A Hybrid Compact Neural Architecture for Visual Place Recognition

Marvin Chancán[1,3], Luis Hernandez-Nunez[2,3], Ajay Narendra[4], Andrew B. Barron[4], and Michael Milford[1]

Abstract—State-of-the-art algorithms for visual place recognition, and related visual navigation systems, can be broadly split into two categories: computer-science-oriented models including deep learning or image retrieval-based techniques with minimal biological plausibility, and neuroscience-oriented dynamical networks that model temporal properties underlying spatial navigation in the brain. In this letter, we propose a new compact and high-performing place recognition model that bridges this divide for the first time. Our approach comprises two key neural models of these categories: (1) *FlyNet*, a compact, sparse two-layer neural network inspired by brain architectures of fruit flies, *Drosophila melanogaster*, and (2) a one-dimensional continuous attractor neural network (CANN). The resulting *FlyNet+CANN* network incorporates the compact pattern recognition capabilities of our FlyNet model with the powerful temporal filtering capabilities of an equally compact CANN, replicating entirely in a hybrid neural implementation the functionality that yields high performance in algorithmic localization approaches like SeqSLAM. We evaluate our model, and compare it to three state-of-the-art methods, on two benchmark real-world datasets with small viewpoint variations and extreme environmental changes – achieving 87% AUC results under day to night transitions compared to 60% for Multi-Process Fusion, 46% for LoST-X and 1% for SeqSLAM, while being 6.5, 310, and 1.5 times faster, respectively.

Index Terms—Biomimetics, Localization, Visual-Based Navigation

I. Introduction

PERFORMING visual place recognition (VPR) reliably is a challenge for any robotic system or autonomous vehicle operating over long periods of time in real-world environments. This is mainly due to a range of visual appearance changes over time (e.g. day/night or weather/seasonal cycles), viewpoint variations or even perceptual aliasing (e.g. multiple places may look similar) [1]. Convolutional neural networks (CNN), heavily used in a range of computer vision tasks [2], have also been applied to the field of VPR with great success over the past five years [3], [4]; typically only

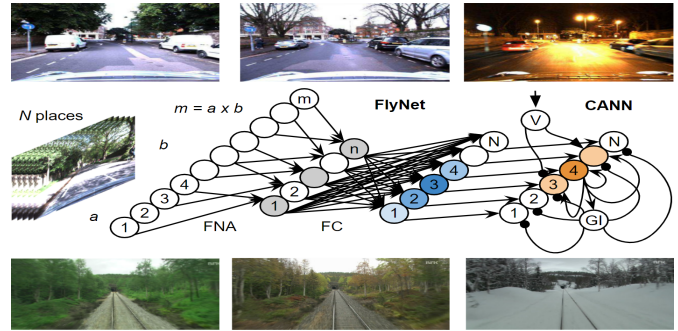Fig. 1. **FlyNet+CANN hybrid neural architecture**. Our FlyNet model comprises a hidden layer inspired by the *Drosophila* olfactory neural circuit, FlyNet algorithm (FNA), and a fully connected (FC) output layer. We integrate FlyNet with a continuous attractor neural network (CANN) to perform appearance-invariant visual place recognition. Experiments on two real-world datasets, Oxford RobotCar (top) and Nordland (bottom), show that our hybrid model achieves competitive results compared to conventional approaches, but with a fraction of computational footprint (see Fig. 2).
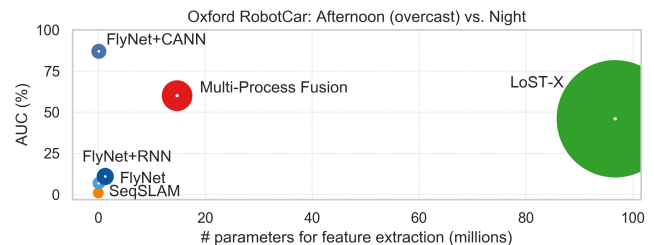

Fig. 2. **Oxford RobotCar AUC performance vs. Network Size**. Footprint comparison for the most challenging appearance change (day to night).

used in real-time with dedicated hardware (GPU) though [5]–[7]. However, as vanilla CNN models, trained on benchmark datasets such as ImageNet [8] or Places365 [9], generally neglect any temporal information between consecutive images. Conversely, sequence-based algorithms such as SeqSLAM [10] are often applied on top of these models to achieve state-of-the-art results on VPR tasks by matching two or more sequences of images.

Related research in visual navigation has recently used computer-science-oriented recurrent neural networks (RNN) [11] in an attempt to model the multi-scale spatial representation and network dynamics found in the entorhinal cortex of mammalian brains [12], [13]. While the results are promising, these systems are tested only in small synthetic environments, and the integration of neuroscience-oriented recurrent models such as continuous attractor neural networks (CANN) [14], [15] is not well explored. Only recently, analytic theories to unify both types of recurrent networks, trained on navigation tasks, have been proposed [16].

In this work, we propose a hybrid neural network that incorporates both computer-science- and neuroscience-oriented models, as in recent work [17], [18], but for VPR tasks for the first time[1]. Our approach comprises two key components (see Fig. 1): FlyNet, a compact neural network inspired by the *Drosophila* olfactory neural circuit, and a 1-*d* CANN as our temporal model that encodes sequences of images to perform appearance-invariant VPR using real data. The resulting FlyNet+CANN model achieves competitive AUC results on two benchmark datasets, but with far less parameters, minimal training time and smaller computational footprint than conventional deep learning and algorithmic-based approaches. In Fig. 2, for instance, the area of the circle is proportional to the number of layers per model, being 213 for the ResNet-based LoST-X pipeline [19], 13 for Multi-Process Fusion [20], and 3 for our proposed FlyNet+CANN hybrid model.

