

Applications of Spiking Neural Networks in Visual Place Recognition

Somayeh Hussaini, *Member, IEEE*, Michael Milford, *Senior Member, IEEE*, and Tobias Fischer, *Senior Member, IEEE*

Abstract—In robotics, Spiking Neural Networks (SNNs) are increasingly recognized for their largely-unrealized potential energy efficiency and low latency particularly when implemented on neuromorphic hardware. Our paper highlights three advancements for SNNs in Visual Place Recognition (VPR). Firstly, we propose Modular SNNs, where each SNN represents a set of non-overlapping geographically distinct places, enabling scalable networks for large environments. Secondly, we present Ensembles of Modular SNNs, where multiple networks represent the same place, significantly enhancing accuracy compared to single-network models. Each of our Modular SNN modules is compact, comprising only 1500 neurons and 474k synapses, making them ideally suited for ensembling due to their small size. Lastly, we investigate the role of sequence matching in SNN-based VPR, a technique where consecutive images are used to refine place recognition. We analyze the responsiveness of SNNs to ensembling and sequence matching compared to other VPR techniques. Our contributions highlight the viability of SNNs for VPR, offering scalable and robust solutions, and paving the way for their application in various energy-sensitive robotic tasks.

Index Terms—Neurorobotics, Localization, Biomimetics, Visual Place Recognition

I. INTRODUCTION

SPIKING Neural Networks (SNNs) represent a cutting-edge paradigm in neuromorphic computing, mirroring the intricate workings of biological neural systems [1]–[4]. In these networks, every neuron possesses its distinct activation state. Unlike conventional neural networks, where neuron activations are typically continuous values, neurons in SNNs convey information through intermittent spikes, which are initiated when the neuron’s activation surpasses a particular threshold [1], [5], [6]. These spiking networks exhibit promising attributes when deployed on neuromorphic hardware, offering notable energy efficiency and low-latency data processing [2], [7]–[10]. Despite these potential advantages, SNNs have seen minimal adaptation in robotics, due to limitations such as difficulty in supervised training of SNNs due to the non-differentiable activation function of spiking neurons and a lack of tools and resources [2], [4], [6], [11], [12].

One robotics application that could benefit substantially from the emerging neuromorphic computing paradigm is

This work received funding from Intel Labs to TF and MM, and from ARC Laureate Fellowship FL210100156 to MM. The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. The authors would like to thank Dr. Adam Hines, Therese Joseph, and Connor Malone for their valuable insights on the drafts of this manuscript, and the QUT eResearch services for providing computational resources via the QUT HPC system. (*Corresponding Author: Somayeh Hussaini*)

The authors are with the QUT Centre for Robotics, School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, QLD 4000, Australia (e-mail: {s.hussaini, michael.milford, tobias.fischer}@qut.edu.au).

the Visual Place Recognition (VPR) task, a vital process in robotic navigation. At its core, the objective is seemingly straightforward: given a query image of a place, find the corresponding place out of a potentially very large list of previously visited places, also called the reference dataset [13]–[18]. However, there are immense underlying challenges such as changes in appearance due to different times of the day, variations in seasons, weather conditions, and perceptual aliasing (where two geographically distant places may look very similar), leading to significant appearance discrepancies between reference and query images of the same place [14], [15], [17].

Visual place recognition is a critical component in robot localization tasks such as loop closure detection in SLAM, and global re-localization of mobile robots [13], [16], [18], [19]. It is also relevant to image retrieval and landmark recognition tasks [13], [15]. Within robot navigation, VPR can minimize localization errors by recognizing previously visited places and updating the map of the environment despite appearance changes [13], [15], [20], which enables mobile robots to operate over extended periods.

To operate in real-time on resource-constrained robots, such as space exploration and disaster recovery where long mission times are desirable, conventional SOTA VPR methods might not be applicable as they often have high computational demands [21], motivating the use of SNNs within VPR. We are further inspired by the remarkable ability of animals with relatively small brains, such as rodents, to effectively perform navigation in complex environments. Although SNNs are not as mature as their deep learning counterparts, the benefits of SNNs include low power usage and low latency, particularly when deployed on neuromorphic hardware [2], [3], [8]. VPR is a particularly intriguing task for SNNs as VPR is amenable to enhancement strategies such as ensembling and using temporal data via sequences of images.

The objective of our work is to explore an alternative SNN-based approach to current state-of-the-art VPR techniques based on deep learning that is scalable and efficient, and has the prospect to be ideally suited for energy-sensitive robotic tasks. We achieve this by demonstrating the potential of a simple three-layer SNN in achieving place recognition at a significant scale. Our approach leverages widely adopted strategies for enhancing the robustness of conventional machine learning approaches, specifically focusing on modularity, ensembling, and sequence matching techniques. The subsequent paragraphs will offer a succinct overview of these key strategies.

Modularity, a prominent concept in machine learning, entails the design of systems comprising distinct modules, each dedicated to a specific task [22]–[24]. These modules can be combined to form more intricate systems, offering scalability

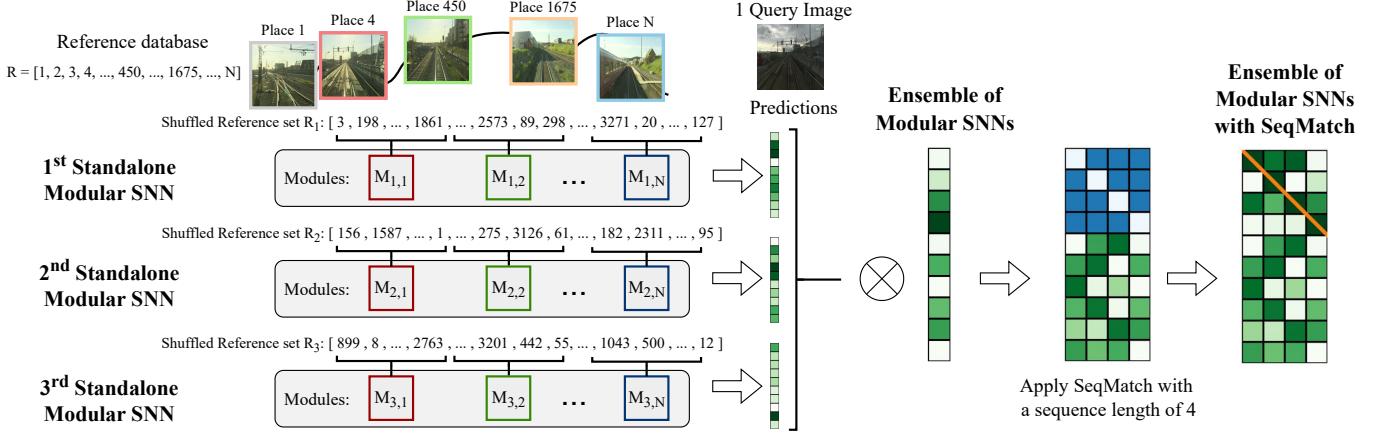


Fig. 1. Schematic of the proposed algorithm: The basic building blocks in our work are independent Spiking Neural Network (SNN) modules that learn small subsets of the reference database. At inference time, the place predictions of all these modules are fused in parallel in what we dub a “Standalone Modular SNN”, enabling the scalability of our approach to a large number of places. We further make use of the potential massively parallel processing capabilities of neuromorphic processors by introducing ensembles in which multiple Modular SNNs learn representations for the same place, and demonstrate that SNNs are more responsive to ensembling compared to conventional techniques. Finally, we demonstrate the high responsiveness of these Ensembles of Modular SNNs to sequence matching.

benefits beyond what individual modules can achieve [22]–[24]. Modularity has been employed in [25] for a SLAM system with heterogeneous sensor configuration, a 3D place recognition task [26], a condition and environment-invariant place recognition task [27], and a SLAM system [28]. These works assign a module to each structurally different sub-task of a system. In our work, we similarly use the concept of modularity to enable robustness to more challenging scenarios, however we create modules with the same architecture which are assigned to learn small segments of the dataset.

Ensembling¹ is the strategy of combining multiple models to boost accuracy, reduce overfitting, and yield more robust models [1], [29], [31]–[34]. While there is an inherent trade-off between the number of ensemble members and the energy efficiency of a system [29], [31], the benefits of ensembling, particularly its enhanced accuracy and reliability, are invaluable to navigation-related robotics applications. Ensembling has been previously utilized in the context of monocular SLAM [35] for improved data association, terrain segmentation for learning data collected at different times [36], and place recognition with image-based [37], [38] and event-based [39] input, where multiple VPR techniques fuse to enhance place recognition robustness. Similarly, our work also utilizes ensembling to improve upon the robustness and generalization ability of our system but differs with these works in terms of how we define our ensemble members. Our ensemble members all are tasked to do the same task, have the same architecture and differ in random shuffling of input images and initial learned weights. A similar approach of employing ensembles was previously demonstrated in uncertainty estimation to provide sufficient diversity among the members, and improve the overall performance of the system [40].

In the context of VPR, a sequence matching technique

uses consecutive reference frames instead of single frames, to match a query image of a place to its corresponding reference image [41]–[43]. By analyzing a series of images, this technique enables enhanced resilience against temporary environmental disruptions and improves localization accuracy [41]–[43]. Sequence matching techniques have been widely employed in place recognition [41]–[47], where often a decoupled approach consisting of an initial image-based retrieval method and subsequent sequence score aggregation [41], [44], [47] is employed. Similar to these techniques, our paper utilizes sequence matching as an additional step after image retrieval. We further demonstrate the effectiveness of sequence matching for SNN-based approaches, and provide an indicator for responsiveness of conventional and SNN-based VPR methods to sequence matching.

In this work, we claim the following contributions (Figure 1):

- 1) We present the concept of modular spiking neural networks (Modular SNNs) for scalable place recognition. Each module of the network *specializes* in a small subset of places in the environment at training time and operates *independently* of all other networks. After training the modules independently, we address the lack of *global* regularization by detecting hyperactive neurons, those that frequently respond to images *outside* their training data, and subsequently ignoring them during deployment. The query image at inference time is provided to all modules in parallel, and their place predictions are fused.
- 2) While the first contribution serves as a functional framework for scalable place recognition with Modular SNNs, this second contribution further enhances its capabilities through Ensembles of Modular SNNs, where *multiple* networks learn a representation for the *same place*, leading to improved robustness and generalization capabilities. Each ensemble member constitutes a Modular SNN with variations in the weight initialization and the set of distinct places learned by a module. Our results demonstrate that SNN ensemble members exhibit

¹We note ensembling [29] and model fusion [30] have been used interchangeably in literature to denote the practice of combining multiple models or feature representations for improved performance. Throughout this work, we refer to this technique as ensembling.

higher variations in their match predictions compared to conventional counterparts, which results in significantly higher responsiveness to ensembling.

- 3) We analyze the responsiveness of our Modular and Ensemble of Modular SNNs to sequence matching, which captures the temporal information inherent in the data, by considering multiple consecutive reference places for predicting a single query image, as opposed to considering one single reference image. We also present an indicator that predicts the effectiveness of applying sequence matching to both conventional VPR methods and our spiking networks, to provide insights into the improvements conferred by applying a sequence matching technique.
- 4) We provide extensive evaluations of our SNN performance, and compare them to conventional VPR techniques, i.e. Sum-of-absolute differences (SAD) [41], DenseVLAD [48], NetVLAD [49], AP-GeM [50], GCL [51], CosPlace [52], and MixVPR [53] across multiple benchmark datasets, namely Nordland [54], Oxford Robot-Car [55], SFU-Mountain [56], Synthia Night to Fall [57], and St Lucia [58]. Compared to previous work [59], we evaluate our approach on datasets that are up to two orders of magnitude larger. We also introspect our SNNs and provide insights to their responsiveness when paired with sequence matching, and contrast that to that of conventional VPR techniques. Finally, we provide a proof-of-concept deployment of our Modular SNN on AgileX’s Scout Mini robot [60] in a small indoor environment, operating in real-time on CPU.

The first contribution on Modular SNNs was previously presented at the IEEE International Conference on Robotics and Automation (ICRA) 2023 [61]. This work substantially extends on [61] by employing ensembling and sequence matching techniques to significantly enhance place recognition capabilities and analyzing the impact of these techniques across the entire set of evaluated models. We introduce an indicator that predicts the responsiveness of both conventional VPR techniques and our spiking networks to sequence matching. We also provide significantly extended evaluations, now covering six datasets, up from the initial two. Furthermore, we demonstrate the scalability of our approach by applying it to datasets that are up to an order of magnitude larger in terms of the number of learned places, and we do not require a calibration step anymore, which previously required paired images for a subset of the query/reference datasets. For the first time, we demonstrate our Modular SNN in a real-time proof-of-concept CPU-based robot deployment in a small indoor environment. We also benchmark against seven distinct VPR techniques, encompassing 14 variations, compared to the three techniques previously compared against in [61]. Our new contributions demonstrate the enhanced effectiveness of SNNs for VPR, including increased scalability and robustness, paving the way for application in energy-efficient robotic navigation tasks.

In the following sections, we will delve into the related works (Section II) to provide context for spiking neural networks and visual place recognition, discuss our methodology (Section III),

and experimental setup (Section IV), present our results with analysis (Section V), and propose future research directions (Section VI), offering a thorough exploration of spiking neural networks for scalable visual place recognition.

II. RELATED WORKS

In this section, we offer an overview of neuromorphic computing and spiking neural networks (Section II-A) and explore the applications of spiking neural networks (SNNs) in robot localization (Section II-B). We then delve into the visual place recognition task (Section II-C), non-spiking biologically inspired VPR approaches (Section II-D) and provide insights into ensembling (Section II-E) and sequence matching techniques (Section II-F).

A. Neuromorphic Computing and Spiking Neural Networks

The field of neuromorphic computing focuses on hardware, sensors, and algorithms inspired by biological neural networks, aiming to capture the robustness, generalization capability, energy efficiency and low latency advantages seen in nature [2], [7], [8]. The fundamental properties of neuromorphic computing that enable its advantageous features encompass highly parallel operations, the integration of processor and memory components, and the utilization of asynchronous event-driven computation with sparse temporal activity [2], [3], [8].

SNNs represent a set of algorithms within the realm of neuromorphic computing, transferring information through discrete spikes instead of the continuous values used by artificial neural networks [1], [5], [6]. SNN architectures typically comprise interconnected neurons linked via synapses (weights), which transfer information from presynaptic (source) neurons to postsynaptic (target) neurons [6]. Each neuron has its own internal state, allowing computation at neuron and synapse levels to be parallelized when deployed on neuromorphic hardware, which optimizes data transfer via collocating processing and memory [21], [62]. Deploying SNNs on neuromorphic hardware enables sparse and asynchronous event-driven processing, significantly reducing power consumption and latency [2], [8], [9].

