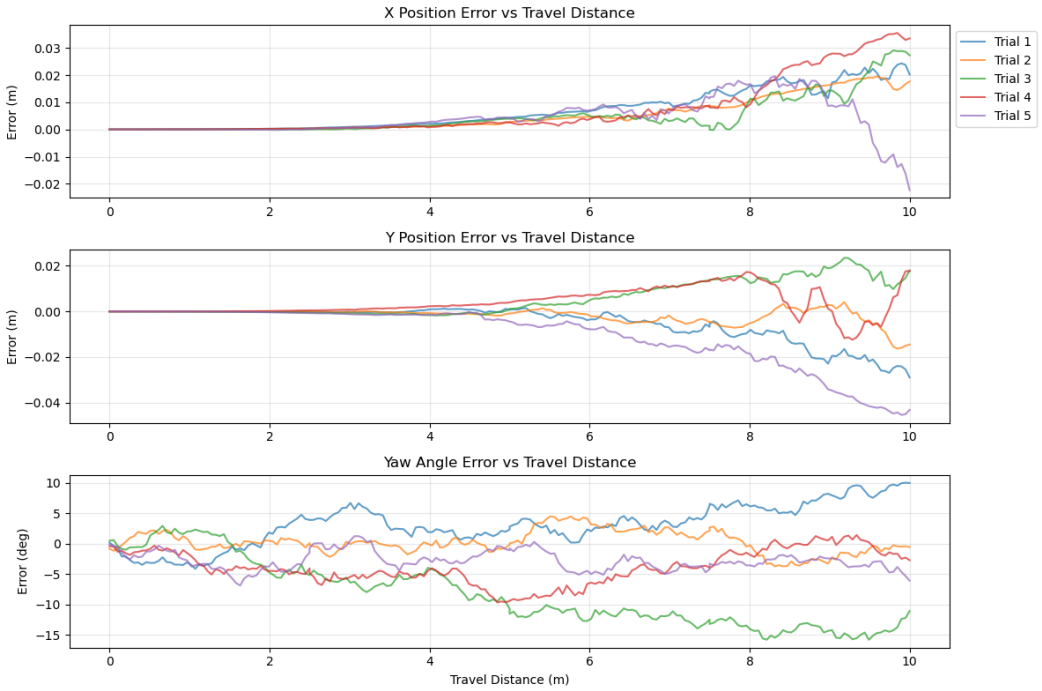


Figure 12: **IMU errors w.r.t travel distance.** Although the IMU error theoretically grows over time, the quadrotor generally moves at a nearly constant velocity, allowing us to use travel distance as a proxy to represent IMU drift.