The rest of the paper is structured as follows. Section II provides a brief overview of VPR research and the biological inspiration for our hybrid neural architecture; Section III describes the FlyNet model in detail; Sections IV and V present the experiments and results, respectively, where we compare our approach to three state-of-the-art VPR methods; and Section VI provides discussion around our biologically-inspired model as well as future work.

II. RELATED WORK

This section outlines some key biological background for navigation in insect and mammalian brains, reviews the use of deep-learning-based approaches for VPR, and discusses recent developments in temporal filtering techniques for sequential data to further improve performance.

A. Navigation in Biological Brains and Robots

Our understanding of how animals navigate using vision has been used as an inspiration for designing effective localization, mapping and navigation algorithms. RatSLAM [21] is one example of this, using a model based on the rodent brain to perform visual SLAM over large real-world environments for the first time [22]. Likewise, researchers have developed a range of robotic navigation models based on other animals including insects [23]–[25].

Insects such as ants, bees and flies exhibit great capabilities to navigate [26]–[30]. In fact, their brains share the same general structure [26], [31], with the central complex being closely associated with navigation, orientation and spatial learning [32], [33]. Place recognition is, however, most likely mediated by processing within the *mushroom bodies* (MB), a separate pair of structures within their brains that are known to be involved in classification, learning, and recognition of both olfactory and visual information in bees and ants [32]. They receive densely coded and highly processed input from the sensory lobes, which then connects *sparsely* to a large number of intrinsic neurons within the MB. Their structure has been likened to a *multi-layer perceptron* (MLP) and considered optimal for learning and classifying complex input [34].

These impressive capabilities, achieved with relatively small brains, make them attractive models for roboticists. For FlyNet, we take inspiration from algorithmic insights found in the fruit fly olfactory neural circuit. Our focus here is primarily on taking high-level inspiration from the size and structure of the fly brain and investigating the extent to which it can be integrated with recurrent-based networks for VPR tasks, much as in the early RatSLAM work and related development [35].

B. Deep Neural Networks for Visual Place Recognition

CNN models have been applied to a range of image recognition tasks, including VPR, with great success across many challenging real-world datasets with both visual appearance and viewpoint changes [19], [36]–[38], and large-scale problems [39], [40]. Despite their success, these approaches often rely on the use of CNN models pre-trained on various computer vision datasets [5], [6], [36], [41]. Training these models in an end-to-end fashion specifically for VPR has also recently been proposed [4], [36], [42]. However, they are still using common network architectures, e.g. AlexNet [43], VGG [44] or ResNet [45], with slight changes to perform VPR tasks. All these systems share common undesirable characteristics with respect to their widespread deployability on real robots including large network sizes, extensive computing, and training requirements. In contrast, we propose the usage of compact neural models such as FlyNet to alleviate these requirements, while leveraging the temporal information found in most VPR datasets by using an equally compact CANN model.

C. Modeling Temporal Relationships

To access and exploit the power of temporal information in many applications, researchers have developed a range of RNN including long short-term memory (LSTM) [11]. These temporal-based approaches have been applied specifically to visual navigation [12] and spatial localization [13] in artificial agents. In a nice closure back to the inspiring biology, these approaches led to the emergence of grid-like representations, among other cell types found in mammalian brains [46], when training RNN cells to perform path integration [14] and navigation [16]. RatSLAM [21], one of the older approaches to filtering temporal information in a neural network, incorporated multi-dimensional CANN models with pre-assigned weights and structure set up to model the neural activity dynamics of place and grid cells found in the rat mammalian brain. Other non-neural techniques have been developed including SeqSLAM [10] in order to match sequences of pre-processed frames to provide an estimate of place, with a range of subsequent works [47]–[50].

The work to date has captured many key aspects of the VPR problem, investigating complex but powerful deep learning-based approaches, bio-inspired models that work in simulation or small laboratory mazes, and mammalian-brain based models with competitive real-world robotics performance. In this letter, we attempt to merge the desirable properties of several of these computer-science- and neuroscience-oriented models by developing a new bio-inspired, hybrid neural network for VPR tasks based on insect brain architectures such as FlyNet,

[1]Project page: `mchancan.github.io/projects/FlyNet`

which is extremely compact and can incorporate the filtering capabilities of a 1-*d* CANN to achieve competitive localization results. We also show how our compact FlyNet model can easily be adapted to other temporal filtering techniques including SeqSLAM and RNN.

III. METHOD OVERVIEW

We briefly describe recent development inspired by fruit fly brains such as the *fly algorithm* [51]. We then present our FlyNet algorithm (FNA) inspired by the *fly algorithm*, and propose our single-frame, multi-frame, and hybrid models.

A. Fly Algorithm

Recent research in brain-inspired computing suggests that *Drosophila* olfactory neural circuits identify odors by assigning similar neural activity patterns to similar input odors [51], [52]. These small brain cells perform a three-step procedure as the input odor goes through a three-layer neural circuit [51]. First, the firing rates across the first layer are centered to the same mean for all odors (removing the odor concentration dependence). Second, a binary, sparse random matrix connects the second layer to the third layer, where each neuron receives and sums about 10% of the firing rates from the second layer. Third, through a winner-take-all (WTA) circuit, only the highest-firing 5% neurons across the third layer are used to generate a specific binary tag of the input odor.

The *fly algorithm* is then proposed in [51] to mimic the pattern recognition capabilities found in the fly brain, at a broad level and from a functional computer science perspective. Being mathematically defined as a binary locality-sensitive hash (LSH) function; a new class of LSH algorithms (see Eq. 1) but with relevant differences such as requiring significantly fewer computations as it uses sparse, binary random projections instead of dense, Gaussian random projections typical in LSH functions [53].

$$Pr[h(p) = h(q)] = sim(p,q) \qquad (1)$$

where $sim(p,q)$ is the similarity function, and $h : \mathbb{R}^m \to \mathbb{Z}^n$ is the LSH function if for any $p, q \in \mathbb{R}^m$, Pr is $sim(p,q) \in [0,1]$.