Various common approaches exist for implementing SNNs, which we mention here to acknowledge their existence. One approach is to train an Artificial Neural Network (ANN) using back-propagation and convert the trained model to an SNN for inference. Strategies for this conversion include activation function approximations [63], and optimizations [64], [65], constrained training to resemble spiking form [66], [67], and optimized spiking neuron models [68]. Due to limited weight precision, these approaches often do not fully utilize the inherent energy efficiency of SNNs [3]. The neuronal dynamics of spiking neurons are non-differentiable which means that back-propagation cannot be directly applied to train SNNs for complex tasks. To address this, a number of works have focused on approximating back-propagation for SNNs [69]–[71]. However, these methods require offline training with large datasets and perform poorly in continual learning settings due to catastrophic forgetting [2]. Lastly, some training paradigms are inspired by the modulation of synaptic strength (weight) based

on neuronal activity [72]. Among these, Spike-Time Dependent Plasticity (STDP) updates weights according to the relative timing of spikes received from presynaptic and postsynaptic neurons [3], [72]. In this study, we employ the unsupervised STDP as our primary training algorithm for SNNs.

We acknowledge the challenges in achieving competitive accuracy with these unsupervised learning approaches. To address this, we introduce a novel methodology that combines STDP with a regularization term to detect and remove *hyperactive* neurons, and incorporates enhancement techniques familiar to the conventional machine learning domain, namely modularity, ensembling, and sequence matching. This approach aims to enhance SNN accuracy, robustness to significant appearance changes, and scalability in learning a larger number of places.

B. Spiking Neural Networks for Robot Localization

The capabilities of SNNs have been demonstrated in a wide range of computer vision and robotics tasks. These include pattern recognition [73], control [74]–[78], manipulation [79]–[81], object tracking [82], [83], and scene understanding [84], [85]. Many works have employed SNNs to address tasks related to robot localization, which is the problem under investigation in this work. These works include RatSLAM [86] which computationally models place, grid and border cells of the rat hippocampus [87], a navigation controller for mapping unknown environments [88], a pose estimation and map formation method [89], a light-weight system for uni-dimensional SLAM [90], and a SLAM model that utilizes representations of continuous spatial maps to produce compressed structures from multiple domains [91]. In previous work [59], we presented a SNN specifically for VPR. This network had a limited capacity, recognizing up to just 100 places.

While some of these systems have also been deployed on neuromorphic hardware [87], [89], [90], [92], [93], so far, the performance of SNN-based methods for robot localization have only been demonstrated in simulated environments [88], [89], [91], constrained indoor environments [87], [90], [92]–[94], or small-scale outdoor environments [59].

In addition to spike-based *processing*, the use of event-based cameras for spike-based *sensing* has shown promising advantages in robotic navigation systems [95]–[97], SLAM systems [92]–[94], and place recognition [98], owing to their unique ability to output asynchronous pixel-level brightness changes rather than conventional images, having a high dynamic range and remaining unaffected by motion blur [99]. Although our research presently focuses on conventional image frames, the exploration of event-based cameras in the literature highlights the potential of neuromorphic computing in robotics navigation, which is further discussed in Section VI as part of our future works.

C. Visual Place Recognition

The visual place recognition (VPR) task is to determine whether a place has been previously visited, even when faced with appearance changes and perceptual aliasing [13]–[15], [17], [18], [20]. VPR is most-commonly framed as an image retrieval problem, where feature representations of a given

query image are compared to the feature representations of all previously visited places, i.e. the reference database. The predicted place of the query image is the true position of the reference place that is most similar to the query in feature space. The VPR problem can also be posed as a template matching problem, similar to an image classification problem, where associated templates of all reference images are extracted to represent each place either via a single image [48], [49], [100], [101] or a sequence of images [41], [42], [102].

Learning-based approaches are dominating in VPR. Notably, NetVLAD [49], a method based on Vector of Locally Aggregated Descriptors (VLAD) [103], has been influential in producing robust feature representations. Recent advances partitioned training datasets into smaller segments, similar to our approach, and employed an ensemble of classifiers for each segment, facilitating large-scale VPR. For instance, the ‘Divide and Classify’ method [104] divides the reference dataset into non-overlapping classes. Each segment is processed by an individual classifier, and the collective ensemble is employed for predictions, enabling fast inference for large-scale VPR. On the other hand, Cosplace [52] reimagines the training process, viewing it as a classification task and sidestepping the resource-intensive process of mining positive and negative samples inherent in contrastive learning. Notably, both these works [52], [104] and ours share a common thread: framing VPR as a classification task to further scale place recognition capabilities.

We now review hierarchical approaches to demonstrate the broader landscape of methods aimed at improving recognition accuracy and robustness in VPR, providing a foundation for our exploration of ensembling techniques. Hierarchical techniques have been previously used for coarse-to-fine refinement frameworks via a monolithic neural network to efficiently predict hierarchical features (HF-Net) [105], probabilistic approaches [100], bio-inspired approaches [106], multi-process fusion [107], global-to-local VPR pipeline to guide local feature matching via global descriptors [108], and hierarchical decomposition of the environment [109]. In the latter approach, places with similar visual features are grouped together in nodes to reduce search space while maintaining high accuracy [109].

D. Non-spiking Biologically Inspired VPR Approaches

Biologically inspired VPR methods seek to emulate the navigational capabilities of animals with relatively small brains such as ants, bees, and rodents [86], [106], [110]–[113] to design energy-efficient and high-performing solutions. These non-spiking biologically inspired techniques offer valuable reference points for our spike-based work.

The place cells and head direction cells in rodent hippocampus inspired RatSLAM [86]. RatSLAM was later extended to include grid cells in [106] and extended to 3D in NeuroSLAM [111]. Inspired by the principles of Hierarchical Temporal Memory related to the human neocortex, [110] details a minicolumn network to pool spatial information and preserve temporal memory. [112] combines a pattern recognition module, inspired by fruit flies olfactory neural circuit, with a one-dimensional continuous attractor network serving as the temporal filter. Drawing on head direction cell

mechanisms, [113] details a calibration method to correct head direction errors from path integration via visual landmarks.

E. Ensembles of Neural Networks

One well-known approach to improve the predictive performance of neural networks is to use an ensemble of models [1], [29], [31]–[34]. Ensembles have been shown to generalize well and prevent issues such as over-fitting and instability, which makes these approaches suitable to a wide range of applications in different domains [1], [29], [31]–[34]. Challenges in deploying ensembles include requiring sufficient diversity in the output of the individual ensemble members, the trade-off between the computational complexity and the number of ensemble members, and the predictive performance and latency of the ensemble [1], [29], [31]–[34]. Although ensembles are typically limited in scalability, our work anticipates leveraging neuromorphic computing, renowned for its highly efficient parallelization capabilities [8], which alleviates scalability concerns in our ensembling approach due to the small size of each individual network.

Ensemble techniques have been used for a wide variety of robotic applications including image segmentation [114], gaze estimation [115], and uncertainty estimation [40]. In the field of SNNs, a variety of ensemble methods have been applied to diverse pattern recognition tasks [116]–[121]. These methods include unsupervised ensembles of spiking expectation maximization networks [116], and ensembles of evolutionary SNN algorithms [119] that have been effective in digit recognition. Additionally, there are heterogeneous ensembles for few-shot online learning [117], and SNN ensembles that utilize a convolutional structure with unsupervised STDP learning [118], and ensembles of Liquid State Machines [121] which have been applied to image [117], [118], [121] and audio recognition [121] tasks. Furthermore, reservoir computing ensembles have been explored for multi-object behavior recognition [120].

This work employs an ensemble of modular SNNs. The diversity within the ensemble members is created via variations in the initialization of the learned weights and the unique set of randomly selected distinct places learned by a module. The work most similar to ours is [122] which uses an ensemble of SNNs for digit recognition. While [122] allocates portions of an input image to different ensemble members for learning, our method processes full images across all modules, whereby each module is trained on a geographically distinct subset of reference places. All module responses are then fused for predicting the corresponding reference place of a given query image.

F. Sequence Matching Techniques for Neural Networks

One common approach to improve the robustness of a VPR method, especially against sudden high appearance changes and perceptual aliasing, involves utilizing the temporal information inherent in the database and query sets used in the mobile robot localization task [13]–[15], [17], [20]. One such set of algorithms that use the temporal information are sequence matching techniques, which can be classified into similarity-based methods, feature-based methods, and approaches that learn sequential information [13].

Similarity-based sequence matching techniques aggregate the similarity scores of a pair of sequences [13], which is advantageous as these methods can be used as a filtering process to single-image VPR methods. Similarity-based sequence matching techniques have been developed via local velocity search [41], convolutional operations [44], flow network built via a directed acyclic graph [123], and Hidden Markov Models [124]. However, similarity-based sequence matching techniques do not consider the underlying feature representation method and do not incorporate learning mechanisms [43]. Furthermore, elimination of single-image high-confidence false matches cannot be guaranteed without additional contextual information, and their scalability tends to increase linearly with the growth in the size of the reference dataset [42].

Conversely, feature-based techniques integrate a series of single-image descriptors into a unified descriptor to determine the predicted location of a query image, enabling the sequence descriptor to encompass visual information from both the current place and the preceding places in the sequence [13], [42], [125], [126]. Learning-based approaches generate a single summary sequential descriptor representing a sequence of images [43]. They exploit sequential temporal cues via methods including Transformers, and Convolutional-based architectures [45], or Long Short-Term Memory architectures [46].

In this study, we examine the impact of sequence matching across both conventional and SNN-based VPR techniques. We choose SeqSLAM [41] for analysis due to its simplicity and compatibility as a post-processing step for single-image VPR techniques. We provide analysis on the responsiveness of these techniques to sequence matching. We also introduce an indicator to predict the effectiveness of applying sequence matching, offering new insights into the efficacy of sequence matching across conventional and SNN-based VPR techniques.

III. METHODOLOGY

We first introduce the training regime for a single, compact spiking network that represents a small region of the environment (Section III-A). By combining the predictions of these localized networks at deployment time within a modular scheme (Section III-B) and introducing global regularization (Section III-C), we enable large-scale visual place recognition. We then present and analyze two enhancements to our modular networks: 1) ensembling, where a single place is represented by multiple networks (Section III-D) and 2) sequence matching, where we make use of multiple query and reference images for place matching (Section III-E).

A. Preliminaries

Our Modular SNN approach is homogeneous, i.e. each module within the Modular SNN has the same architecture and uses the same hyperparameters. The modules differ in their training data. Each module's training data consists of non-overlapping geographically distinct places of the environment. This specialization makes each module an expert in recognizing a small number of places. The SNN architecture introduced in this section, which constitutes a single SNN module in our modular approach, follows [59], [73] and is briefly introduced

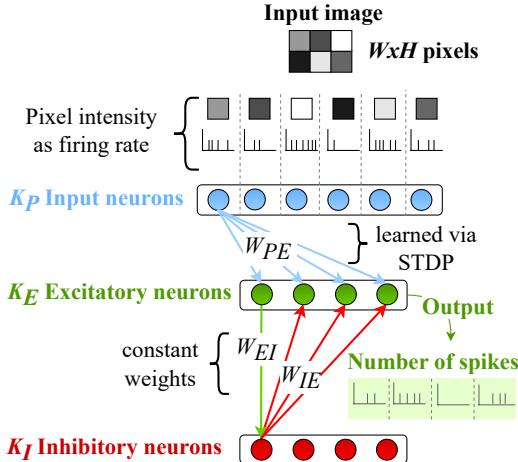


Fig. 2. SNN Module Architecture: Our Modular SNN is comprised by SNN modules that all have the same three-layer SNN architecture illustrated in this figure. Each module converts an input image to spike trains via rate coding, where the firing rate of input neurons is based on pixel intensities. The total number of input neurons is equal to the number of pixels in the input image, denoted as $K_P = W \times H$. These input neurons (blue dots) are fully connected to a layer of excitatory neurons (blue arrows connecting to green dots). Each excitatory neuron is connected to a single inhibitory neuron (single green arrow connecting to a red dot), which in turn connects to and inhibits all other excitatory neurons except its paired excitatory neuron (red arrows connecting back to green dots). The synaptic weights from excitatory to inhibitory neurons, W_{EI} , and from inhibitory back to excitatory neurons, W_{IE} , are fixed constants. The synaptic weights from input neurons to excitatory neurons, W_{PE} , are learned via the unsupervised Spike-Time Dependent Plasticity (STDP) mechanism that enables excitatory neurons to respond to different places. The number of output spikes from these excitatory neurons is used for place predictions.

for completeness in this section. We emphasize that we do not claim a novel SNN architecture.

Network structure: Each expert module, as illustrated in Figure 2, consists of a three-layer network architecture: 1) The input layer transforms each input image into Poisson-distributed spike trains via pixel-wise rate coding. The number of input neurons K_P corresponds to the number of pixels in the input image: $K_P = W \times H$, where W and H correspond to the width and height of the input image respectively. 2) The K_P input neurons are fully connected to K_E excitatory neurons. Each excitatory neuron learns to represent a particular place, and a high firing rate of an excitatory neuron indicates high similarity between the learned and presented place. Note that multiple excitatory neurons can learn the same place. 3) Each excitatory neuron connects to exactly one of the K_I inhibitory neurons. These inhibitory neurons inhibit all excitatory neurons except the excitatory neuron it receives a connection from. This enables lateral inhibition, resulting in a winner-takes-all system, that generates competition among excitatory neurons for effective learning.

Neuronal dynamics: The spiking neurons are modeled through their neuronal dynamics using the Leaky-Integrate-and-Fire (LIF) model [5]. The neuronal dynamics of excitatory neurons' internal voltage is as follows:

$$\frac{dV}{dt} = (E_{\text{rest},e} - V) + g_e(E_{\text{exc},e} - V) + g_i(E_{\text{inh},e} - V), \quad (1)$$

where τ_e is neuron time constant, $E_{\text{rest},e}$ is the resting membrane potential, or the internal voltage when the neuron

is not actively receiving spikes, and $E_{\text{exc},e}$ and $E_{\text{inh},e}$ are the equilibrium potentials of the excitatory and inhibitory synapses with synaptic conductance g_e and g_i respectively. The internal voltage of the inhibitory neurons are modeled using the same LIF neuronal dynamics, with neuron time constant τ_i , resting membrane potential $E_{\text{rest},i}$, and excitatory and inhibitory equilibrium potentials $E_{\text{exc},i}$ and $E_{\text{inh},i}$. The equilibrium membrane potentials determine the minimum and maximum internal voltage levels, in this case, depending on their negative or positive sign.

Network connections: The synaptic weights between input neurons and excitatory neurons, W_{PE} , are modeled via conductance changes to ensure stable network activity. When a synapse receives a presynaptic spike, the synaptic conductance is instantaneously increased by its synaptic weight and the synaptic conductance otherwise decays exponentially, as modeled by:

$$\tau_{g_e} \frac{dg_e}{dt} = -g_e, \quad (2)$$

where the time constant of the excitatory postsynaptic neuron is τ_{g_e} . The same model is used for inhibitory synaptic conductance g_i with the inhibitory postsynaptic potential time constant τ_{g_i} .