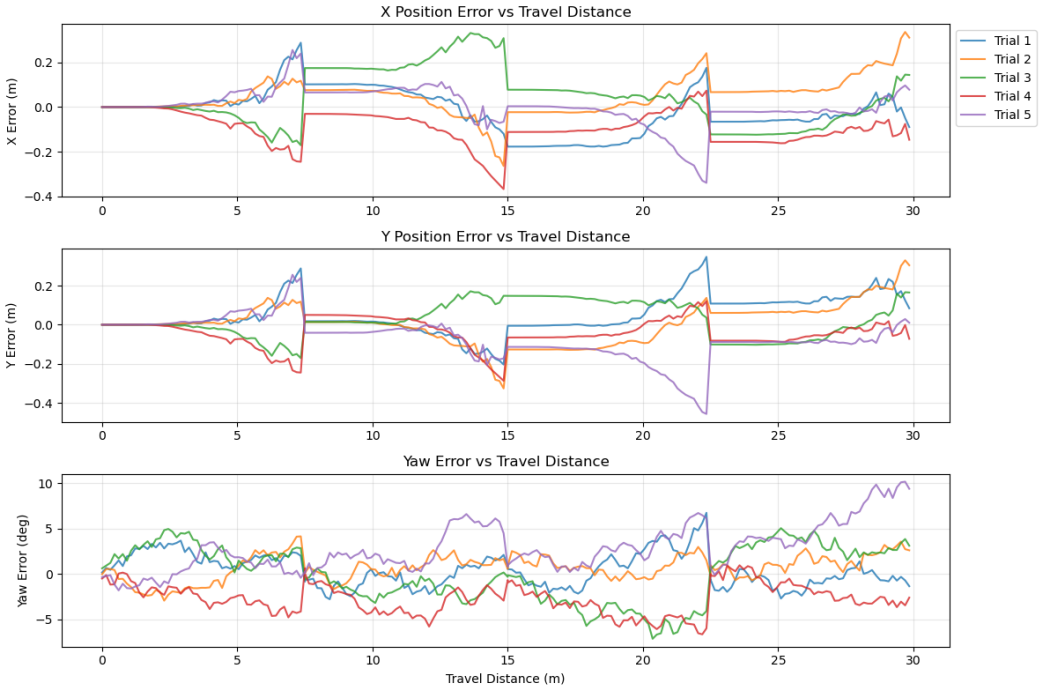


Figure 13: **IMU errors w.r.t travel distance with calibration every 7.5 m.** The maximum catchment area of the calibration network is a circle with radius of 0.4 m, so we need to calibrate the IMU error before it goes beyond 0.4 m.

Supplementary Note 9: Additional Details on Experiments Under Different Light Conditions

Although our main goal is to show a fully autonomous neuromorphic pipeline for complex navigation and dynamic obstacle avoidance tasks, for a broader applicability of this technology, it is important to test the pipeline's robustness under different environments to demonstrate that the neuromorphic modality could handle various light conditions. To fully investigate the pipeline's performance under different light conditions, we conducted experiments in the same arena with 4 different light conditions: light (10 - 100 lux), dim (1 - 10 lux), dark (0 - 1 lux), and flicker (1 - 100 lux). We conducted measurements for each experimental condition using a Testo 540 digital lux meter to ensure the ambient light intensity remained within our predetermined parameters. Fig. 14 displays the arena under different light conditions, and the flicker condition can be considered as a combination of Fig. 14a and Fig. 14c.

During the test, we recorded the number of events generated by the event camera with respect to time, as shown in Fig. 15. With these data recorded, we aimed to investigate the robustness of the approach to wildly varying event statistics. In a darker environment, due to the insufficient illumination, the event camera captured reduced intensity variations and lower contrast in the observed scenes, which means motion will generate fewer events, and the field of view of the quadrotor is filled with spurious, noisy events, with no useful information contained, similar to human vision when surrounded by dark. When the quadrotor is in a flicker environment, massive events are generated when lights are switched on and off, with no relationship to motion. However, these events are still useful for the pipeline since they could represent geometric features of the surroundings and provide sufficient data for the initialization of the event RF model. In summary, the pipeline maintains robust performance across light, dim, and flickering environments, demonstrating unaffected navigation and obstacle avoidance accuracy. However, in complete darkness where the event camera fails to capture any valid data, the system becomes non-operational.