B. Proposed FlyNet Algorithm

We leverage the *fly algorithm* from a computer vision perspective to propose our FlyNet algorithm (FNA), see Algorithm 1. The FNA mapping, shown in Fig. 3, uses a sampling ratio of 10% within the first layer, similar to the *fly algorithm*. A WTA circuit of 50% (instead of 5% as in the *fly algorithm*), is then used to generate a binary, compact output representation of our input image. Additional details on the choice and sensitivity of these parameters are provided in Section V-A. We also perform an image preprocessing step, to obtain **x**, before applying Algorithm 1. Details on this procedure are outlined in Section IV-A.

C. FlyNet-based Models

We implement a range of VPR models that leverage the FNA compact representations, including one single-frame model, and three multi-frame models with temporal filtering capabilities, see Fig. 4.

Algorithm 1 FlyNet Algorithm (FNA)

> **Input: x** $\in \mathbb{R}^m$
> **Output: y** $\in \mathbb{Z}^n$, $n < m$

1: Initialize $\mathbf{W} \in \mathbb{Z}^{n \times m}$: A binary, sparse random connection matrix between the input **x** and the output **y**.
2: Compute the output $\mathbf{y} = \mathbf{Wx}$: Each output y_j receives and sums 10% randomly selected input values x_i.
3: WTA circuit: Set the top 50% output values y_i to 1, and the remaining to 0.
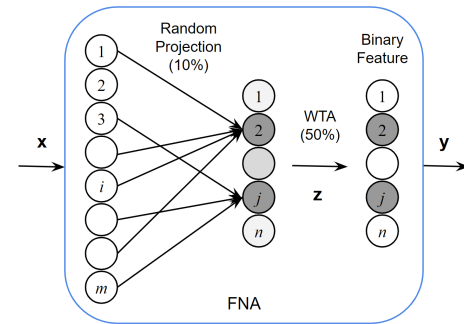

Fig. 3. **FNA mapping**. The *random projection* here shows only the connections to z_2 and z_j within the second layer, but all the units in that layer connect with 10% of the input units x_i.
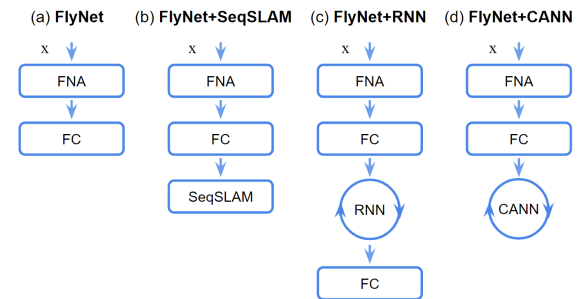

Fig. 4. **FlyNet baselines**. Proposed (a) single-frame and (b, c, d) multi-frame models including the (d) hybrid FlyNet+CANN neural network.

1) FlyNet: The FlyNet model, shown in Fig. 4 (a), is our bio-inspired two-layer neural network that comprises the FNA as a hidden layer, and a fully connected (FC) output layer. We configure FlyNet to have a gray-scale input image dimension of $m = 32 \times 64$, and an output dimension of $n = 64$. The FNA output **y** then feeds into a 1000-way *linear* MLP which computes a particular class score for each input image.

2) FlyNet+SeqSLAM: We incorporate the SeqSLAM algorithm [10] on top of our single-frame FlyNet network, as per previous research described in Sections I and II, see Fig. 4 (b). The resulting model is a multi-frame baseline which we can compare along with our other temporal filtering-based models FlyNet+RNN and FlyNet+CANN.

3) FlyNet+RNN: It is a purely neural model that incorporates a vanilla RNN on top of FlyNet for temporal information processing, see Fig. 4 (c). We also investigated the use of other types of RNN such as gated recurrent units (GRU) and LSTM. However, they showed no significant performance improvements despite having far more parameters.

4) ***FlyNet+CANN****:* It is our hybrid and also a purely neural model for sequence-based VPR tasks, see Fig. 4 (d). We implemented a variation of the CANN architecture introduced in the RatSLAM work [22], but using a 1-*d* CANN model proposed in [54], motivated by its suitability as a compact neural network-based way to implement the filtering capabilities of SeqSLAM [10]. As described in Section II-C, a CANN is a type of recurrent network that utilizes pre-assigned weights within its configuration. In Fig. 1 (middle) we show our detailed FlyNet+CANN implementation where, in contrast to an RNN, a unit within the CANN layer can excite or inhibit itself and units nearby using excitatory (arrows) or inhibitory (rounds) connections, respectively, and can also include a global inhibitor (GI) unit in its main structure. For this implementation, activity shifts in our 1-*d* CANN model, representing movement through the environment, were implemented with a direct shift and copy action. Although this could be implemented with more biologically faithful details such as velocity (V) units and asymmetric connections, as in prior CANN research [55].

IV. EXPERIMENTS

To evaluate the capabilities of our proposed FlyNet-based models, we conduct extensive experiments on two of the most widespread benchmarks used in VPR, the Nordland [56] and Oxford RobotCar [57] datasets. We compare FlyNet (alone) with other related single-frame VPR methods and neural networks. Furthermore, we also compare our hybrid, multi-frame neural network to three state-of-the-art, multi-frame VPR approaches: SeqSLAM [10], LoST-X [19], and Multi-Process Fusion (MPF) [20]. In this section, we describe in detail these network configurations and dataset preparation.