Weight updates: The synaptic weights between the inhibitory and excitatory neurons are defined with constant synaptic weights, W_{EI} and W_{IE} . The synaptic weights between input neurons and excitatory neurons, W_{PE} , are learned via the biologically inspired unsupervised learning mechanism Spike-Timing-Dependent-Plasticity (STDP). The weights are increased if the presynaptic spike occurs before a postsynaptic spike, and decreased otherwise. The synaptic weight change Δw_{pe} after an input neuron p receives a postsynaptic spike from an excitatory neuron e is defined by:

$$\Delta w_{pe} = \eta(x_{pe,\text{pre}} - x_{pe,\text{tar}})(w_{\max} - w_{pe})^\mu, \quad (3)$$

where η is the learning rate, $x_{pe,\text{pre}}$ records the number of presynaptic spikes, $x_{pe,\text{tar}}$ is the presynaptic trace target value when a postsynaptic spike arrives, w_{\max} is the maximum weight, and μ is a ratio for the dependence of the update on the previous weight.

Local regularization of excitatory neurons: To prevent individual excitatory neurons from dominating the response, homeostasis is implemented through an adaptive neuronal threshold. The voltage threshold of the excitatory neurons is increased by a constant Θ after the neuron fires a spike, otherwise the voltage threshold decreases exponentially. We note that the homeostasis provides regularization only on the *local*, expert-specific scale, not on the *global* modular-level scale.

Neuronal assignment: The network training encourages the network to discern the different patterns (i.e. places) that were presented during training. As the training is unsupervised, one needs to assign each of the K_E excitatory neurons to one of the L training places ($K_E \gg L$). Following [73], we record the number of spikes $S_{e,l}$ of the e -th excitatory neuron when presented with an image of the l -th place. The highest average response of the neurons to place labels across the local training

data is then used for the assignment A_e , such that neuron K_e is assigned to place l^* if:

$$A_e = l^* = \arg \max_l S_{e,l}. \quad (4)$$

Place matching decisions: Following [73], given a query image q , the matched place \hat{l} is the place l which is the label assigned to the group of neurons with the highest sum of spikes to the query image (i.e. $A_e = \hat{l}$):

$$\hat{l} = \arg \max_l \sum_{e[A_e=l]} S_{e,l}^q. \quad (5)$$

B. Modular Scheme

Modular network structure: The previous section described how to train individual spiking networks following [59], [73]. In the following sections, we present our novel modular spiking network, which consists of a set of $\mathcal{M} = \{M_1, \dots, M_i, \dots, M_N\}$ expert modules. The i -th module is tasked to learn the places contained in non-overlapping subsets $R_i \in \mathcal{R}$ of the reference database \mathcal{R} , whereby

$$\mathcal{R} = \bigcup_{i \in \{1, \dots, N\}} R_i \text{ with } R_i \cap R_j = \emptyset \quad \forall i \neq j. \quad (6)$$

All subsets are of equal size, i.e. $|R_i| = \kappa$. Therefore, at training time the modules are independent and do not interact with each other. This modular approach is aimed at improving the scalability of the individual spiking networks presented in Section III-A, allowing it to map to a larger number of places than the non-modular approach explored in [59].

Modular place matching decision: At deployment time, the query image q is provided as input to *all* modules *in parallel*. The predicted place of the Modular SNN, \hat{l}_M , is obtained by considering the spike outputs of *all* modules, rather than just a single module as in Eq. (5). We thus refine Eq. (5) to integrate the contributions of all N modules:

$$\hat{l}_M = \arg \max_l \sum_{i=1}^N \sum_{e[A_e=l]} S_{e,l}^{q,i}. \quad (7)$$

C. Hyperactive Neuron Detection

The basic fusion approach, that considers all spiking neurons of all modules as presented in Equation (7), is problematic. As the modules are only ever exposed to their local subset of the training data, there is a lack of global regularization to unseen training data outside of their local subset. In the case of spiking networks, this phenomenon leads to “hyperactive” neurons that are spuriously activated when stimulated with images from outside their training data. We decided to detect and remove these hyperactive neurons. This global regularization enhances place recognition capability within the modularity technique, allowing mapping to a larger number of places.

To detect hyperactive neurons, we do not require access to query data. We feed the entire reference dataset to each SNN module after training, and record the cumulative number of spikes $S_{e,l}^i$ fired by neurons K_e^i of each module $M_i \in \mathcal{M}$ in response to the entire reference dataset \mathcal{R} . $S_{e,l}^i$ indicates the

number of spikes fired by neuron K_e of module M_i in response to place l . Neuron K_e^i is considered hyperactive if

$$\sum_l S_{e,l}^i \geq \theta, \quad (8)$$

where θ is a threshold value that is determined as described in Section IV-A. The place match is then obtained by the highest response of neurons that are assigned to place \hat{l}_M after ignoring all hyperactive neurons:

$$\overline{\hat{l}_M} = \arg \max_l \sum_{i=1}^N \sum_{e[A_e=l]} S_{e,l}^{q,i} \mathbb{1}_{\sum_l S_{e,l}^i < \theta}, \quad (9)$$

where the indicator function $\mathbb{1}$ filters all hyperactive neurons.

D. Ensemble of Modular SNNs

Ensemble network structure: In this section, we introduce ensembles of Modular SNNs. The purpose of these ensembles is to improve robustness and generalization ability. The main idea is that each place l is represented by multiple complementary ensemble members. Specifically, the ensemble is represented as a set of $\mathcal{E} = \{E_1, \dots, E_m, \dots, E_M\}$ homogeneous ensemble members (i.e. their network architecture is the same), where each member is an independent Modular SNN. The ensemble members are all trained in parallel on the entire reference database \mathcal{R} (Figure 1).

We generate diversity among the ensemble members through random initialization of learned weights, and random shuffling of the order of input images. This approach aligns with prior work that demonstrated substantial performance improvements in such settings [40].

Ensemble place matching decision: At deployment time, a query image q is provided as input to all ensemble members in parallel. The predicted place of the Ensemble of Modular SNNs, $\overline{\hat{l}_E}$, is determined as the place l which corresponds to the place label that has been assigned to a group of neurons (among all ensemble members) demonstrating the highest cumulative spike activity in response to the input query image (i.e. $A_e = \overline{\hat{l}_E}$). Eq. (9) is revised as follows to accommodate for the M ensemble members and their corresponding N modules:

$$\overline{\hat{l}_E} = \arg \max_l \sum_{m=1}^M \sum_{i=1}^N \sum_{e[A_e=l]} S_{e,l}^{q,i,m} \mathbb{1}_{\sum_l S_{e,l}^i < \theta}. \quad (10)$$

Creation of distance matrix: To compute the single-frame distance matrix for the Ensemble of Modular SNNs, D_{single} , we compute the cumulative spike activity for all neurons assigned to each place label for each query image q . This contrasts with taking only the maximum that is considered as the prediction in Eq. 10. Specifically, given a query image q , the q -th column of the single-frame similarity matrix S_{single} is defined as:

$$S_{\text{single}}^{(q)} = \sum_{m=1}^M \sum_{i=1}^N \sum_{e[A_e=l]} S_{e,l}^{q,i,m} \mathbb{1}_{\sum_l S_{e,l}^i < \theta}. \quad (11)$$

We then convert this similarity matrix, S_{single} , to a distance matrix, D_{single} , by subtracting the maximum value of the similarity matrix from each element:

$$D_{\text{single}}(u, v) = \max(S_{\text{single}}) - S_{\text{single}}(u, v) \quad (12)$$

The same process is applied to our Modular SNN, where all responses from Eq. 9 (rather than just the maximum response used for the prediction) for a given query image q constitute one column of the distance matrix.

E. Sequence Matching

This section briefly introduces the sequence matching technique, SeqSLAM [41], and its convolutional formulation as introduced in SeqMatchNet [43], which we do not claim as our contribution. In Section V-C, we will analyze the receptiveness of our Ensemble of Modular SNNs and contrast it to that of conventional techniques.

Given the single-frame distance matrix D_{single} from Eq. 12, we apply the sequence matching operation to the distance value between the reference image at row v and query image at column u to obtain the value of the sequential distance matrix D_{seq} at the same corresponding location:

$$D_{\text{seq}}(u, v) = \sum_{x \in \{1, \dots, L_{\text{seq}}\}} \sum_{y \in \{1, \dots, L_{\text{seq}}\}} D_{\text{single}}(u + x, v + y) I_{L_{\text{seq}}}(x, y), \quad (13)$$

where $I_{L_{\text{seq}}}$ is an identity matrix acting as a square filter kernel with dimensions of $L_{\text{seq}} \times L_{\text{seq}}$ pixels.

This sequence matching operation assumes that the reference images and query images are aligned, as convolving the single-frame distance matrix with an identity matrix is a linear temporal alignment process. To account for misalignment between the reference and query images, contextual information about the varying speeds between the reference traversal and the query traversal can be used, via Dynamic Time Warping [127] or linear search as done in SeqSLAM [41].

IV. EXPERIMENTAL SETUP

In this section, we cover our implementation details in Section IV-A, the datasets that we used for evaluation in Section IV-B, the proof-of-concept robot deployment details in Section IV-C, and the evaluation metric in Section IV-D. Furthermore, in Section IV-E, we provide the VPR techniques used for comparison, along with details on the image dimensions in Section IV-F.

A. Implementation Details

We implement our SNNs using the Brian2 simulator. We publicly release our code at: <https://github.com/QVPR/VPRSNN>. Each SNN module contains $K_E = 400$ excitatory neurons, $K_I = 400$ inhibitory neurons, and $K_P = W \times H$ input neurons, where the input image height and width is 28 pixels in each dimension. We select the network's hyperparameters as follows: For the number of training epochs, we use a fixed value of 30 epochs for all datasets. For the threshold value θ (Eq. (8)) to detect the hyperactive neurons for each individual Modular SNN evaluated on a dataset, we draw the threshold value from a uniformly random distribution with $U(40, 100)$ for

small-scale datasets (< 1000 places) and $U(600, 800)$ for large-scale datasets (≥ 1000 places). The threshold value of each Modular SNN within an Ensemble of Modular SNNs evaluated on a dataset, is chosen separately based on this uniformly random selection process. For our Ensemble of Modular SNNs in the results section, Section V, we used three or five ensemble members, with the exact number specified in each part.

Following [73], we select biologically plausible ranges for all parameters of a single SNN module, except the time constant of the synaptic conductance of the inhibitory neurons, τ_{g_i} . We modified τ_{g_i} from 2 ms, as used by [73], to 0.5 ms. The parameter values for a single SNN module are as follows: For the LIF neuronal dynamics, the voltage threshold of the excitatory neurons is increased by a constant $\Theta = 0.05$ mV after a neuron spikes. The equilibrium potentials are $E_{\text{exc},e} = E_{\text{exc},i} = 0$ mV for both excitatory and inhibitory neurons, $E_{\text{inh},e} = -100$ mV for excitatory neurons and $E_{\text{inh},i} = -85$ mV for inhibitory neurons. For excitatory neurons, the τ_e time constant is 100 ms, and the resting membrane potential, $E_{\text{rest},e}$, is -65 mV. For inhibitory neurons, the τ_i time constant is 10 ms, and the resting membrane potential, $E_{\text{rest},i}$, is -60 mV. The time constant of the synaptic conductance for the excitatory neurons is $\tau_{g_e} = 1$ ms. For STDP learning, the ratio for the dependence of update on the previous weight is $\mu = 1$, and the maximum weight is $w_{\text{max}} = 1$. The learning rate η is $1e-4$ if the presynaptic input neuron fires before the postsynaptic excitatory neuron, and $1e-2$ if the reverse occurs. The constant synaptic weights between inhibitory and excitatory neurons are $W_{EI} = 10.4$ and $W_{IE} = 17.0$.

We train each SNN module in our Modular SNN with $R_i = 25$ images, and the total number of SNN modules depends on the size of the reference dataset. For instance, given a dataset with $\mathcal{R} = 1000$ reference places, we assign each of our SNN modules to learn $R_i = 25$ distinct places, resulting in a total of 40 SNN modules. For each dataset, we train our SNN-based approach only on the reference set without any pre-training, and use the corresponding query set for testing. We use QUT's High Performance Computing (HPC) to run each SNN module in a separate CPU job. Most jobs were run on an Intel Xeon Gold 6140 CPU.

B. Datasets

We evaluate our work using several datasets that cover seasonal changes [54], [57], differences in the time of day [55]–[57], as well as rural [54], [56], suburban [58] and urban [55], [57] environments with additional challenges due to occlusions and glare [58]. We now briefly describe these datasets, and provide sample images in Figure 3. Our code base² provides details on how these datasets are used.

The Nordland dataset [54]: captures a 728 km train path in Norway recorded in spring, summer, fall and winter. As commonly done in the literature [128]–[130], the data segments where the speed of the train is below 15km/h is removed using the provided GPS data. We used the Nordland dataset configured into two different sets: 1) Reference: Fall; query: Summer (also referred to as Nordland FS); 2) Reference:

²<https://github.com/QVPR/VPRSNN>

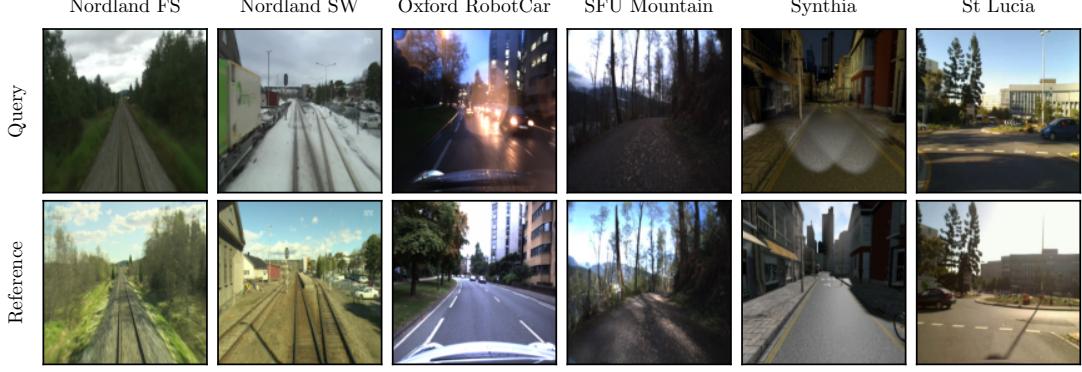


Fig. 3. **Sample images from the six VPR datasets employed in our research:** These datasets encompass a diverse range of environments including urban locales undergoing seasonal transitions, varying illuminations from day to night, high-glare-induced illumination shifts, scenes with occlusions, railway lines, and forested areas.

Spring; query: Winter (also referred to as Nordland SW). We considered every image in a traverse as a distinct place, obtaining 27575 places for each traverse.

The Oxford RobotCar dataset [55]: has over 100 traverses captured in Oxford city and it is recorded at varying conditions including different time of the day, and different seasons. As done in prior works [59], [131], we selected the front left stereo frames from the Rain (2015-10-29-12-18-17) traverse as the reference dataset and from the Dusk (2014-11-21-16-07-03) traverse as the query dataset. For each traverse, we considered every image as a distinct place, gaining 3800 places.