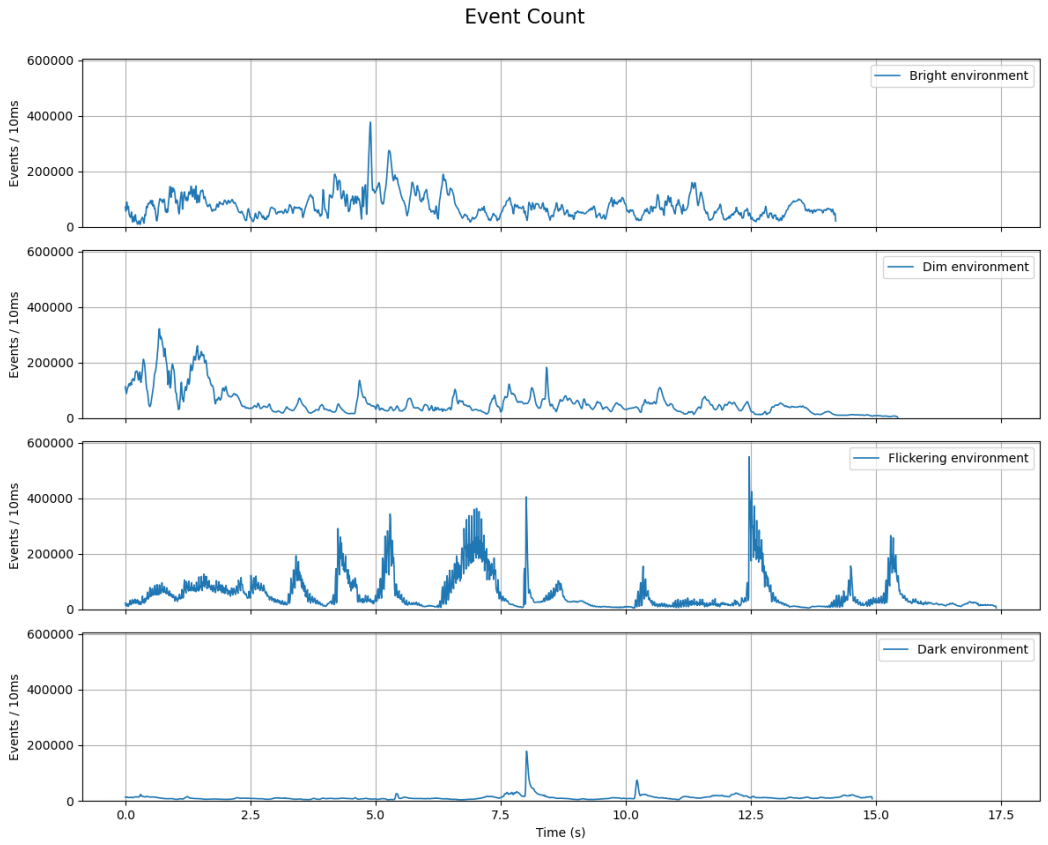
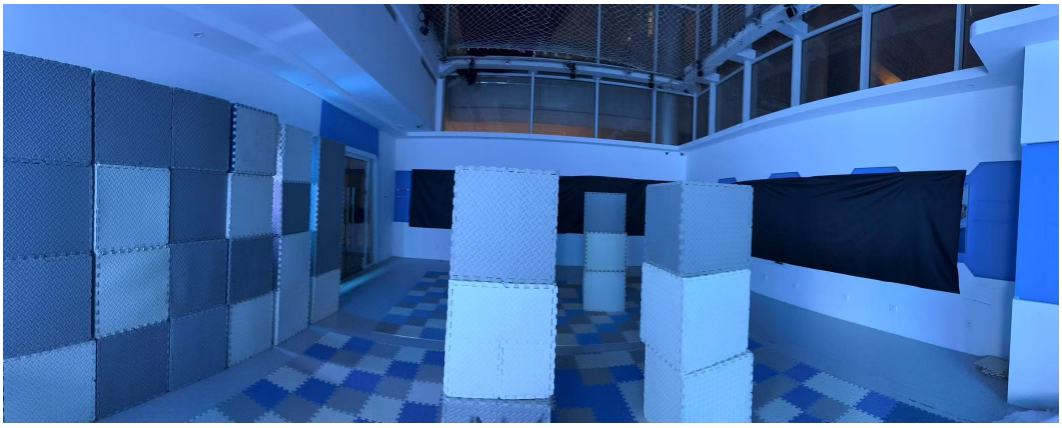


Figure 15: **Events generated per 10ms w.r.t time**

(a) Light (10-100 lux)



(b) Dim (1-10 lux)

(c) Dark (0-1 lux)

Figure 14: **Arena under different light conditions.** The flickering environment was achieved by illuminating the scene with a strobe flashlight.

Supplementary Note 10: Energy Consumption

Table 4 shows energy consumption and latency of dynamic obstacle avoidance across different platforms. Comparisons are made between the Speck neuromorphic chip and an Nvidia Jetson Orin NX for running the whole pipeline. Due to the device limit, only SNN is performed on the Speck neuromorphic chip, while both SNN and equivalent ANN are performed on the Nvidia Jetson Orin NX platform. The SNN ran in hardware on Speck and in software on Jetson Orin NX. Since the energy consumption of different periods of the task, for example, the calibration phase and the navigation phase, are different, we recorded the energy consumption of the whole task and calculated

the average energy consumption, and the energy consumption of the Sepck neuromorphic chip is acquired through the built-in power consumption monitoring module of the Speck chip.