A. Real-World Datasets

1) ***Nordland****:* The Nordland dataset, introduced in [56] for VPR research, comprises four single traverses of a train journey, in northern Norway, including extreme seasonal changes across spring, summer, fall, and winter. This dataset is primarily used to evaluate generalization over visual appearance changes, as instantiated through its four-season coverage. In our experiments, we use three traverses to perform VPR at 1 fps as in [56]. We particularly use the summer traversal for training, and the remaining for testing, see Table I.

2) ***Oxford RobotCar****:* The Oxford RobotCar dataset [57] provides over 100 traverses with different lighting (e.g. day, night) and weather (e.g. direct sun, overcast) conditions through a car ride in Oxford city; which implicitly contains various challenges of pose and occlusions such as pedestrians, vehicles, and bicycles for instance. In our evaluations, we use the same subsets as in [19] with overcast (autumn) for training, and day/night for testing, see Table I.

Data Preprocessing. In all our experiments, we use a sequence of 1000 images per traversal (reference or query) and provide full resolution RGB images to all the models, being 1920×1080 for Nordland and 1280×960 for Oxford RobotCar. Our FlyNet baselines convert the images into single-channel (gray-scale) frames normalized between [0, 1], and

TABLE I
SEQUENCE-BASED DATASETS FOR VPR (REFERENCE/QUERY)

Dataset	Appearance Changes	Viewpoint Changes
Nordland Railway	Small (summer/fall) Extreme (summer/winter)	Small
Oxford RobotCar	Small (overcast/day) Extreme (overcast/night)	Moderate

then resize them to 32×64. While the state-of-the-art methods apply their default image preprocessing before feeding their models.

B. Evaluation Metrics

We evaluate the VPR performance of our models using precision-recall (PR) curves and area under the curve (AUC) metrics. The tolerance used to consider a query place as a correct match is being within 20 frames around the ground truth location for the Nordland dataset, and up to 50 meters (10 frames) away from the ground truth for the Oxford RobotCar dataset, as per previous research [19], [20], [58].

C. Comparison of FlyNet to other Neural Networks

We compare FlyNet (alone) with a range of related single-frame models including FC networks that use dropout [59], a vanilla CNN model often used in visual navigation research [60], [61], and the well-known NetVLAD method [36]. We train all these models end-to-end using a 1000-way *linear* MLP classifier—except for both the off-the-shelf NetVLAD backbone and the FNA layer in FlyNet (as its sparse matrix **W** stays unchanged). Average accuracy results over ten experiments using different seed numbers are shown in Fig. 5.

For FlyNet, we use its FC output layer as the *linear* classifier, as shown in Fig. 4 (a). For the FC networks, we use a three-layer MLP with 64–64–1000 units respectively, as in the FlyNet architecture. We then obtain the FC+Dropout network by using dropout rates of 90% and 50% for the first and second layers of the FC model, respectively, in order to approximate the FlyNet sparsity and for fair comparison purposes. For the CNN model, we use 2 *convolutional* layers but with gray-scale input images of 32×64 as in FlyNet. For NetVLAD, we use RGB images of 244×244, as required by its off-the-shelf VGG-16 [44] model, but we reduce their output representation dimensionality from 4096-*d* to 64-*d* to be comparable in size with the FlyNet representation. It is worth noticing that we do not reduce the CNN and NetVLAD model sizes down to the same size as FlyNet as they use pre-defined (rigid) architectures inherent to their approaches. We use the Adam optimizer [62] for training, and a learning rate set to 0.001 for all our experiments.

D. FlyNet Baselines Experiments

We trained and tested our four FlyNet baselines, described in Section III-C, in order to obtain our best performing model and compare it against existing state-of-the-art VPR methods. In Table II, we show the number of layers, weights, and units
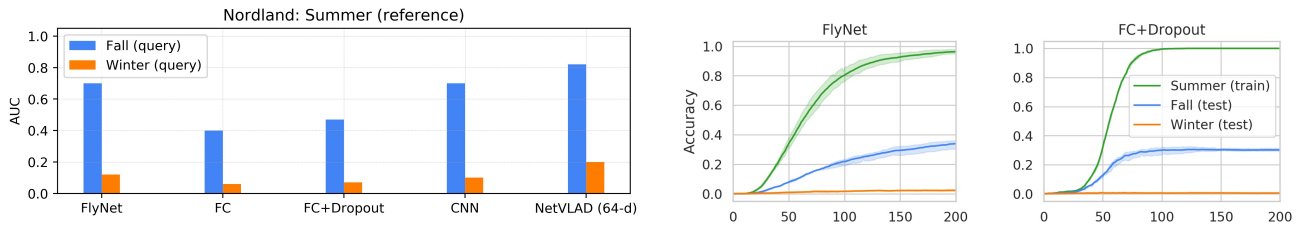
Fig. 5. **Comparison of FlyNet (alone) to other single-frame neural networks**. AUC results across different models on the Nordland dataset (left). Average accuracy over 10 training experiments vs. number of epochs for FlyNet (middle) and a fully connected (FC) network with dropout (right).
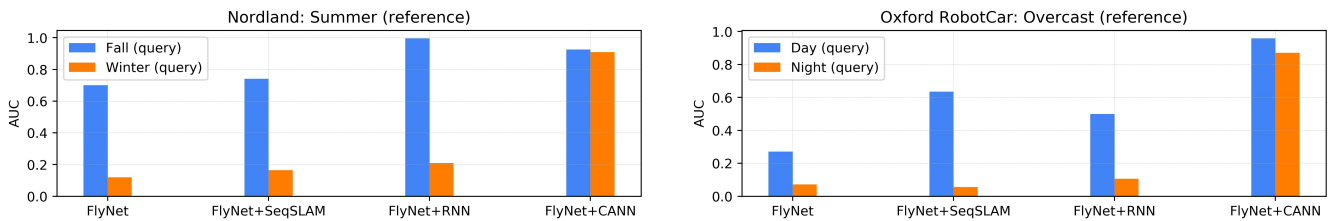

Fig. 6. **FlyNet baselines**. AUC results of single-frame and multi-frame FlyNet-based models on Nordland (left) and Oxford RobotCar (right) datasets.

for each model. For FlyNet and FlyNet+RNN, the FNA hidden layer used 64 units, and their FC layers used 1000 units, see Fig. 4 (a, c). The number of recurrent units for FlyNet+RNN was 512. For FlyNet+CANN, the CANN layer used 1002 units. We also show the AUC performance of our FlyNet baselines on both the Nordland and Oxford RobotCar datasets in Fig. 6 to further analyze these results in Section V-B.