The SFU-Mountain dataset [56]: has more than 8 hours of trail driving in Burnaby Mountain, British Columbia Canada, using the Clearpath Husky robot covering sunny, rainy and snowy conditions [56]. Following prior works [132], [133], we used the front right stereo frames from the dry traverse for reference and from the dusk traverse for query. We considered each image in the traverse as a different place, and used the author's place sampling configurations [56], utilizing the entire dataset where each traverse contains 375 places.

The Synthia Night-to-Fall dataset [57]: is a synthetic dataset that was initially designed for semantic scene understanding in a city-like driving scenario. In our approach, following the methodology described in [134], we utilized the SEQS-04 foggy (reference) and nighttime (query) traverses. Segments in which the vehicle remained stationary were excluded, and we sampled approximately every 1 frame per second resulting in 250 places for each traverse.

The St Lucia dataset [58]: comprises several traverses along a route within the St Lucia suburb of Brisbane. For our experiments, we employed a traverse conducted during the early morning (190809-0845) as the reference traverse and another in the late afternoon (180809-1545) as the query traverse. We omitted the segments where the vehicle was at rest and sampled places approximately every 15 meters. We used only the unique places from the reference traverse, obtaining 500 places. For the query traverse, we included sections where places were visited multiple times within the same traverse, obtaining 1037 places.

C. Proof-of-concept Robot Deployment Setup

For our proof-of-concept CPU-based robot deployment experiment of our Modular SNN, we used AgileX's Scout Mini robot [60] to navigate the QUT Centre for Robotics floor. This area is a shared space with other researchers, where there are moving people, and robots, as well as relocated objects. The robot was tele-operated at a speed of approximately 1 m/s. It collected reference and query traverses at a frequency of 1 Hz at different times of the day, encountering challenges such as occlusions, brightness changes, and slight lateral and frontal viewpoint changes. In this experiment, we used a 32GB Intel Core i7 CPU for processing, and an image-based camera (Intel's RealSense D435) for capturing images of the environment.

D. Evaluation Metrics

We evaluate the performance of our SNN-based approaches and baseline methods on all datasets using the recall at N ($R@N$) evaluation metric. This performance metric considers a prediction as a correct match if at least one of the top N predictions is correct [13], [134]. We deem a query image as correctly paired only if it aligns *exactly* with the correct reference place, employing a ground truth tolerance of zero. In the case of sequence matching, the query sequence has to match exactly to the reference sequence.

Let P_k be the set of top N predictions for the k^{th} query and G_k be the set of ground truth matches for the k^{th} query with zero ground truth tolerance. Let \mathcal{Q} be the total number of queries. The recall at N ($R@N$) can be defined as:

$$R@N = \frac{1}{Q} \sum_{k=1}^Q \mathbb{1}(P_k \cap G_k \neq \emptyset), \quad (14)$$

where $\mathbb{1}(\cdot)$ is the indicator function, which is 1 if at least one of the top N predictions for the k^{th} query, P_k , is correctly matched to its ground truth G_k , and 0 otherwise.

E. Baseline Methods

We employ several conventional VPR approaches to evaluate the performance of our methods. These techniques encompass a non-learned method: Sum-of-Absolute-Differences (SAD) [41], as well as learning-based methods:

DenseVLAD [48], NetVLAD [49], AP-GeM [50], Generalized Contrastive Loss (GCL) [51], CosPlace [52], and MixVPR [53]. We also include a comparison to a previous Non-modular SNN [59] for VPR. This comparison is included to show the scalability advantages of ensembling and modularity in terms of both performance and computational time as the number of places to learn increases. These approaches are detailed as follows:

Sum-of-Absolute-Differences (SAD) [41]: is a simple baseline technique which computes the pixel-wise difference between each query image and all reference images. While the sum-of-absolute differences method is simple, this method is not very robust to drastic varying conditions such as changes in lighting, viewpoint, and seasonal changes.

NetVLAD [49]: aggregates local image descriptors with learnable aggregation weights by employing the VLAD technique [103] to create a fixed-size global descriptor. This model has a VGG16 [135] backbone pretrained on ImageNet [136]. The NetVLAD layer, which is the pooling layer of the network, is trained on the Google Landmarks [137], Mapillary Street Level Sequences [138], and Pittsburgh [139] datasets separately. As NetVLAD is trained on urban environments, the model might not generalize well to completely different types of environments such as rural areas.

DenseVLAD [48]: utilizes densely sampled SIFT [140] image descriptors, and aggregates these features using VLAD [103]. While DenseVLAD is very robust to high illumination, some limitations of DenseVLAD include lack of robustness to occlusions, very dark conditions with limited dynamic range, and rural areas with vegetation.

AP-GeM [50]: employs the Generalized-Mean pooling layer (GeM) [141] and uses a listwise loss formulation that directly optimizes for the Average Precision (AP) performance metric. This model uses a CNN-based backbone pretrained on ImageNet [136] to extract feature representations and aggregates it into a compact representation. We used three variations of AP-GeM; a ResNet50 backbone [142] trained on Landmarks-clean [143] dataset, a ResNet101 backbone [142] trained on Landmarks-clean [143] dataset, and a Res101 backbone [142] trained on Google-Landmarks [137] Dataset. Similar to NetVLAD, AP-GeM generalization is reliant on the type of environment used for training.

GCL [51]: is trained via a Generalized Contrastive Loss (GCL) using graded similarity labels for image pairs. We trained the last two layers of the network on the Mapillary Street-Level Sequences (MSLS) dataset [138] using a VGG16 backbone [135] pretrained on ImageNet [136] with GeM [141] as the pooling layer.

CosPlace [52]: uses a classification framework to train the model. This approach splits the training dataset into square geographical cells using the ground truth data. During training, the network iterates over CosPlace Groups, which are non-overlapping classes that are grouped together. The network uses the Large Margin Cosine Loss (LCML) [144] with a fully connected layer, that is only present at training, for each group of the dataset. The network consists of a CNN backbone (a VGG-16 backbone), a GeM pooling layer, and a fully connected output layer. At inference, the model outputs

compact discriminative feature descriptors.

MixVPR [53]: is a global feature aggregation method, that takes the feature maps of intermediate layers of a CNN-based backbone (a ResNet backbone), that are processed by a cascade of Feature Mixer layers, comprised of multi-layer perceptrons, that provide each element of the feature map with global relationships to all other elements. MixVPR is a SOTA VPR technique across multiple VPR benchmark datasets.

Non-modular SNN [59]: is a single three-layer SNN model that is enlarged to accommodate for learning a significantly larger number of places. The model is trained on the reference dataset, and tested on the query dataset.

F. Image Dimensions for Different Techniques

For our SNN-based methods, we performed the following pre-processing steps: resized each input image to 28×28 pixels, converted the images to grayscale, applied patch normalization by dividing the images into 7×7 pixel patches, and normalizing each patch to a range of -1 to 1 using its mean and standard deviation. Finally, we scaled the pixel values to be between 0 and 255 . For GCL and NetVLAD, the images were resized to 640×480 pixels, while for AP-GeM, DenseVLAD, CosPlace, and MixVPR the native image resolutions were used (Nordland: 640×360 pixels, Oxford RobotCar: 1280×960 pixels, SFU-Mountain: 752×480 pixels, Synthia Night To Fall: 300×200 pixels, St Lucia: 640×480 pixels). For SAD, the input images were resized to 28×28 pixels and patch-normalized with patch sizes of 7×7 pixels, matching the low-dimensional input image sizes that we used in our work.

V. RESULTS

Section V-A analyzes how each component of our methodology affects the overall performance of the system. Subsequently, Section V-B compares the performance of our Ensemble of Modular SNNs *without* sequence matching against conventional VPR techniques. Section V-C extends this comparison to include sequence matching. Section V-D provides detailed analyses and introduces an indicator to assess the responsiveness of VPR techniques to sequence matching. We also provide an evaluation of the ensembling effect on our Modular SNN in Section V-E. This is followed by an ablation study on the effect of ensemble member randomization in our Ensemble of Modular SNNs in Section V-F. Additionally, Section V-G provides an analysis of the computational efficiency and scalability aspects of our approach. Finally, Section V-H concludes this section with a proof-of-concept robot deployment in a small indoor environment.

A. Component-Wise Contributions to SNN Performance

This section overviews the performance contributions of each component of our approach, namely modularity (i.e. expert modules that learn small subsets of the reference dataset), ensembling (i.e. representing each place by multiple modules), and sequence matching (i.e. using multiple reference and query images for place matching). Figure 4 demonstrates that the performance of our Modular SNN is significantly increased

TABLE I

R@1 PERFORMANCE COMPARISONS OF OUR MODULAR SNN AND ENSEMBLES OF MODULAR SNNs TO CONVENTIONAL VPR TECHNIQUES WITH A SEQUENCE MATCHER AT DIFFERENT SEQUENCE LENGTHS (SL): SL1 (WITHOUT A SEQUENCE MATCHER), SL2, SL4, AND SL10. THE MAIN KEY TAKEAWAYS INCLUDE: OUR ENSEMBLE OF MODULAR SNNs 1) SHOWS COMPETITIVE PERFORMANCE, COMPARABLE TO MULTIPLE VPR METHODS ACROSS VARIOUS DATASETS, 2) OBTAINS THE HIGHEST R@1 IMPROVEMENT WITH A SEQUENCE MATCHER, COMPARED TO VPR TECHNIQUES WITH SIMILAR-PERFORMING BASELINES EXCEPT ON OXFORD ROBOTCAR, AND 3) CONSISTENTLY OUTPERFORMS THE MEAN R@1 OF ITS INDIVIDUAL MEMBERS ACROSS ALL DATASETS.

Datasets	Nordland FS				Nordland SW				Oxford RobotCar RD				SFU Mountain				Synthia Night to Fall				St Lucia				Mean R@1			
Method	SL1	SL2	SL4	SL10	SL1	SL2	SL4	SL10	SL1	SL2	SL4	SL10	SL1	SL2	SL4	SL10	SL1	SL2	SL4	SL10	SL1	SL2	SL4	SL10	SL1	SL2	SL4	SL10
NetVLAD (Landmarks)	0.57	0.65	0.73	0.80	0.12	0.16	0.21	0.30	0.08	0.10	0.14	0.22	0.40	0.48	0.62	0.87	0.79	0.84	0.87	0.87	0.26	0.45	0.60	0.79	0.37	0.45	0.53	0.64
NetVLAD (Mapillary)	0.50	0.60	0.70	0.78	0.10	0.14	0.20	0.31	0.08	0.10	0.14	0.24	0.32	0.42	0.53	0.76	0.67	0.76	0.87	0.88	0.30	0.42	0.59	0.79	0.33	0.41	0.51	0.63
NetVLAD (Pittsburgh)	0.43	0.54	0.65	0.76	0.08	0.11	0.14	0.21	0.12	0.15	0.21	0.33	0.39	0.56	0.71	0.90	0.81	0.86	0.89	0.87	0.26	0.43	0.57	0.78	0.35	0.44	0.53	0.64
AP-GeM (ResNet-50)	0.47	0.56	0.66	0.75	0.07	0.10	0.13	0.19	0.11	0.13	0.18	0.29	0.40	0.53	0.75	0.92	0.59	0.76	0.83	0.87	0.35	0.57	0.78	0.95	0.33	0.44	0.55	0.66
AP-GeM (ResNet-101)	0.48	0.58	0.67	0.76	0.06	0.08	0.09	0.13	0.12	0.15	0.19	0.31	0.43	0.58	0.75	0.95	0.62	0.75	0.84	0.85	0.34	0.54	0.72	0.90	0.34	0.45	0.54	0.65
AP-GeM (ResNet-101, LM18)	0.67	0.74	0.78	0.82	0.20	0.26	0.34	0.47	0.12	0.15	0.20	0.29	0.61	0.76	0.88	0.96	0.73	0.84	0.87	0.87	0.55	0.75	0.91	0.98	0.48	0.58	0.67	0.73
DenseVLAD	0.65	0.73	0.77	0.81	0.16	0.26	0.37	0.54	0.19	0.24	0.32	0.47	0.67	0.78	0.86	0.96	0.77	0.80	0.87	0.87	0.61	0.80	0.92	0.99	0.51	0.60	0.68	0.77
SAD	0.52	0.61	0.69	0.78	0.15	0.22	0.30	0.44	0.19	0.25	0.29	0.37	0.41	0.51	0.63	0.80	0.48	0.61	0.68	0.78	0.24	0.32	0.41	0.40	0.33	0.42	0.50	0.60
GCL	0.51	0.62	0.71	0.78	0.14	0.21	0.30	0.50	0.05	0.06	0.07	0.12	0.45	0.57	0.69	0.86	0.52	0.68	0.83	0.88	0.29	0.47	0.67	0.84	0.33	0.43	0.55	0.66
MixVPR	0.78	0.81	0.82	0.83	0.73	0.84	0.91	0.96	0.23	0.28	0.36	0.51	0.86	0.94	0.99	1.00	0.84	0.87	0.89	0.88	0.99	1.00	1.00	1.00	0.65	0.72	0.78	0.81
CosPlace	0.73	0.77	0.79	0.82	0.60	0.72	0.84	0.93	0.12	0.14	0.18	0.25	0.74	0.87	0.97	1.00	0.82	0.85	0.87	0.88	0.90	0.97	1.00	1.00	0.74	0.79	0.83	0.86
Modular SNN (Ours)	0.42	0.55	0.67	0.78	0.12	0.19	0.29	0.47	0.15	0.21	0.28	0.40	0.36	0.56	0.78	0.95	0.35	0.49	0.64	0.82	0.24	0.38	0.57	0.84	0.27	0.40	0.54	0.71
Ens of 3 AP-GeM	0.64	0.71	0.77	0.81	0.14	0.18	0.23	0.31	0.16	0.18	0.24	0.35	0.61	0.72	0.86	0.97	0.73	0.81	0.87	0.87	0.55	0.69	0.87	0.97	0.47	0.55	0.64	0.71
Ens of 3 GCL	0.52	0.62	0.71	0.78	0.15	0.23	0.33	0.54	0.05	0.05	0.07	0.11	0.46	0.57	0.70	0.86	0.54	0.68	0.82	0.88	0.29	0.47	0.68	0.83	0.33	0.44	0.55	0.66
Ens of 3 Modular SNNs (Ours)	0.50	0.62	0.72	0.80	0.17	0.25	0.35	0.54	0.20	0.25	0.33	0.44	0.43	0.64	0.83	0.98	0.39	0.57	0.72	0.85	0.27	0.44	0.65	0.89	0.33	0.46	0.60	0.75
Ens of 5 GCL	0.52	0.62	0.70	0.78	0.15	0.24	0.35	0.55	0.05	0.05	0.07	0.11	0.47	0.56	0.70	0.86	0.54	0.67	0.82	0.88	0.30	0.47	0.67	0.83	0.34	0.44	0.55	0.67
Ens of 5 Modular SNNs (Ours)	0.52	0.63	0.72	0.80	0.18	0.26	0.37	0.55	0.21	0.27	0.35	0.44	0.46	0.67	0.85	0.98	0.44	0.59	0.72	0.85	0.32	0.49	0.69	0.92	0.35	0.49	0.62	0.76

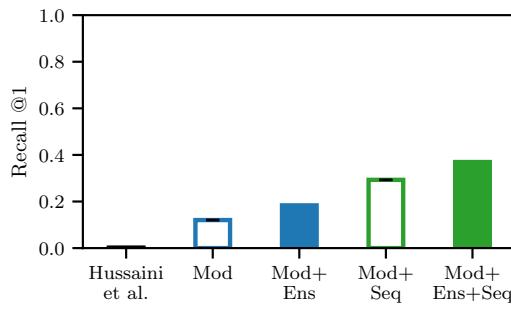


Fig. 4. **Component-wise ablation study:** Introducing modularity (Mod) where multiple SNNs represent small subsets of the reference dataset enables large-scale place recognition, significantly outperforming the Non-modular baseline SNN by Hussaini et al. [59]. Both ensembling (five ensemble members; Mod+Ens) and sequence matching (sequence length four; Mod+Seq) individually enhance the R@1 of the Modular SNN, by 6.3% and 17.2% respectively. Their combined application (Mod+Ens+Seq) further elevates the performance, surpassing the benefits of the individual techniques and resulting in a 24.9% R@1 improvement overall. Error bars indicate performance variations among the five ensemble members (standard deviation). The experiment was conducted on the Nordland dataset (Reference: Spring, Fall; query: Winter).

when each of these techniques, ensembling (R@1 increase of 6.3% with five ensemble members) and sequence matching (R@1 increase of 17.2% for a sequence length of four), is applied separately. Moreover, the combination of ensembling and sequence matching techniques further improves the R@1 of the Modular SNN (R@1 increase of 24.9%). As the ensembling and sequence matching techniques are commutative, the order of application of these two techniques produces identical outcomes. The Non-modular SNN by Hussaini et al. [59] performs poorly, even when ensembling and sequence matching are applied (R@1 is less than 0.2%).