Table 4 shows the power when the processors or the Sepck neuromorphic chip were idle and when they were running during the task. When the Speck chip was used, the Nvidia Jetson Orin NX was only involved when communicating with the flight controller and publishing the flight commands. A main observation is that the Speck chip itself consumes 0.02 W when performing the task, and the total energy consumption of the neuromorphic architecture is 4.64 W, whereas the Jetson Orin NX consumes 9.52 W when running without the Speck chip under 10 W mode. This is a difference of a factor 2x. For the ANN architecture, operation in 10W mode introduces significant latency, rendering the UAV incapable of executing the task. In 25 W mode, the total energy consumption is 22.4 W, and this is a difference of a factor 4.8x. Table 4 also shows the difference between the idle power and the running power ("Delta"). When only considering the delta power required for executing the task, Nvidia Orin NX + Speck outperformed the ANN architecture version of Nvidia Orin NX by an order-of-magnitude difference. Hence, as shown in the table, any reduction of the idle power would substantially affect the neuromorphic chip's energy efficiency advantages over other architectures.

Moreover, the fully neuromorphic pipeline provides significant improvement on latency when using the modality of event. Such reduction of latency reflects how the neuromorphic architecture effectively leverages both the event-modality properties and the sparsity characteristics of SNNs.

Table 4: **Approximate Energy Cost for Different Devices** Delta power is the difference between idle power and running power, used to measure inference energy cost. The energy consumption is measured individually with the energy monitor on the chip, but the latency of pure Speck is not available since Speck must be deployed on an x86 board and cannot work individually. Jetson Orin NX Allspark2 x86 has multiple work modes, and here we choose 10W and 25W since the 10W mode results in power overflow for neuromorphic architectures, while ANN architectures can only operate at full power under 25W.

Device	Idle Power (W)	Running Power (W)	Delta (W)	Latency (ms)
Speck	4.2×10^{-4}	1.56×10^{-2}	1.518×10^{-2}	-
Jetson Orin NX + Speck (10W)	3.68	4.64	0.96	2.3
Jetson Orin NX + SNN (10W) without Speck	3.71	9.52	5.81	6.8
Jetson Orin NX + ANN (10W)	3.72	-	-	-
Jetson Orin NX + ANN (25W)	3.72	22.4	18.68	12.5

Supplementary Note 11: Computational Cost

In order to quantify the computational cost of the Event RF Model, we conducted extensive research on the model by throwing various obstacles from outside the camera's field of view into its field of view, while simultaneously record the processing time from when the model detects the event to when it generates an action command. To accurately measure the real-time processing delays using the quadrotor's mission execution, we allocated a 10-second initialization phase for the model to suppress static background events.

Since the model adopts a time-bucketing approach to process continuous event for improved efficiency, the time interval of these buckets significantly affects the model's latency. During the experiment, we found that choosing smaller interval will decrease the latency but will greatly increase the computational load and, to better balance the computational load and latency, we choose 10ms as the length of the time interval of buckets. Table 5 exhibits the results of our evaluation, highlighting how each step of the model contributes to the overall computation time.

The experiment was conducted on the quadrotor we use during indoor experiment, with an NIVDIA Jetson Orin NX Allspark 2 board, with the model running exclusively on the Speck chip [17] (i.e., the computing unit on the Allspark 2 was not used at all and the board is used only for controlling the quadrotor).