TABLE II
FLYNET BASELINES FOOTPRINT

Architecture	# layers	# params	# neurons
FlyNet	2	64k	1064
FlyNet+RNN	4	1.3m	2576
FlyNet+CANN	3	72k	2066

E. Comparison to existing State-of-the-Art Methods

We compare our best performing FlyNet-based model with the algorithmic technique SeqSLAM (without FlyNet attached), and two deep-learning-based methods: LoST-X and Multi-Process Fusion.

1) SeqSLAM: SeqSLAM [10] shows state-of-the-art VPR results under challenging visual appearance changes. We use the MATLAB implementation in [56], with a sequence length of 20 frames, a threshold of 1, and the remaining SeqSLAM parameters using its default values.

2) LoST-X: The multi-frame LoST-X pipeline [19] uses visual semantics to perform VPR over day/night cycles, with further development for opposing viewpoints in [64]. LoST-X uses the RefineNet model [63], a ResNet-101-based model, as its semantic feature encoder, which is pre-trained on the Cityscapes dataset [65] for high-resolution segmentation.

3) Multi-Process Fusion (MPF): MPF [20] is also a multi-frame VPR technique. We use the VGG-16 network [44] trained on Places365 [9] to encode the images and feed the MPF sequence-based dynamic algorithm.

V. RESULTS

In this section, we analyze the experiments shown in Section IV, along with Figs. 5 and 6, and describe the results of PR curves and related AUC metrics for visual place recognition.

A. FlyNet vs. other Single-frame Networks and VPR Models

From Fig. 5 (left), we can see that FlyNet is directly competitive with both FC networks, despite FlyNet having over 3 times fewer parameters (64k vs. 199k). Potentially using 32 times less memory, as the FNA layer require only 1-bit per binary weight, as per previous research [66], compared to the corresponding layer using 32-bit floating-point weights in the FC models. On the other hand, for the CNN and NetVLAD models, with 6 and 234 times more parameters than FlyNet respectively, the larger the model the better the results we obtained. Under small environmental changes (e.g. summer to fall) both networks achieved over 70% AUC, comparable to FlyNet. However, under extreme visual changes (e.g. summer to winter) all these models show relatively similar results, below 12% AUC, except for NetVLAD with 20% AUC.

In Fig. 5 (middle, right), we show in further detail the average training results of FlyNet against the FC model with dropout across 200 epochs. Additional experiments to support the choice of the FlyNet parameters (e.g. sampling ratio of 10% and WTA circuit of 50%) were also conducted. For the sampling ratio, we gradually increased it from 10% to 90% but no further accuracy improvement than 96% was obtained. For the WTA parameter, we varied it between 5% and 95% but, as we moved away from 50% WTA, the training accuracy decreased to 25% and 40%, respectively.

B. FlyNet Baselines Evaluations

Although there are significant performance differences at a single-frame matching level, Fig. 6 shows that when using sequence-based filtering techniques these differences reduce significantly. Meaning that using the more compact networks
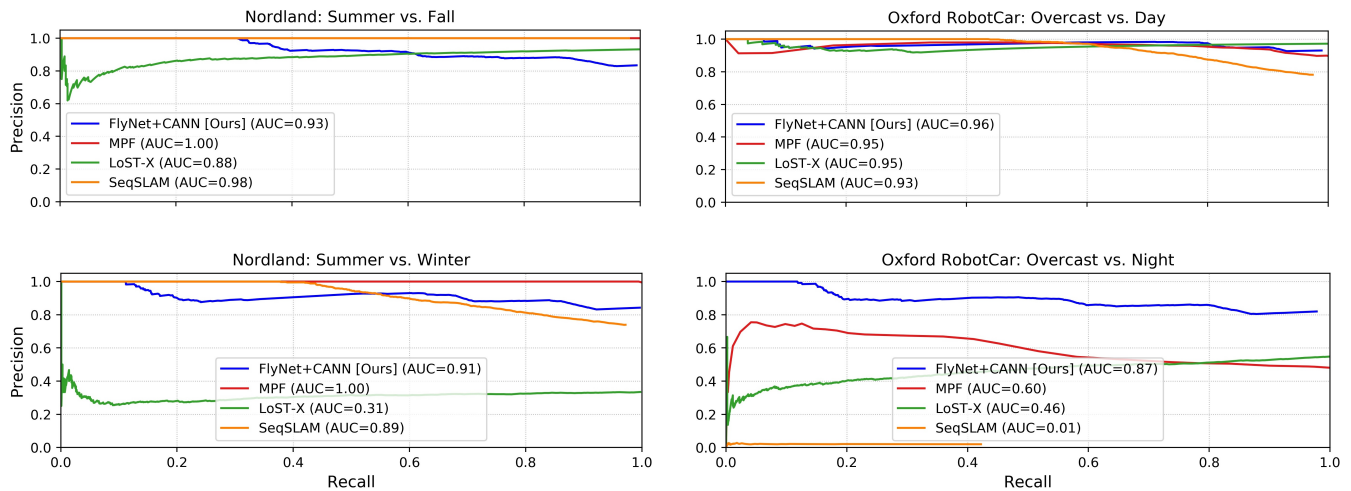
Fig. 7. **PR performance** of FlyNet+CANN vs. SeqSLAM, LoST-X and MPF on 1000-places of the Nordland (left) and Oxford RobotCar (right) dataset.
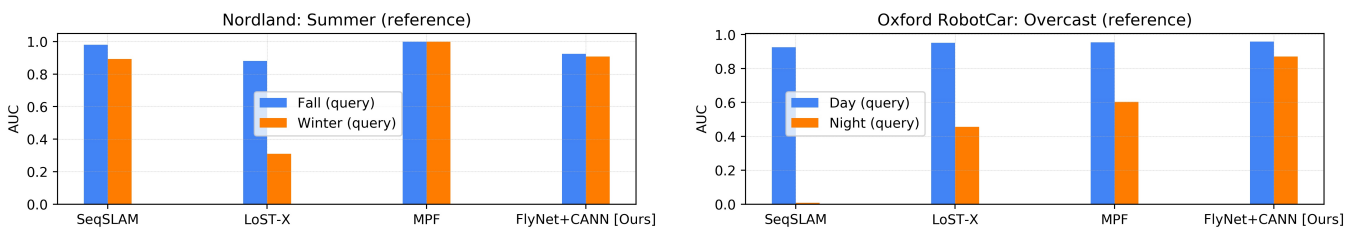