B. Comparison of Ensemble of Modular SNNs without a Sequence Matcher to Conventional VPR Techniques

This section compares our Ensemble of Modular SNNs without a sequence matcher to the conventional VPR techniques outlined in Section IV-E on the datasets detailed

in Section IV-B. As emphasized in [145], the efficacy of different visual place recognition methods fluctuates across different environments. The aim of our work is to demonstrate competitive but not necessarily state-of-the-art performance of our approach for visual place recognition.

Table I shows that our Ensemble of Modular SNNs consistently delivers competitive results, that are in close proximity to some of the leading VPR methods across various datasets. On average across all datasets, the top-performing methods include CosPlace, MixVPR, DenseVLAD, and AP-GeM (ResNet-101, LM18). It is worth noting that conventional VPR techniques have inherent advantages. They operate on much larger image dimensions as they falter with smaller image sizes which are used in our approach, and oftentimes benefit from extensive pretraining on large VPR datasets.

Figure 10 presents qualitative performance of our Modular SNN, Ensemble of Modular SNNs, and relevant VPR techniques used for comparison across all datasets, for both correct and incorrect prediction of query image instances.

C. Comparison of Ensemble of Modular SNNs with a Sequence Matcher to Conventional VPR Techniques

This section extends the comparisons, this time incorporating a sequence matcher. Table I shows the performance of our Ensemble of Modular SNNs as well as our standalone Modular SNN when integrated with a sequence matcher with a sequence length of two, four, and ten, *separately*, compared to conventional VPR techniques and/or their ensemble forms with a sequence matcher of same sequence lengths across all datasets.

Compared to VPR techniques with roughly similar baseline performance, our Ensemble of Modular SNNs (with five ensemble members) obtains the overall highest improvement with a sequence matcher averaged across all datasets: the mean R@1 of our Ensemble of Modular SNNs with a sequence matcher across all datasets is among the top five models, despite the lower baseline performance without a sequence matcher. Specifically, our Ensemble of Modular SNNs with a four-frame sequence matcher achieved a 39.1% R@1 performance gain

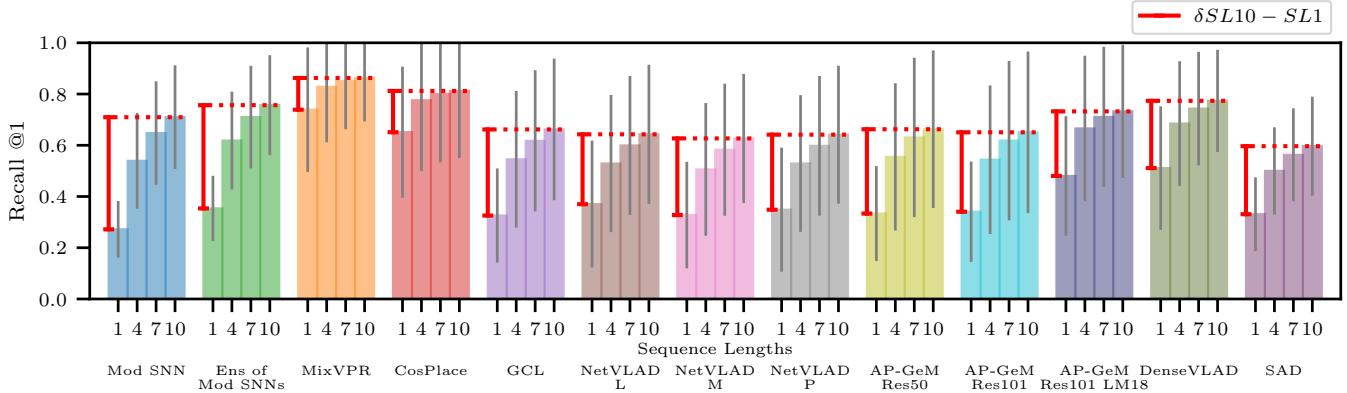


Fig. 5. R@1 performance improvements with sequence matching: The plot shows the mean R@1 performance of each method across all datasets when employing sequence matching using four, seven and ten frames, compared to the single-frame approach (SL1). The gray lines represent the standard deviation of the R@1 model performance across all datasets. The red bar demonstrates the mean R@1 performance improvement of a method without a sequence matcher to the performance of the method with a sequence matcher of sequence length ten. Our Modular and Ensemble of Modular SNNs obtain the highest R@1 improvement with a sequence matcher (from without a sequence matcher to a sequence matcher of sequence length ten). The mean R@1 performance of both our Ensemble of Modular SNNs (with five ensemble members), and Modular SNN without a sequence matcher (SL1) is competitive with multiple VPR techniques, and incorporating a sequence matcher with sequence lengths of four, seven, and ten enables our SNN-based approaches to obtain the highest R@1 improvement compared to similar-performing VPR baselines. Notably, the R@1 performance of our SNN-based approaches with a sequence matcher of sequence length ten frames slightly surpasses that of AP-GeM (ResNet101, LM18), and is in close approximately to that of DenseVLAD, both of which have higher-performing baselines (without a sequence matcher).

on the SFU Mountain dataset with a baseline performance of 46.1%, surpassing the 22.9% increase of the comparable VPR technique, GCL, with a similar baseline performance of 47.2%.

On the Nordland SW dataset, before applying the sequence matcher, the R@1 of our model was 18.3%, comparable to the baseline performance of AP-GeM (ResNet101, LM18), which stood at 19.6%. Upon integrating the sequence matcher with four frames, the performance of our model increased to 36.9%, an increase that slightly surpasses the post-sequence matching performance gains observed in AP-GeM (ResNet101, LM18), which reported an increase to 34.1%.

Figure 5 illustrates the R@1 of all techniques with a sequence matcher of sequence lengths four, seven, and ten averaged across all datasets. With the integration of a sequence matcher of sequence length ten, our Ensemble of Modular SNNs achieves higher R@1 improvements over all other VPR methods. We note that methods such as MixVPR and CosPlace, which have an already significantly higher baselines, compared to all other evaluated techniques, do not significantly benefit from being paired with a sequence matcher.

It is noteworthy that at longer sequence lengths, the R@1 performance of all techniques converge towards perfect R@1 values, which makes it challenging to distinguish between the performances of these techniques with a sequence matcher. The variations in absolute performance gain of different techniques are more pronounced in shorter sequence lengths, offering clearer insights into the adaptability. Sequence matchers with shorter lengths are apt for indoor settings or high-speed contexts where successive visual scenes change swiftly, and reducing relocalization delays.

D. An Indicator for Sequence Matching Responsiveness

This section examines whether the responsiveness of VPR techniques to sequence matching can be predicted, and offers insights that corroborate their respective behaviors. Specifically,

we investigate how the sparsity of correct matches influences the R@1 performance when sequence matching is employed. Increased sparsity in correct matches denotes larger gaps between next correct matches across all predictions of query images due to limitations in the performance of a method.

We define the correct match sparsity as the mean distance to the next correct match for all predictions of query images, projected to log space and min-max normalized per dataset, which enables dataset-agnostic evaluation. Let D be the distance matrix of size $L_R \times L_Q$, where L_R is the number of reference places, and L_Q is the number of query places. Further, let GT be a binary matrix of the same size, where a 1 indicates true matches, and 0 indicates false matches. The indices of correct matches are identified as follows:

$$\{q \mid q = \arg \min_i D(r, i) \wedge GT(r, i) = 1\}. \quad (15)$$

The difference sequence Δq can then be defined as:

$$\Delta q = \{\Delta q_1, \Delta q_2, \Delta q_3, \dots, \Delta q_{n-1}\}, \quad (16)$$

where $\Delta q_i = q_{i+1} - q_i$ for $i = \{1, 2, \dots, n-1\}$. The mean distance \tilde{d} to the next correct match for all correct matches in the distance matrix is calculated as:

$$\tilde{d} = \frac{1}{|\Delta q|} \sum_i \Delta q_i. \quad (17)$$

Figure 6 presents the relationship between the correct match sparsity \tilde{d} and the R@1 performance ratio for a sequence matcher with a length of four against a sequence matcher with a length of one. For better visualization, this figure shows $\log(\tilde{d})$ with a min-max-normalization across different datasets, such that the sparsity ranges between 0 and 1.

Central to our discussion, the data points representing our Modular SNN evaluated on different datasets, distinguished by their larger blue points, show a higher SL4 to SL1 ratio compared to data points with a similar correct match sparsity,

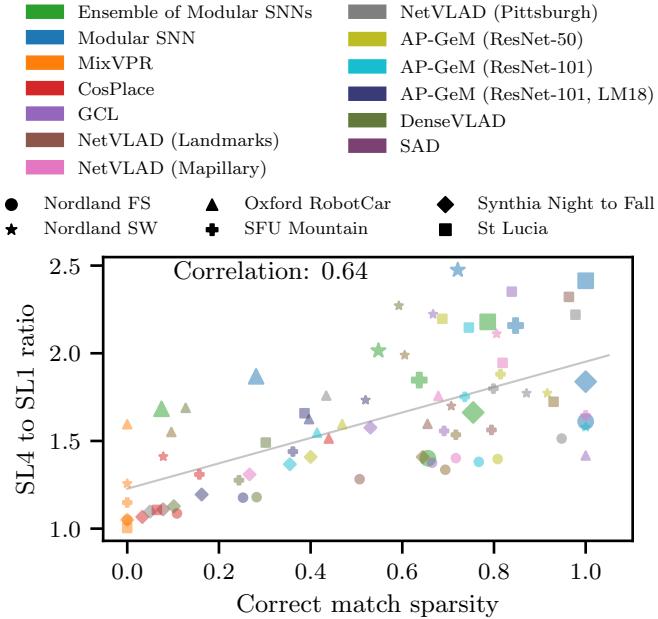


Fig. 6. Indication of sequence matching responsiveness: The figure shows the correct match sparsity against the R@1 performance ratio of methods with a sequence matcher of sequence lengths four to one. Correct match sparsity is defined as the mean distance to the next correct match for all predictions of query images, projected to log space and min-max normalized for each dataset to allow for dataset agnostic evaluation. Higher sparsity indicates wider gaps between correct matches, suggesting potential under-performance. The gray line represents the best fit line for all data points. Key observations are: 1) Our Modular SNN and Ensemble of Modular SNNs (with five ensemble members) generally show a higher SL4 to SL1 ratio compared to nearby data points, regardless of the dataset the methods were evaluated on, indicating strong adaptability to sequence matching. Exceptions include our Ensemble's performance on Nordland FS, Synthia Night to Fall, and our Modular SNN evaluated on Nordland FS, which are comparable to other methods. 2) Across all data points, methods with high correct match sparsity show greater responsiveness to sequence matching, regardless of the method observed.

emphasizing the robustness and adaptability of our Standalone Modular SNN to sequence matching. An exception is the evaluation on Nordland FS which showed similar performance to nearby data points. Similarly, our Ensemble of Modular SNNs, represented by their larger green points, has a high responsiveness to sequence matching, with most data points being higher than methods with a similar correct match sparsity. However, on Nordland FS and Synthia Night to Fall, our Ensemble of Modular SNNs has similar performance to nearby methods.

As correct match sparsity correlates to method performance, it means that methods with lower baseline performance generally benefit more from sequence matching compared to methods with higher baseline performance. Methods with higher-performing baselines include MixVPR and CosPlace, whose data points mainly occupy the lower left of the plot. Due to min-max normalization across datasets, low correct match sparsity indicates the highest-performing methods, while high sparsity suggests lower performance within each dataset.

E. Ensembling: How Much Does It Help?

This section evaluates the effect of ensembling on our Modular SNN, and provides comparisons to GCL and AP-

GeM ensembles. The ensemble members in the case of our Modular SNN are homogeneous as they share the same network architecture and training data; their differences lie in the random initialization values of weights and the random sequence of input images, as elaborated in Section III-D. The GCL ensembles are also homogeneous with consistent network architecture and training datasets and only differing in random initialization of weights and order of input images. The AP-GeM ensembles are heterogeneous, showing diverse Convolutional Neural Network (CNN) backbone and/or training datasets. The three architectures are a ResNet50 and a ResNet101, both trained on the Landmarks-clean dataset, and a Res101 trained on the Google-Landmarks Dataset, as described in Section IV-E.

We created the GCL and AP-GeM ensembles by averaging the feature representations of each reference and query set across all ensemble members, and then computed the distance matrix based on these averaged representations³.

Figure 7 presents the R@1 performance improvement of our Ensemble of Modular SNNs (both with three, and five ensemble members), ensemble of GCL (both with three, and five ensemble members), and an ensemble of AP-GeM models (with three ensemble members) relative to the mean R@1 of their respective ensemble members across all datasets. Individual Modular SNNs perform relatively consistent with little variation in R@1 performance. The Ensemble of Modular SNNs (both with three, and five ensemble members) consistently outperforms the average R@1 of its individual members. Across all datasets, the ensembles with three members achieve an average R@1 of 32.5%, compared to their individual members' mean R@1 of 26.6%. Similarly, ensembles with five members reach an average R@1 of 35.4%, exceeding their members' mean R@1 of 26.7%.