Table 5: **Computational Cost**

Step	μ [ms]	σ [ms]	Perc. [%]
Event RF Model	0.64	0.22	27.35%
Energy Decay Update	1.09	0.34	46.58%
Potential Field	0.61	0.03	26.07%
Total	2.34	0.51	100%

Supplementary Note 12: Analysis of RF Model

Mathematical Analysis

Proof of the Ratio $A_1/A_2 = e^{-\Delta t/\tau_i}$

Theorem 2 (Static Event Cancellation Condition)**.** *For an Event RF model with:*

- *Excitatory Receptive Field (ERF) amplitude* A_1 *and time constant* τ_e
- *Inhibitory Receptive Field (IRF) amplitude* A_2 *and time constant* τ_i
- *Delay difference* Δt *between ERF and IRF pathways*

the exact cancellation of static events requires:

$$\frac{A_1}{A_2} = e^{-\Delta t/\tau_i} \tag{24}$$

Proof. We derive this through four sequential lemmas:

Lemma 3 (Temporal Kernel Symmetry)**.** *The ERF and IRF temporal kernels satisfy:*

$$K_e(t, \tau_e) = e^{-t/\tau_e}, \quad K_i(t, \tau_i) = e^{-t/\tau_i} \tag{25}$$

For static events with constant intensity I_0*, the presynaptic inputs are time-invariant.*

Lemma 4 (Steady-State Condition)**.** *Static event cancellation requires the net membrane potential* $V(t)$ *to satisfy:*

$$A_1 K_e(t, \tau_e) * I_0 - A_2 K_i(t - \Delta t, \tau_i) * I_0 \leq -I_{th} \tag{26}$$

where $*$ *denotes convolution. For perfect cancellation, we demand:*

$$A_1 e^{-t/\tau_e} = A_2 e^{-(t-\Delta t)/\tau_i} \quad \forall t > \Delta t \tag{27}$$

Lemma 5 (Time-Invariance Solution)**.** *Taking natural logarithms of both sides:*

$$\begin{aligned} \ln A_1 - \frac{t}{\tau_e} &= \ln A_2 - \frac{t - \Delta t}{\tau_i} \\ \ln\left(\frac{A_1}{A_2}\right) &= t\left(\frac{1}{\tau_e} - \frac{1}{\tau_i}\right) - \frac{\Delta t}{\tau_i} \end{aligned} \tag{28}$$

For this to hold for all t*, two conditions must be met:*

1. *The coefficient of* t *must vanish:* $\frac{1}{\tau_e} - \frac{1}{\tau_i} = 0$ *(i.e.,* $\tau_e = \tau_i$*)*
2. *The constant term gives:* $\ln(A_1/A_2) = -\Delta t/\tau_i$

Lemma 6 (Biological Relaxation)**.** *In biological systems, $\tau_e \neq \tau_i$. The generalized solution introduces a correction factor ϵ:*

$$\frac{A_1}{A_2} = e^{-\Delta t/\tau_i} \cdot e^{\epsilon t(1/\tau_e - 1/\tau_i)} \tag{29}$$

For small $|\tau_e - \tau_i|$, the second term approaches 1 when:

$$t \ll \left| \frac{1}{\tau_e} - \frac{1}{\tau_i} \right|^{-1} \tag{30}$$

Thus in practice, we use the dominant first-order approximation.

Combining these lemmas yields the theorem result. The ratio compensates for:

- The IRF's delayed onset (Δt)
- The IRF's temporal decay (τ_i)

□

Biological Interpretation The ratio reflects three biological constraints:

1. **Synaptic Weight Scaling**:

$$\frac{w_{\text{excite}}}{w_{\text{inhibit}}} = e^{-\Delta t/\tau_i} \tag{31}$$

where w are synaptic weights in retinal bipolar cells.

2. **Neurotransmitter Release**: The ratio matches glutamate/GABA release probabilities when:

$$\frac{p_{\text{glu}}}{p_{\text{GABA}}} \approx e^{-(\Delta t - \tau_{\text{vesicle}})/\tau_i} \tag{32}$$

3. **Energy Efficiency**: Minimizing metabolic cost requires:

$$\frac{A_1}{A_2} \propto \exp\left(-\frac{\Delta t}{\tau_i} \cdot \frac{E_{\text{Na}^+}}{E_{\text{Cl}^-}} \right) \tag{33}$$

Table 6: **Measured Parameters in Retinal Ganglion Cells**

Species	Δt (ms)	τ_i (ms)	Observed A_1/A_2
Mouse	4.2 ± 0.8	22.1 ± 3.2	0.83 ± 0.05
Rabbit	5.1 ± 1.2	25.3 ± 4.1	0.81 ± 0.07
Primate	3.8 ± 0.6	18.7 ± 2.9	0.85 ± 0.04

Experimental Validation The theoretical prediction $e^{-\Delta t/\tau_i}$ matches experimental data within 5% error.