Fig. 8. **AUC results** of FlyNet+CANN compared to SeqSLAM, LoST-X, and MPF on the Nordland (left) and Oxford RobotCar (right) dataset.

is viable in a range of applications where temporal filtering is practically feasible. It is possible then to leverage our compact FlyNet network and integrate it with a range of sequence-based methods such as SeqSLAM, RNN or CANN models and achieve competitive results. For FlyNet+SeqSLAM, the performance of FlyNet (alone) was significantly improved (see Fig. 6). Similarly, the RNN layer on top of FlyNet improved even further these results. However, when integrating the output of FlyNet with a 1-d CANN we were able to outperform these models, even under extreme environmental changes (e.g. day to night, summer to winter); we then choose this hybrid approach to compare against existing state-of-the-art methods.

C. State-of-the-Art Analysis

Figs. 7 and 8 show quantitative results for FlyNet+CANN and state-of-the-art VPR methods. Fig. 7 (left) shows the PR performance curves on the Nordland dataset, where MPF is performing better while being able to recall almost all places at 100% precision on both fall and winter testing traverses. Achieving also the highest AUC results, see Fig. 8 (left). On the other hand, the semantic-based approach LoST-X can recall a few matches at 100% precision on both testing traverses (fall and winter). In contrast, FlyNet+CANN achieves state-of-the-art results comparable with SeqSLAM and MPF in all these tested traverses, see Fig. 8 (left).

Similarly, PR performance on the Oxford RobotCar dataset is shown in Fig. 7 (right). Also notable in this case is that FlyNet+CANN again achieves state-of-the-art results that are now comparable with SeqSLAM, LoST-X, and MPF approaches. Our hybrid model consistently maintains its PR

TABLE III
PROCESSING TIME COMPARISON ON THE NORDLAND DATASET

VPR System	Feature Ext.	Place Match.	Avg. Time (fps)
FlyNet+CANN	**35 sec**	**25 sec**	**0.06 sec (16.66)**
MPF	1.9 min	4.6 min	0.39 sec (2.56)
LoST-X	110 min	200 min	18.6 sec (0.05)
SeqSLAM	50 sec	40 sec	0.09 sec (11.11)

performance even under extreme environmental changes (e.g. overcast to night), see Fig. 7 (right-bottom). In Fig. 8 (right), we also show how FlyNet+CANN outperforms the remaining methods in terms of AUC results, and Fig. 9 shows qualitative generalization results on both datasets.

D. Computational Performance

The processing time required to perform appearance-invariant VPR by our hybrid model is compared to those from state-of-the-art methods in terms of running time for (1) feature extraction, (2) visual place matching between query and reference traverses, and (3) average place recognition time for a single query image from a 1000-image reference database. This *Avg. Time (3)* is calculated as (*Feature Ext. (1) + Place Match. (2)*)/1000. Processing time results on the Nordland dataset are reported in Table III, where we show that our hybrid approach can be up to 6.5, 310, and 1.5 times faster than MPF, LoST-X, and SeqSLAM, respectively.

Fig. 10 shows a similar comparison presented in Fig. 2 but with moderated appearance changes (overcast to day) on the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2020.2967324, IEEE Robotics and Automation Letters

CHANCÁN *et al.*: A HYBRID COMPACT NEURAL ARCHITECTURE FOR VISUAL PLACE RECOGNITION 7

Fig. 9. **Generalization results**. Sample images (reference) of the Nordland summer (left-top) and Oxford RobotCar overcast traverses (right-top). Corresponding frames retrieved (query) using our FlyNet+CANN model from the Nordland winter (left-bottom) and Oxford RobotCar night traverses (right-bottom).
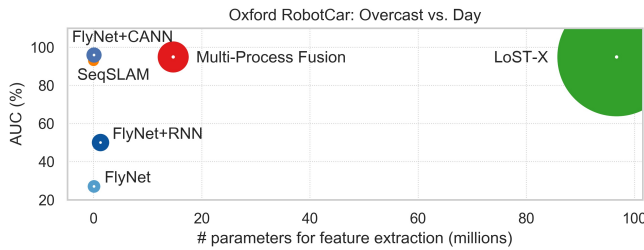

Fig. 10. **Oxford RobotCar AUC performance vs. Model Size**. Similar to Fig. 1, it compares small appearance changes (overcast vs. day).

Oxford RobotCar dataset. In this figure, again, the area of the circle is proportional to the number of layers per model, except for SeqSLAM which performs an algorithmic matching procedure. State-of-the-art methods like MPF, LoST-X, and SeqSLAM achieve better AUC results than in Fig. 2 with 95%, 95% and 93% respectively, where FlyNet+CANN also shows competitive results with 96% AUC.