While the homogeneous GCL ensemble members achieve consistent R@1 performance, which is similar to the consistency in performance of our Modular SNN ensemble members, the GCL ensemble (both with three, and five ensemble members) shows little to no improvement in R@1 over its individual member average (R@1 improvements of less than 2%) in both three and five ensemble member instances. It is likely that the different GCL ensemble members all converge to the same local minima because of the loss function used in this approach.

The ensemble members of the heterogeneous AP-GeM models exhibit a wide range of R@1 performance. The ensemble performance is equal to and/or inferior to the R@1 of the best-performing ensemble member across all six datasets (see Table I). Across all datasets, the mean R@1 for the ensemble is 47.3%, falling slightly below the 48.0% R@1 of the top-performing member, AP-GeM (ResNet-101, LM18), even though the average R@1 of the ensemble members is 38.5%. Consequently, applying the ensembling technique to these AP-GeM models diminishes the performance of the best-performing member.

³Additionally, we explored creating these ensembles through element-wise summation of the distance matrices of all ensemble members. However, we selected the averaged feature representation method because it performed better than the method of combining distance matrices.

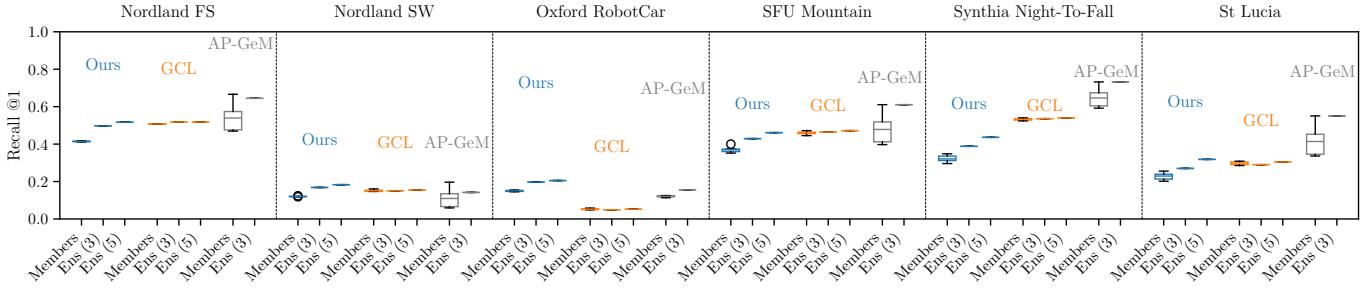


Fig. 7. The effect of ensembling: The plot shows the R@1 performance of Ensembles of Modular SNNs (both with three, and five ensemble members; blue), GCL ensembles (both with three, and five ensemble members; orange), and an AP-GeM ensemble (with three ensemble members; gray) on all six evaluated datasets. Our Ensemble of Modular SNNs show superior R@1 performance over its individual members, while GCL ensembles exhibit minimal gains. The AP-GeM ensemble members have a varied R@1 performance spectrum, with the AP-GeM ensemble performance matching or falling short of its best-performing member across all datasets. For detailed metrics across all datasets, refer to Table I.

TABLE II

ABLATION STUDY ON ENSEMBLE OF THREE MODULAR SNNs WITH AND WITHOUT RANDOMIZATION OF INITIAL WEIGHTS, AND SHUFFLED ORDER OF IMAGES ON THE OXFORD ROBOTCAR DATASET (REFERENCE: RAIN; QUERY: DUSK). BEST CONFIGURATION IS IN **BOLD**.

Randomized Weights	Shuffled Order	R@1
✗	✗	0.10
✓	✗	0.11
✗	✓	0.19
✓	✓	0.20

F. Ablation Study on Member Randomization in Ensemble of Modular SNNs

This section provides an ablation study on the R@1 performance of our Ensemble of Modular SNNs with and without input image order shuffling, and with and without different random weight initialization that is applied to the ensemble members. In our previous conference paper [61], we provided consecutive input images of a traverse to the modules for training. We initialized the weights of all modules using the same random values. Here, to increase the diversity among the ensemble members, we instead shuffled the reference images of each traverse for the training process, and then fed these shuffled images to the modules (Figure 1). Moreover, we initialized the weights of each member, Modular SNN, using different random values, while within each member, using the same set of random values for all modules.

Table II presents the R@1 performance of these four ensemble variants on the Oxford RobotCar dataset, where each ensemble contains three ensemble members. The first variant is an ensemble model where no randomization is applied, resulting in identical ensemble members and thus mirroring the performance of a single member, which is 10.4%. The ensemble members in the second variant differ only in their initial random weights, resulting in a R@1 of 11.0%. The third variant varies the ensemble members with only in the shuffled order of images, yielding in a R@1 performance of 19.1%. Lastly, the fourth variant combines both randomization of weights and shuffled order of images, producing the highest R@1 performance of 19.8%.

Excluding the first variant, which negates the ensembling effect, the remaining three variants show similar R@1 perfor-

mance improvements over their mean R@1 member performances. Randomizing the shuffled order of images significantly enhances ensemble performance compared to just randomizing the weights. The combination of both strategies obtains the highest R@1, albeit with a marginal improvement over the third variant, where only shuffled order of images are randomized.

G. Computational Scalability

Figure 8 provides insights into the scalability and computational efficiency of our Modular SNN and Ensemble of Modular SNNs with two ensemble members, against the Non-modular SNN from [59]. The plot shows that the query times of our Modular SNN and Ensemble of Modular SNNs scale linearly as the number of learned places increases. Meanwhile, the Non-modular SNN faces scalability issues beyond 400 places. We anticipate that implementing our Modular and Ensemble of Modular SNNs on neuromorphic hardware could substantially enhance processing speed through hardware parallelism. This is one of our future research directions, as described in more detail in Section VI.

H. Proof-of-Concept Robot Deployment

We also conducted a proof-of-concept deployment of our Modular SNN on AgileX’s Scout Mini robot [60] in a small indoor environment operating in real-time on a CPU. In this experiment, we first collected the reference dataset containing 100 images and trained our Modular SNN offline with four SNN modules (each assigned to learn 25 place labels). During inference, the robot moved through the environment, collecting images at 1 Hz, and predicting the place labels of the query images. The robot’s speed was approximately 1 m/s during both the collection of the reference set and the inference time.

Figure 9 shows the reference path taken by the robot, with correct and incorrect predictions of our Modular SNN at query time marked by green and red markers, respectively. The figure indicates that the model generally performs well, with R@1 = 75.0%. Most misclassifications occurred at curves or where the query image position is slightly shifted laterally and/or frontally. This is due to the limitation in viewpoint tolerance of our approach, which is discussed in Section VI as part of our future work.

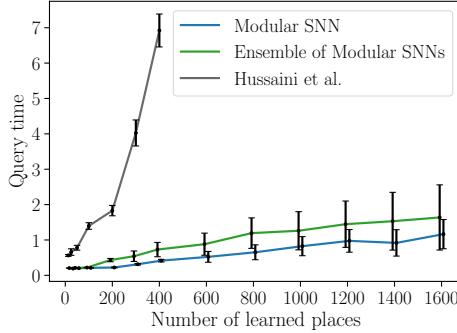


Fig. 8. **Performance scalability comparison:** We show the average query processing time for a single query image as the number of places (and thus overall network size) increases. We contrast our Modular and Ensemble of Modular SNNs with two ensemble members, with the approach from [59]. Both the Modular SNN and Ensemble of Modular SNNs demonstrate to scale linearly with the number of reference places. In contrast, the Non-modular SNN did not scale well, and we were only able to test up to 6400 output neurons (400 places). We expect both our Ensemble and non-Ensemble variants to scale even better and yield even lower inference times on neuromorphic hardware due to their massive parallel processing capabilities.

We used a ground truth tolerance of 3 m, due to the high visual overlap between consecutive places. The inference time of our Modular SNN ranged from 1.1 to 2 seconds per image. This experiment validated the feasibility of our Modular SNN approach for real-time robot deployment in a small indoor environment.

VI. DISCUSSIONS AND CONCLUSIONS

This paper has shed light on the capabilities of spiking neural networks (SNNs) in the realm of visual place recognition. Through a series of enhancements, we have showcased their utility and promise in this field.

Firstly, we introduced scalable SNNs that we dubbed Modular SNNs which represent a small region of the environment and have enhanced adaptability and efficiency in expansive environments. This innovation significantly broadens the applicability of SNNs for place recognition tasks.

Building on the foundation of Modular SNNs, we further enhanced our approach by introducing the Ensemble of Modular SNNs in our second contribution. In this case, multiple SNNs are employed to represent a single place, which demonstrated a substantial improvement in place recognition robustness and generalization ability. We have shown that the responsiveness of SNNs to ensembling is higher compared to conventional techniques that employed homogeneous and heterogeneous ensembles. This is evident as the average R@1 of our Ensemble of Modular SNNs across all datasets is consistently higher than the average R@1 of its ensemble members, highlighting that the ensembling technique significantly amplifies the capabilities of our Modular SNN approach. Moreover, our Ensemble of Modular SNNs has demonstrated competitive performance, in close proximity with some of the leading VPR methods, across various datasets.

Lastly, in addition to ensembling, we also explored the impact of sequence matching, a technique that further augments our system's performance by utilizing multiple consecutive images for place matching. Pairing our Ensemble of Modular SNNs with sequence matching exhibited a higher R@1

performance improvement compared to VPR techniques with similar baselines, except in the case of the Oxford RobotCar dataset. This reinforces the transformative role of sequence matching on enhancing SNN capabilities for visual place recognition. We also provided an indicator for sequence matching responsiveness, applicable to general VPR techniques, which demonstrated the competitive adaptability of our SNN-based solutions to sequence matching compared to that of VPR techniques.

Our work follows the similar trend seen in the recent state-of-the-art conventional VPR techniques such as [52] that frame the visual place recognition problem as a classification task, enabling large-scale recognition capability by bypassing the computationally heavy process of computing the pairwise distance matrix for all query and reference feature representations. We highlight that our approach is trained only on the reference set, which has geographical overlap with the query set of the same dataset used for testing. This strategy is advantageous for real-world applications as it avoids pretraining on large datasets and uses fewer training images compared to conventional VPR techniques, which rely on extensive training datasets that are typically geographically separate from the reference and query set of the dataset used for evaluation. Furthermore, we demonstrate the performance of our SNN approach using low resolution image sizes, which scales well with increasing number of places in terms of storage complexity. In comparison, conventional VPR techniques typically need to store reference image feature descriptors, posing challenges for scaling to large datasets due to associated increase in storage requirements.

In the proof-of-concept CPU-based robot deployment of our work, we only used our Modular SNN approach, instead of our Ensemble of Modular SNN approach, because using multiple Modular SNNs to create the ensemble requires higher memory and results in an increase in latency. Future deployment of our SNN-based approach on neuromorphic hardware can provide significant improvements in the power usage and the latency of our approach.

Looking ahead, our future work aims to leverage these findings and explore new frontiers in neuromorphic computing. We plan to implement our approach on specialized neuromorphic hardware platforms, particularly Intel's Loihi 2 [8], to harness its inherent advantages in obtaining high energy efficiency and reduced latency. Although ensemble methods often face scalability issues, we see potential in neuromorphic computing, known for its exceptional parallel processing capabilities, to address these concerns. Such deployment will enable using our approach as a loop closure component for SLAM, in addition to using it as a re-localization method as presented in Section V. Implementing k-Nearest Neighbor (kNN) on Intel's Loihi [146] has achieved comparable accuracy to CPU with 10 times less power and a latency of just 3.03 ms [7]. A population-coded spiking network for robotic control on Loihi was 140 times more energy-efficient than on Jetson TX2, with similar performance [147]. A spiking network for object classification on Loihi saw only a 3% accuracy reduction compared to CPU, with 0.72 ms latency and 310 mW power consumption [148].

A limitation of our work is the minimal robustness to

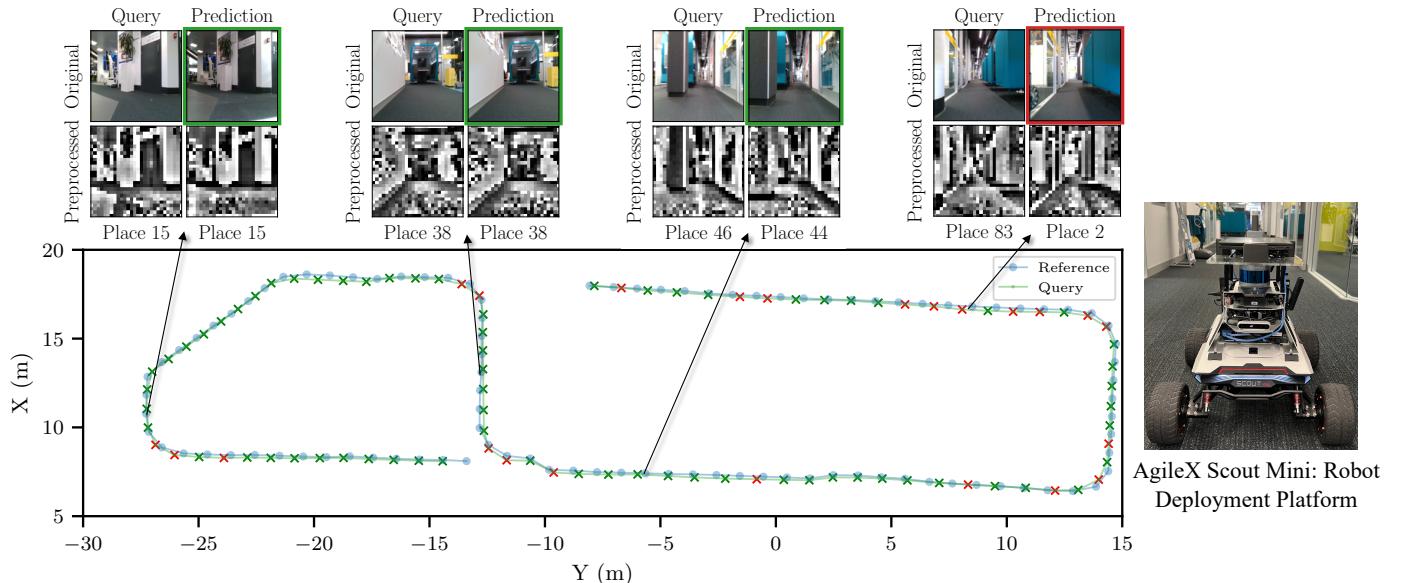


Fig. 9. Robot deployment feasibility study: [Left] Real-time deployment of our Modular SNN in a small indoor environment on a CPU. First, the reference dataset is collected and the Modular SNN is trained offline. During inference, the robot moves through the environment, collecting images, and predicting their place labels in real-time. The reference and query images are collected at different times of the day. The images were collected at 1 Hz with the robot moving at approximately 1 m/s. The blue line and points represent the reference path and images, while the green line represents the query path. Correct predictions are marked with green crosses, and incorrect predictions are marked with red crosses. The four samples of the query and predicted places show the original images and their preprocessed forms, which are used as input to our Modular SNN. [Right] Proof-of-concept robot deployment testing platform. We used an AgileX Scout Mini [60] robot equipped with a Intel's RealSense D435 camera.

viewpoint shift, an aspect that most VPR techniques address effectively. Enhancing the resilience of our system to viewpoint change remains a priority for us, as it is crucial for reliable place recognition in more challenging situations. To overcome this challenge, we can incorporate an attention-based mechanism similar to recent SOTA Transformer-based VPR approaches [149], [150], or divide each input image into smaller patches, each processed by a module, similar to regional descriptors such as PatchNetVLAD [128]. However, the necessity for robustness against significant viewpoint shifts may vary depending on the specific downstream application of our visual place recognition system. For instance, in scenarios where the VPR system acts as a loop closure component of a Simultaneous Localization and Mapping (SLAM) process, the limitations of the SLAM system in loop closure might render extreme viewpoint robustness less critical [20].