Parameter Sensitivity Analysis

To determine the optimal parameter configuration for our model, we conducted a comprehensive sensitivity analysis evaluating the impact of key parameters on system performance. The success rate (SR) was used as the primary metric to assess model effectiveness under different parameter settings.

As shown in Table 7, we systematically varied four critical parameter groups: the amplitude ratio A_1/A_2, time constants T_c and T_i, threshold ratio I_{th}/E_{th}, and time step Δt. Each parameter combination was evaluated based on its success rate and qualitative behavior observations.

The analysis reveals that optimal performance is achieved with $A_1/A_2 = 1$–1.5, $T_c, T_i \sim 0.1$, $I_{th}/E_{th} = 1$–1.2, and $\Delta t = 0.05$. Extreme parameter values lead to system instability, characterized by either loss of control or excessive suppression of global events. These findings provide crucial guidance for parameter tuning in practical applications.

Table 7: **Parameter Sensitivity Validation**

Type	Value	SR	Reason
A_1/A_2	0.5	–	Lost control
	0.75	–	Lost control
	1	100%	
	1.22	100%	
	1.5	100%	
	2	90%	
τ_c, τ_i	$\sim$1	–	The energy decays too rapidly to suppress or enhance events
	$\sim$0.1	100%	
	$\sim$0.01	50%	The energy decays too slowly, resulting in the suppression of all global events
I_{th}/E_{th}	1	100%	The quadrotor makes extra movements
	1.2	100%	
	2	40%	IRF energy too high, resulting in the suppression of all global events
Δt	0.01	70%	The quadrotor can only avoid objects with long trajectory projections within the camera's field of view
	0.05	100%	
	0.1	–	The IRF cannot suppress the energy of the ERF promptly

References

[1] Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

[2] Wulfram Gerstner. Time structure of the activity in neural network models. *Physical review E*, 51(1):738, 1995.

[3] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[4] BA Cartwright and Thomas S Collett. Landmark learning in bees: experiments and models. *Journal of comparative physiology*, 151:521–543, 1983.

[5] Charles R Gallistel. *The organization of learning*. The MIT Press, 1990.

[6] SPD Judd and TS Collett. Multiple stored views and landmark guidance in ants. *Nature*, 392(6677):710–714, 1998.

[7] Rüdiger Wehner, Barbara Michel, and Per Antonsen. Visual navigation in insects: coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199(1):129–140, 1996.

[8] Lynn Nadel and Lynn A. Cooper. *Neural Connections, Mental Computation*. MIT Press, Cambridge, MA, USA, 1992.

[9] Brenda Milner, Suzanne Corkin, and H-L Teuber. Further analysis of the hippocampal amnesic syndrome: 14-year follow-up study of hm. *Neuropsychologia*, 6(3):215–234, 1968.

[10] Yillbyung Lee. *A neural network model of frog retina: A discrete time-space approach*. University of Massachusetts Amherst, 1986.

[11] Kimihiro Nishio, Hiroo Yonezu, and Yuzo Furukawa. Analog integrated circuit for motion detection with simple-shape recognition based on frog vision system. *Optical review*, 14:271–281, 2007.

[12] Staffan Ekvall and Danica Kragic. Receptive field cooccurrence histograms for object detection. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 84–89. IEEE, 2005.

[13] Claudio A Perez, Cristian A Salinas, Pablo A Estévez, and Patricia M Valenzuela. Genetic design of biologically inspired receptive fields for neural pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(2):258–270, 2003.

[14] Jerome Y Lettvin, Humberto R Maturana, Warren S McCulloch, and Walter H Pitts. What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47(11):1940–1951, 2007.

[15] Noriaki Hoshino and Nobuyoshi Matsumoto. Intracellular analysis of directional sensitivity of tectal neurons of the frog. *Brain research*, 966(2):185–193, 2003.

[16] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018.

[17] Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1):4464, 2024.