E. Influence of Bio-inspiration

In Figs. 7–10 and Table III, we show how our proposed FlyNet+CANN model achieves competitive visual localization results compared to existing deep learning and algorithmic-based VPR techniques, but with significantly fewer parameters, a smaller footprint and reduced processing time. Although we could have used conventional, pre-trained CNN models instead of FlyNet, our objective is to demonstrate to what extent we can draw inspiration from the brain's structural and functional connections between neural cells. Making possible to develop a sample-efficient, high-performing hybrid neural model, which structure is aligned with algorithmic insights found in the brain, as outlined in previous work [67], [68], but for VPR tasks.

For FlyNet, it has the same number of layers and sparse structure found in the fly olfactory neural circuit. Although the fly brain expands the dimensionality of their input odor [51], e.g. from m to $n = 40m$ (see Fig. 3). We experimentally found that by reducing this dimension, e.g. from m to $n = m/32$ instead, the FlyNet training accuracy remained around 96%, as shown in Fig. 5 (middle), while preserving the desired compact network structure.

For FlyNet+CANN, the integration of a 1-d CANN model, for temporally filtering the output of FlyNet, enabled the use of a relatively low-performance but fast network to get better VPR results for our whole hybrid model, which is also able to generalize across challenging environmental changes (see Fig. 9), while being up to three orders of magnitude faster than existing VPR methods, see Table III.

VI. CONCLUSION

We proposed a new bio-inspired, hybrid model for visual place recognition based by part on the fruit fly brain and integrated with a compact neurally-inspired continuous attractor neural network. Our model was able to achieve competitive place recognition performance and generalize over challenging environmental variations (e.g. day to night, summer to winter), compared to state-of-the-art approaches that have much larger network size and computational footprints. It was also, to the best of our knowledge, the furthest in capability an insect-based place recognition system has been pushed with respect to demonstrating real-world appearance-invariant VPR without resorting to full deep learning architectures.

Future research bridging the divide between well-characterized insect neural circuits [69], [70] as well as recent deep neural network approaches and computational models of network dynamics related to spatial memory and navigation [71] are likely to yield further performance and capability improvements, and may also shed new light on the functional purposes of these biological neural networks.

REFERENCES

[1] S. Lowry *et al.*, "Visual Place Recognition: A Survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[3] Z. Chen *et al.*, "Convolutional Neural Network-based Place Recognition," in *Proc. Australas. Conf. Robot. Autom.*, 2014.

[4] Z. Chen *et al.*, "Deep learning features at scale for visual place recognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 3223–3230, 2017.

[5] N. Sünderhauf *et al.*, "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free," in *Proc. Robot.: Sci. Syst.*, 2015.

[6] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the Performance of ConvNet Features for Place Recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 4297–4304, 2015.

[7] Z. Xin *et al.*, "Real-time Visual Place Recognition Based on Analyzing Distribution of Multi-scale CNN Landmarks," *J. Intell. Robot. Syst.*, vol. 94, pp. 777–792, 2018.

[8] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 248–255, 2009.

[9] B. Zhou *et al.*, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[10] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1643–1649, 2012.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2020.2967324, IEEE Robotics and Automation Letters

8   IEEE ROBOTICS AND AUTOMATION LETTERS. PREPRINT VERSION. ACCEPTED DECEMBER, 2019

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[12] A. Banino *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, no. 7705, pp. 429–433, 2018.

[13] C. J. Cueva and X.-X. Wei, "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[14] B. McNaughton *et al.*, "Path integration and the neural basis of the 'cognitive map'," *Nat. Rev. Neurosci.*, vol. 7, pp. 663–678, 2006.

[15] L. M. Giocomo, M. Moser, and E. I. Moser, "Computational Models of Grid Cells," *Neuron*, vol. 71, pp. 589–603, 2011.

[16] B. Sorscher *et al.*, "A unified theory for the origin of grid cells through the lens of pattern formation," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 10003–10013, 2019.

[17] J. Pei *et al.*, "Towards artificial general intelligence with hybrid Tianjic chip architecture," *Nature*, vol. 572, pp. 106–111, 2019.

[18] Z. Yang *et al.*, "DashNet: A Hybrid Artificial and Spiking Neural Network for High-speed Object Tracking," *ArXiv*, abs/1909.12942, 2019.

[19] S. Garg, N. Süenderhauf, and M. Milford, "LoST? Appearance-Invariant Place Recognition for Opposite Viewpoints using Visual Semantics," in *Proc. Robot.: Sci. Syst.*, 2018.

[20] S. Hausler, A. Jacobson, and M. Milford, "Multi-Process Fusion: Visual Place Recognition Using Multiple Image Processing Methods," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1924–1931, 2019.

[21] M. J. Milford *et al.*, "RatSLAM: a hippocampal model for simultaneous localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, pp. 403–408, 2004.

[22] M. J. Milford and G. F. Wyeth, "Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1038–1053, Oct. 2008.

[23] A. Cope *et al.*, "The green brain project—Developing a neuromimetic robotic honeybee," in *Biom. and Biohybrid Syst.*, pp. 362–363, 2013.

[24] B. Webb, "Using robots to model animals: a cricket test," *Robotics and Autonomous Systems*, vol. 16, no. 2, pp. 117–134, 1995.

[25] J. Dupeyroux *et al.*, "Antbot: A six-legged walking robot able to home like desert ants in outdoor environments," *Science Robotics*, vol. 4, 2019.

[26] A. B. Barron and J. A. Plath, "The evolution of honey bee dance communication: a mechanistic perspective," *J. Exp. Biol.*, vol. 220, no. 23, pp. 4339–4346, 2017.