We are also exploring the possibility of using event cameras [99] to directly input event data to further enhance energy efficiency and reduce latency, moving away from our current strategy of converting traditional image data to rate-coded event streams. This includes adapting the components of our spiking network architecture, such as neuronal dynamics and learning mechanisms, to the sparse temporal nature of event data and modifying the definition of a place to suit event-based input characteristics.

Our research illuminates the significant potential of SNNs for robotic navigation, presenting a solution that is scalable, and robust for place recognition tasks. Our SNN-based approach is particularly responsive to ensembling and sequence matching techniques, as evidenced by its performance increase when these techniques are applied. These techniques significantly enhance its robustness to high appearance changes, and its generalization ability across diverse environments. These ad-

vancements in SNN technology not only enhance the efficiency of robotic navigation systems but also have vast applicability across various real-world robotics applications. Our findings are particularly promising for resource-constrained robots, such as those deployed in challenging environments including space and underwater, where the focus on edge computing and considerations for SWaP (Size, Weight, and Power) emphasize its suitability for these rigorous settings.

REFERENCES

- [1] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 04, pp. 295–308, 2009.
- [2] Y. Sandamirskaya, M. Kaboli, J. Conradt, and T. Celikel, "Neuromorphic computing hardware and neural architectures for robotics," *Sci. Robot.*, vol. 7, no. 67, p. eabl8419, 2022.
- [3] C. D. Schuman *et al.*, "Opportunities for neuromorphic computing algorithms and applications," *Nat. Comput. Sci.*, vol. 2, no. 1, pp. 10–19, 2022.
- [4] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara, and N. Le, "Spiking neural networks and their applications: A review," *Brain Sci.*, vol. 12, no. 7, p. 863, 2022.
- [5] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
- [6] J. D. Nunes, M. Carvalho, D. Carneiro, and J. S. Cardoso, "Spiking neural networks: A survey," *IEEE Access*, vol. 10, pp. 60 738–60 764, 2022.
- [7] E. P. Frady *et al.*, "Neuromorphic nearest neighbor search using Intel's Pohoiki Springs," in *Proc. Neuro-inspired Comput. Elements Worksh.*, 2020.
- [8] M. Davies *et al.*, "Advancing neuromorphic computing with Loihi: A survey of results and outlook," *Proc. IEEE*, vol. 109, no. 5, pp. 911–934, 2021.
- [9] J. Pei, *et al.*, "Towards artificial general intelligence with hybrid Tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, 2019.
- [10] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, 2014.
- [11] J. Yik, S. H. Ahmed *et al.*, "NeuroBench: Advancing neuromorphic computing through collaborative, fair and representative benchmarking," *arXiv preprint arXiv:2304.04640*, 2023.



Fig. 10. Qualitative results: The plot showcases the performance of our Standalone Modular SNN, Ensemble of Modular SNNs (with five ensemble members), and various VPR methods across diverse datasets. To enhance clarity, we have included just one of the three NetVLAD instances, specifically NetVLAD (Landmarks), and one of the three AP-GeM instances, namely AP-GeM (ResNet101, LM18). It details three instances of correct predictions by both Modular SNN and its Ensemble variant (in rows one, four, and five), two cases where the Ensemble yields correct matches despite the incorrect prediction of the Modular SNN as an ensemble member (in rows two and six), and a situation where both Modular and Ensemble of Modular SNNs fail to correctly match the query image to its corresponding reference image (in row three).

- [12] J. K. Eshraghian *et al.*, “Training spiking neural networks using lessons from deep learning,” *Proc. IEEE*, 2023.
- [13] S. Schubert, P. Neubert, S. Garg, M. Milford, and T. Fischer, “Visual place recognition: A tutorial,” *IEEE Robotics & Automation Magazine*, 2023.
- [14] S. Garg, T. Fischer, and M. Milford, “Where is your place, visual place recognition?” in *Int. Jt. Conf. Artif. Intell.*, 2021, pp. 4416–4425.
- [15] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, 2015.
- [16] K. A. Tsintotas, L. Bampis, and A. Gasteratos, “Visual place recognition for simultaneous localization and mapping,” *Autonomous Vehicles Volume 2: Smart Vehicles*, pp. 47–79, 2022.
- [17] C. Masone and B. Caputo, “A survey on deep visual place recognition,” *IEEE Access*, vol. 9, pp. 19 516–19 547, 2021.
- [18] X. Zhang, L. Wang, and Y. Su, “Visual place recognition: A survey from deep learning perspective,” *Pattern Recognit.*, vol. 113, p. 107760, 2021.
- [19] C. Cadena *et al.*, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [20] K. A. Tsintotas, L. Bampis, and A. Gasteratos, “The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 19 929–19 953, 2022.
- [21] C. Frenkel, D. Bol, and G. Indiveri, “Bottom-up and top-down approaches for the design of neuromorphic processing systems: tradeoffs and synergies between natural and artificial intelligence,” *Proc. IEEE*, 2023.
- [22] G. Auda and M. Kamel, “Modular neural networks: a survey,” *Int. J. Neural Syst.*, vol. 9, no. 02, pp. 129–151, 1999.
- [23] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, “Toward transparent AI: A survey on interpreting the inner structures of deep neural networks,” in *IEEE Conf. Secure Trustworthy Mach. Learn.*, 2023, pp. 464–483.
- [24] M. Amer and T. Maul, “A review of modularization techniques in artificial neural networks,” *Artif. Intell. Rev.*, vol. 52, pp. 527–561, 2019.
- [25] M. Colosi *et al.*, “Plug-and-play SLAM: A unified SLAM architecture for modularity and ease of use,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 5051–5057.
- [26] R. Dubé *et al.*, “Segmatch: Segment based place recognition in 3D point clouds,” in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5266–5272.
- [27] S. Garg, A. Jacobson, S. Kumar, and M. Milford, “Improving condition-

- and environment-invariant place recognition with semantic place categorization,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2017, pp. 6863–6870.
- [28] J.-L. Blanco-Claraco, “A modular optimization framework for localization and mapping,” in *Robot. Sci. Syst.*, 2019.
- [29] M. A. Ganaie, M. Hu, A. Malik, M. Tanveer, and P. Suganthan, “Ensemble deep learning: A review,” *Eng. Appl. Artif. Intell.*, vol. 115, p. 105151, 2022.
- [30] W. Li, Y. Peng, M. Zhang, L. Ding, H. Hu, and L. Shen, “Deep model fusion: A survey,” *arXiv preprint arXiv:2309.15698*, 2023.
- [31] Y. Yang, H. Lv, and N. Chen, “A survey on ensemble learning under the era of deep learning,” *Artif. Intell. Rev.*, vol. 56, no. 6, pp. 5545–5589, 2023.
- [32] T. G. Dietterich, “Ensemble methods in machine learning,” in *Int. Worksh. Multiple Classifier Syst.*, 2000, pp. 1–15.
- [33] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.
- [34] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, p. e1249, 2018.
- [35] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman, and D. Kerr, “EO-SLAM: Monocular semi-dense object SLAM based on ensemble data association,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 4966–4973.
- [36] M. J. Procopio, J. Mulligan, and G. Grudic, “Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments,” *J. Field Robot.*, vol. 26, no. 2, pp. 145–175, 2009.
- [37] B. Arcanjo *et al.*, “A-music: An adaptive ensemble system for visual place recognition in changing environments,” *arXiv preprint arXiv:2303.14247*, 2023.
- [38] C. Malone, S. Hausler, T. Fischer, and M. Milford, “Boosting performance of a baseline visual place recognition technique by predicting the maximally complementary technique,” in *IEEE Int. Conf. Robot. Autom.*, 2023, pp. 1919–1925.
- [39] T. Fischer and M. Milford, “Event-based visual place recognition with ensembles of temporal windows,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6924–6931, 2020.
- [40] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” *Adv. Neural Inform. Process. Syst.*, vol. 30, 2017.
- [41] M. J. Milford and G. F. Wyeth, “SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights,” in *IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1643–1649.
- [42] S. Garg and M. Milford, “SeqNet: Learning descriptors for sequence-based hierarchical place recognition,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4305–4312, 2021.
- [43] S. Garg, M. Vankadari, and M. Milford, “SeqMatchNet: Contrastive learning with sequence matching for place recognition & relocalization,” in *Conference on Robot Learning*, 2022, pp. 429–443.
- [44] S. Schubert, P. Neubert, and P. Protzel, “Fast and memory efficient graph optimization via icm for visual place recognition,” in *Robot. Sci. Syst.*, vol. 73, 2021.
- [45] R. Mereu, G. Trivigno, G. Berton, C. Masone, and B. Caputo, “Learning sequential descriptors for sequence-based visual place recognition,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10 383–10 390, 2022.
- [46] J. M. Facil, D. Olid, L. Montesano, and J. Civera, “Condition-invariant multi-view place recognition,” *arXiv preprint arXiv:1902.09516*, 2019.
- [47] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello, and W. Burgard, “Robust visual SLAM across seasons,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2015, pp. 2529–2535.
- [48] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, “24/7 place recognition by view synthesis,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1808–1817.
- [49] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, 2018.
- [50] J. Revaud, J. Almazán, R. S. Rezende, and C. R. d. Souza, “Learning with average precision: Training image retrieval with a listwise loss,” in *Int. Conf. Comput. Vis.*, 2019, pp. 5107–5116.
- [51] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, “Data-efficient large scale place recognition with graded similarity supervision,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 23 487–23 496.
- [52] G. Berton, C. Masone, and B. Caputo, “Rethinking visual geolocation for large-scale applications,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 4878–4888.
- [53] A. Ali-Bey, B. Chaib-Draa, and P. Giguere, “Mixvpr: Feature mixing for visual place recognition,” in *IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 2998–3007.
- [54] N. Sünderhauf, P. Neubert, and P. Protzel, “Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons,” in *IEEE Int. Conf. Robot. Autom. Worksh.*, 2013.
- [55] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 year, 1000 km: The Oxford RobotCar dataset,” *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.
- [56] J. Bruce, J. Wawerla, and R. Vaughan, “The SFU mountain dataset: Semi-structured woodland trails under changing environmental conditions,” in *IEEE Int. Conf. Robot. Autom.*, 2015.
- [57] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3234–3243.
- [58] 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, 2008.
- [59] S. Hussaini, M. Milford, and T. Fischer, “Spiking neural networks for visual place recognition via weighted neuronal assignments,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4094–4101, 2022.
- [60] A. Robotics, “Scout mini: A small size 4wd mobile robot,” 2024, accessed: 2024-06-10. [Online]. Available: <https://global.agilex.ai/products/scout-mini>
- [61] S. Hussaini, M. Milford, and T. Fischer, “Ensembles of compact, region-specific & regularized spiking neural networks for scalable place recognition,” in *IEEE Int. Conf. Robot. Autom.*, 2023, pp. 4200–4207.
- [62] C. D. Schuman *et al.*, “A survey of neuromorphic computing and neural networks in hardware,” *arXiv preprint arXiv:1705.06963*, 2017.
- [63] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, “Conversion of continuous-valued deep networks to efficient event-driven networks for image classification,” *Front. Neurosci.*, vol. 11, p. 682, 2017.
- [64] T. Bu, W. Fang, J. Ding, P. DAI, Z. Yu, and T. Huang, “Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks,” in *Int. Conf. Learn. Represent.*, 2021.
- [65] J. Ding, Z. Yu, Y. Tian, and T. Huang, “Optimal ANN-SNN conversion for fast and accurate inference in deep spiking neural networks,” in *Int. Jt. Conf. Artif. Intell.*, 2021.
- [66] E. Hunsberger and C. Eliasmith, “Training spiking deep networks for neuromorphic hardware,” *arXiv preprint arXiv:1611.05141*, 2016.
- [67] W. Severa, C. M. Vineyard, R. Dellana, S. J. Verzi, and J. B. Aimone, “Training deep neural networks for binary communication with the whetstone method,” *Nat. Mach. Intell.*, vol. 1, no. 2, pp. 86–94, 2019.
- [68] C. Stöckl and W. Maass, “Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes,” *Nat. Mach. Intell.*, vol. 3, no. 3, pp. 230–238, 2021.
- [69] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, “Enabling spike-based backpropagation for training deep neural network architectures,” *Front. Neurosci.*, p. 119, 2020.
- [70] A. Renner, F. Sheldon, A. Zlotnik, L. Tao, and A. Sornborger, “The backpropagation algorithm implemented on spiking neuromorphic hardware,” *arXiv preprint arXiv:2106.07030*, 2021.
- [71] G. Shen, D. Zhao, and Y. Zeng, “Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks,” *Patterns*, vol. 3, no. 6, 2022.
- [72] G.-q. Bi and M.-m. Poo, “Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type,” *J. Neurosci.*, vol. 18, no. 24, pp. 10 464–10 472, 1998.
- [73] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Front. Comput. Neurosci.*, vol. 9, no. 99, pp. 1–9, 2015.
- [74] I. Abadía, F. Naveros, E. Ros, R. R. Carrillo, and N. R. Luque, “A cerebellar-based solution to the nondeterministic time delay problem in robotic control,” *Sci. Robot.*, vol. 6, no. 58, p. eabf2756, 2021.
- [75] A. Vitale, A. Renner, C. Nauer, D. Scaramuzza, and Y. Sandamirskaya, “Event-driven vision and control for UAVs on a neuromorphic chip,” in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 103–109.
- [76] J. Dupeyroux, J. J. Hagenaars, F. Paredes-Vallés, and G. C. de Croon, “Neuromorphic control for optic-flow-based landing of mavs using the loihai processor,” in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 96–102.
- [77] R. K. Stagsted *et al.*, “Event-based PID controller fully realized in neuromorphic hardware: A one DoF study,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 10 939–10 944.