[27] A. Narendra *et al.*, "Mapping the navigational knowledge of individually foraging ants, *Myrmecia croslandi*," in *Proc. R. Soc. B.*, vol. 280, no. 1765, 2013.

[28] J. Degen *et al.*, "Exploratory behaviour of honeybees during orientation flights," *Animal Behaviour*, vol. 102, pp. 45–57, 2015.

[29] T. Warren, Y. Giraldo, and M. Dickinson, "Celestial navigation in *Drosophila*," *J. Exp. Biol.*, vol. 222, 2019.

[30] T. A. Ofstad, C. S. Zuker, and M. B. Reiser, "Visual place learning in *Drosophila melanogaster*," *Nature*, vol. 474, pp. 204–209, 2011.

[31] J. Plath and A. Barron, "Current progress in understanding the functions of the insect central complex," *Current Opinion in Insect Science*, vol. 12, pp. 11–18, 2015.

[32] J. Plath *et al.*, "Different roles for honey bee mushroom bodies and central complex in visual learning of colored lights in an aversive conditioning assay," *Frontiers in Behavioral Neuroscience*, vol. 11, 2017.

[33] K. Pfeiffer and U. Homberg, "Organization and functional roles of the central complex in the insect brain," *Annual Review of Entomology*, vol. 59, pp. 165–184, 2014.

[34] R. Huerta, "Learning pattern recognition and decision making in the insect brain," *AIP Conf. Proc.*, vol. 1510, no. 1, pp. 101–119, 2013.

[35] F. Yu *et al.*, "NeuroSLAM: a brain-inspired SLAM system for 3D environments," *Biol. Cybern.*, vol. 113, no. 5, pp. 515–545, 2019.

[36] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5297–5307, 2016.

[37] H. Noh *et al.*, "Large-Scale Image Retrieval with Attentive Deep Local Features," in *IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3476–3485.

[38] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 Place Recognition by View Synthesis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, pp. 257–271, 2015.

[39] T. Sattler *et al.*, "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[40] M. A. Esfahani, K. Wu, S. Yuan, and H. Wang, "DeepD-SAIR: Deep 6-DOF Camera Relocalization using Deblurred Semantic-Aware Image Representation for Large-Scale Outdoor Environments, *Image and Vision Computing*, vol. 89, pp. 120–130, 2019.

[41] J. Long *et al.*, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[42] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli, "Learning context flexible attention model for long-term visual place recognition," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4015–4022, Oct. 2018.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1097–1105, 2012.

[44] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ArXiv*, abs/1409.1556, 2015.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016.

[46] E. I. Moser, E. Kropff, and M.-B. Moser, "Place Cells, Grid Cells, and the Brains Spatial Representation System," *Annual Review of Neuroscience*, vol. 31, no. 1, pp. 69–89, 2008.

[47] T. Naseer *et al.*, "Robust visual robot localization across seasons using network flows," in *Proc. AAAI Conf. Artif. Intell.*, pp. 2564–2570, 2014.

[48] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1645–1661, 2013.

[49] E. Pepperell *et al.*, "All-environment visual place recognition with SMART," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1612–1618, 2014.

[50] Y. Li *et al.*, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 353–361, 2015.

[51] S. Dasgupta *et al.*, "A neural algorithm for a fundamental computing problem," *Science*, vol. 358, no. 6364, pp. 793–796, 2017.

[52] C. Pehlevan, A. Genkin, and D. B. Chklovskii, "A clustering neural network model of insect olfaction," *2017 51st Asilomar Conf. Signals, Systems, and Computers*, pp. 593–600, 2017.

[53] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," *ArXiv*, abs/1408.2927, Aug. 2014.

[54] P. Miller, "Dynamical systems, attractors, and neural circuits," *F1000 Research*, vol. 5, 2016.

[55] P. Stratton *et al.*, "Using strategic movement to calibrate a neural compass: A spiking network for tracking head direction in rats and robots," *PLOS ONE*, vol. 6, no. 10, pp. 1–15, 2011.

[56] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons," in *Proc. Workshop Long-Term Autonomy IEEE Int. Conf. Robot. Autom.*, 2013.

[57] W. Maddern *et al.*, "1 Year, 1000km: The Oxford RobotCar Dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.

[58] J. Mao *et al.*, "Learning to Fuse Multiscale Features for Visual Place Recognition," in *IEEE Access*, vol. 7, pp. 5723–5735, 2019.

[59] N. Srivastava *et al.*, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[60] P. Mirowski *et al.*, "Learning to navigate in complex environments," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[61] L. Espeholt *et al.*, "Impala: Scalable Distributed Deep-RL with IMPortance eighted Actor-Learner Architectures," in *Proc. Int. Conf. Mach. Learn.*, 2018.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv*, abs/1412.6980, 2014.

[63] G. Lin *et al.*, "RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[64] S. Garg *et al.*, "Semantic-geometric visual place recognition: a new perspective for reconciling opposing views," *Int. J. Robot. Res.*, 2019.

[65] M. Cordts *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[66] I. Hubara *et al.*, "Binarized Neural Networks," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 4107–4115, 2016.

[67] K. Xu *et al.*, "What Can Neural Networks Reason About?," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[68] M. Lechner *et al.*, "Designing Worm-inspired Neural Networks for Interpretable Robotic Control," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 87–94, 2019.

[69] L. Hernandez-Nunez *et al.*, "Reverse-correlation analysis of navigation dynamics in *Drosophila* larva using optogenetics," *eLife*, vol. 4, 2015.

[70] M. E. Berck *et al.*, "The wiring diagram of a glomerular olfactory system," *eLife*, vol. 5, 2016.

[71] M. G. Campbell *et al.*, "Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation." *Nature Neurosci.*, vol. 21, pp. 1096–1106, 2018.