- [78] J. Ding *et al.*, “Biologically inspired dynamic thresholds for spiking neural networks,” *Adv. Neural Inform. Process. Syst.*, vol. 35, pp. 6090–6103, 2022.
- [79] J. C. V. Tieck *et al.*, “Towards grasping with spiking neural networks for anthropomorphic robot hands,” in *IEEE Int. Conf. Artif. Neural Netw.*, 2017, pp. 43–51.
- [80] J. C. V. Tieck, L. Steffen, J. Kaiser, A. Roennau, and R. Dillmann, “Controlling a robot arm for target reaching without planning using spiking neurons,” in *Int. Conf. Cogn. Inform. Cogn. Comput.*, 2018, pp. 111–116.
- [81] K. M. Oikonomou, I. Kansizoglou, and A. Gasteratos, “A hybrid spiking neural network reinforcement learning agent for energy-efficient object manipulation,” *Machines*, vol. 11, no. 2, p. 162, 2023.
- [82] A. Lele, Y. Fang, J. Ting, and A. Raychowdhury, “An end-to-end spiking neural network platform for edge robotics: From event-cameras to central pattern generation,” *IEEE Trans. Cogn. Develop. Syst.*, vol. 14, no. 3, pp. 1092–1103, 2021.
- [83] Y. Luo, H. Shen, X. Cao, T. Wang, Q. Feng, and Z. Tan, “Conversion of siamese networks to spiking neural networks for energy-efficient object tracking,” *Neural. Comput. Appl.*, vol. 34, no. 12, pp. 9967–9982, 2022.
- [84] R. Kreiser, A. Renner, V. R. Leite, B. Serhan, C. Bartolozzi, A. Glover, and Y. Sandamirskaya, “An on-chip spiking neural network for estimation of the head pose of the iCub robot,” *Front. Neurosci.*, vol. 14, p. 551, 2020.
- [85] A. Renner *et al.*, “Neuromorphic visual scene understanding with resonator networks,” *arXiv preprint arXiv:2208.12880*, 2022.
- [86] M. J. Milford, G. F. Wyeth, and D. Prasser, “RatSLAM: a hippocampal model for simultaneous localization and mapping,” in *IEEE Int. Conf. Robot. Autom.*, 2004, pp. 403–408.
- [87] F. Galluppi *et al.*, “Live demo: Spiking RatSLAM: Rat hippocampus cells in spiking neural hardware,” in *IEEE Biomed. Circuits Syst. Conf.*, 2012, pp. 91–91.
- [88] G. Tang and K. P. Michmizos, “Gridbot: An autonomous robot controlled by a spiking neural network mimicking the brain’s navigational system,” in *Int. Conf. Neuromorphic Syst.*, 2018, pp. 1–8.
- [89] R. Kreiser, A. Renner, Y. Sandamirskaya, and P. Pienroj, “Pose estimation and map formation with spiking neural networks: towards neuromorphic SLAM,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2018, pp. 2159–2166.
- [90] G. Tang, A. Shah, and K. P. Michmizos, “Spiking neural network on neuromorphic hardware for energy-efficient unidimensional SLAM,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2019, pp. 4176–4181.
- [91] N. S.-Y. Dumont, P. M. Furlong, J. Orchard, and C. Eliasmith, “Exploiting semantic information in a spiking neural SLAM system,” *Front. Neurosci.*, vol. 17, 2023.
- [92] R. Kreiser, M. Cartiglia, J. N. Martel, J. Conradt, and Y. Sandamirskaya, “A neuromorphic approach to path integration: a head-direction spiking neural network with vision-driven reset,” in *IEEE Int. Symp. Circuits Syst.*, 2018, pp. 1–5.
- [93] R. Kreiser, G. Waibel, N. Armengol, A. Renner, and Y. Sandamirskaya, “Error estimation and correction in a spiking neural network for map formation in neuromorphic hardware,” in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 6134–6140.
- [94] A. Safa, T. Verbelen, I. Ocket, A. Bourdoux, H. Sahli, F. Catthoor, and G. Gielen, “Fusing event-based camera and radar for SLAM using spiking neural networks with continual STDP learning,” in *IEEE Int. Conf. Robot. Autom.*, 2023, pp. 2782–2788.
- [95] M. Milford, H. Kim, M. Mangan, S. Leutenegger, T. Stone, B. Webb, and A. Davison, “Place recognition with event-based cameras and a neural implementation of SeqSLAM,” *arXiv preprint arXiv:1505.04548*, 2015.
- [96] D. Weikersdorfer, R. Hoffmann, and J. Conradt, “Simultaneous localization and mapping for event-based vision systems,” in *Int. Conf. Comput. Vis. Syst.*, 2013, pp. 133–142.
- [97] A. R. Vidal, H. Rebecq, T. Horstschaefter, and D. Scaramuzza, “Ultimate SLAM? Combining events, images, and imu for robust visual SLAM in HDR and high-speed scenarios,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, 2018.
- [98] T. Fischer and M. Milford, “How many events do you need? Event-based visual place recognition using sparse but varying pixels,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12 275–12 282, 2022.
- [99] G. Gallego *et al.*, “Event-based vision: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2020.
- [100] M. Xu, N. Snderhauf, and M. Milford, “Probabilistic visual place recognition for hierarchical localization,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 311–318, 2020.
- [101] M. Cummins and P. Newman, “Fab-map: Probabilistic localization and mapping in the space of appearance,” *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.
- [102] A.-D. Doan, Y. Latif, T.-J. Chin, Y. Liu, T.-T. Do, and I. Reid, “Scalable place recognition under appearance change for autonomous driving,” in *Int. Conf. Comput. Vis.*, 2019, pp. 9319–9328.
- [103] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3304–3311.
- [104] G. Trivigno, G. Berton, J. Aragon, B. Caputo, and C. Masone, “Divide&classify: Fine-grained classification for city-wide visual geolocation,” in *Int. Conf. Comput. Vis.*, 2023, pp. 11 142–11 152.
- [105] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, “From coarse to fine: Robust hierarchical localization at large scale,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 12 716–12 725.
- [106] C. Fan, Z. Chen, A. Jacobson, X. Hu, and M. Milford, “Biologically-inspired visual place recognition with adaptive multiple scales,” *Robo. and Auton. sys.*, vol. 96, pp. 224–237, 2017.
- [107] S. Hausler and M. Milford, “Hierarchical multi-process fusion for visual place recognition,” in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 3327–3333.
- [108] N. V. Keetha, M. Milford, and S. Garg, “A hierarchical dual model of environment-and place-specific utility for visual place recognition,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6969–6976, 2021.
- [109] E. Garcia-Fidalgo and A. Ortiz, “Hierarchical place recognition for topological mapping,” *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1061–1074, 2017.
- [110] P. Neubert, S. Schubert, and P. Protzel, “A neurologically inspired sequence processing model for mobile robot place recognition,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3200–3207, 2019.
- [111] F. Yu, J. Shang, Y. Hu, and M. Milford, “NeuroSLAM: A brain-inspired SLAM system for 3D environments,” *Biol. Cybern.*, vol. 113, no. 5, pp. 515–545, 2019.
- [112] M. Chancán, L. Hernandez-Nunez, A. Narendra, A. B. Barron, and M. Milford, “A hybrid compact neural architecture for visual place recognition,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 993–1000, 2020.
- [113] Z. Bing, D. Nitschke, G. Zhuang, K. Huang, and A. Knoll, “Towards cognitive navigation: A biologically inspired calibration mechanism for the head direction cell network,” *J. Artif. Intell.*, vol. 2, no. 1, pp. 31–41, 2023.
- [114] T. Y. Tan, L. Zhang, C. P. Lim, B. Fielding, Y. Yu, and E. Anderson, “Evolving ensemble models for image segmentation using enhanced particle swarm optimization,” *IEEE access*, vol. 7, pp. 34 004–34 019, 2019.
- [115] T. Fischer, H. J. Chang, and Y. Demiris, “RT-GENE: Real-time eye gaze estimation in natural environments,” in *Eur. Conf. Comput. Vis.*, 2018, pp. 334–352.
- [116] Y. Shim, A. Philippides, K. Staras, and P. Husbands, “Unsupervised learning in an ensemble of spiking neural networks mediated by ITDP,” *PLoS Comput. Biol.*, vol. 12, no. 10, p. e1005137, 2016.
- [117] S. Yang, B. Linares-Barranco, and B. Chen, “Heterogeneous ensemble-based spike-driven few-shot online learning,” *Front. Neurosci.*, vol. 16, p. 850932, 2022.
- [118] Q. Fu and H. Dong, “An ensemble unsupervised spiking neural network for objective recognition,” *Neurocomputing*, vol. 419, pp. 47–58, 2021.
- [119] D. Elbrecht *et al.*, “Evolving ensembles of spiking neural networks for neuromorphic systems,” in *IEEE Symp. Ser. Comput. Intell.*, 2020, pp. 1989–1994.
- [120] J. Yin and Y. Meng, “Reservoir computing ensembles for multi-object behavior recognition,” in *Int. Jt. Conf. Neural Netw.*, 2012, pp. 1–8.
- [121] G. Srinivasan, P. Panda, and K. Roy, “Spiline: Spiking liquid-ensemble computing for unsupervised speech and image recognition,” *Front. Neurosci.*, vol. 12, p. 524, 2018.
- [122] P. Panda, G. Srinivasan, and K. Roy, “EnsembleSNN: Distributed assistive STDP learning for energy-efficient recognition in spiking neural networks,” in *Int. Joint Conf. Neural Networks*, 2017, pp. 2629–2635.
- [123] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, “Robust visual robot localization across seasons using network flows,” in *AAAI Conf. Artif. Intell.*, vol. 28, no. 1, 2014.
- [124] P. Hansen and B. Browning, “Visual place recognition using hmm sequence matching,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2014, pp. 4549–4555.
- [125] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, “Towards life-long visual localization using an efficient matching of binary

- sequences from images,” in *IEEE Int. Conf. Robot. Autom.*, 2015, pp. 6328–6335.
- [126] S. Garg, B. Harwood, G. Anand, and M. Milford, “Delta descriptors: Change-based place representation for robust visual localization,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5120–5127, 2020.
- [127] M. Xu, S. Garg, M. Milford, and S. Gould, “Deep declarative dynamic time warping for end-to-end learning of alignment paths,” in *Int. Conf. Learn. Represent.*, 2022.
- [128] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, “PatchNetVLAD: Multi-scale fusion of locally-global descriptors for place recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 14 141–14 152.
- [129] L. G. Camara and L. Přeučil, “Visual place recognition by spatial matching of high-level CNN features,” *Rob. Auton. Syst.*, vol. 133, p. 103625, 2020.
- [130] 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.
- [131] T. L. Molloy, T. Fischer, M. Milford, and G. N. Nair, “Intelligent reference curation for visual place recognition via bayesian selective fusion,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 588–595, 2020.
- [132] P. Neubert and S. Schubert, “Hyperdimensional computing as a framework for systematic aggregation of image descriptors,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 16 938–16 947.
- [133] S. Lowry and H. Andreasson, “Lightweight, viewpoint-invariant visual place recognition in changing environments,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 957–964, 2018.
- [134] M. Zaffar *et al.*, “VPR-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change,” *Int. J. Comput. Vis.*, vol. 129, no. 7, pp. 2136–2174, 2021.
- [135] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [136] O. Russakovsky *et al.*, “Imagenet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.
- [137] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, “Large-scale image retrieval with attentive deep local features,” in *Int. Conf. Comput. Vis.*, 2017, pp. 3456–3465.
- [138] F. Warburg *et al.*, “Mapillary street-level sequences: A dataset for lifelong place recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 2626–2635.
- [139] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, “Visual place recognition with repetitive structures,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 883–890.
- [140] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.
- [141] F. Radenović, G. Tolias, and O. Chum, “Fine-tuning CNN image retrieval with no human annotation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, 2018.
- [142] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.
- [143] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, “Deep image retrieval: Learning global representations for image search,” in *Eur. Conf. Comput. Vis.*, 2016, pp. 241–257.
- [144] H. Wang *et al.*, “Cosface: Large margin cosine loss for deep face recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 5265–5274.
- [145] S. Schubert and P. Neubert, “What makes visual place recognition easy or hard?” *arXiv:2106.12671*, 2021.
- [146] M. Davies *et al.*, “Loihi: A neuromorphic manycore processor with on-chip learning,” *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.
- [147] G. Tang, N. Kumar, R. Yoo, and K. Michmizos, “Deep reinforcement learning with population-coded spiking neural network for continuous control,” in *Conference on Robot Learning*, 2021, pp. 2016–2029.
- [148] A. Viale *et al.*, “Carsmn: An efficient spiking neural network for event-based autonomous cars on the loihi neuromorphic research processor,” in *Int. Jt. Conf. Neural Netw.*, 2021, pp. 1–10.
- [149] N. Keetha *et al.*, “Anyloc: Towards universal visual place recognition,” *IEEE Robot. Autom. Lett.*, 2023.
- [150] R. Wang *et al.*, “Transvpr: Transformer-based place recognition with multi-level attention aggregation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 13 648–13 657.



Somayeh Hussaini (Member, IEEE) received a Bachelor of Engineering in Mechatronics with First Class Honours from the Queensland University of Technology (QUT), Brisbane, Australia, in 2020. In 2021, she began pursuing a Ph.D. at the same university, which is currently under external examination. She was a QUT finalist in the 2021 3MT Competition and a finalist in the 2022 Queensland Women in STEM Prize. In 2024, she started her role as a postdoctoral research fellow at QUT. Her research interests include robotics, computer vision, and neuromorphic computing.



Michael Milford, FTSE (Senior Member, IEEE) received the Bachelor of Mechanical and Space Engineering degree (2002) and the Ph.D. degree in electrical engineering (2006) from The University of Queensland, Brisbane, QLD, Australia.

Professor Milford’s research models the neural mechanisms in the brain underlying tasks like navigation and perception to develop new technologies in challenging application domains such as all-weather, anytime positioning for autonomous vehicles. He currently holds the position of Director of the QUT

Centre for Robotics, Australian Research Council Laureate Fellow, Professor at the Queensland University of Technology, and is a Microsoft Research Faculty Fellow and Fellow of the Australian Academy of Technology and Engineering.



Tobias Fischer (Senior Member, IEEE) received the B.Sc. degree from Ilmenau University of Technology, Ilmenau, Germany, in 2013, the M.Sc. degree in artificial intelligence from University of Edinburgh, Edinburgh, U.K., in 2014, and the Ph.D. degree from the Personal Robotics Laboratory, Imperial College London, London, U.K., in 2018.

Dr Fischer combines his expertise in robotics, computer vision, and artificial intelligence to provide robots with perceptual abilities allowing safe, intelligent interactions with humans in real-world environments.

Dr Fischer was recently awarded the prestigious Discovery Early Career Researcher Award (DECRA) by the Australian Research Council. His PhD thesis received the UK Best Thesis in Robotics Award 2018 and the Eryl Cadwaladr Davies Award for the best thesis in Imperial’s EEE Department in 2017–2018. He also received multiple best paper awards, including the 2023 IEEE Transactions on Cognitive & Developmental Systems Outstanding Paper Award.