Spiking Neural Networks for Visual Place Recognition via Weighted Neuronal Assignments

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

Abstract—Spiking neural networks (SNNs) offer both compelling potential advantages, including energy efficiency and low latencies and challenges including the non-differentiable nature of event spikes. Much of the initial research in this area has converted deep neural networks to equivalent SNNs, but this conversion approach potentially negates some of the advantages of SNN-based approaches developed from scratch. One promising area for high-performance SNNs is template matching and image recognition. This research introduces the first high-performance SNN for the Visual Place Recognition (VPR) task: given a query image, the SNN has to find the closest match out of a list of reference images. At the core of this new system is a novel assignment scheme that implements a form of ambiguity-informed salience, by up-weighting single-place-encoding neurons and down-weighting “ambiguous” neurons that respond to multiple different reference places. In a range of experiments on the challenging Nordland, Oxford RobotCar, SPEDTest, Synthia, and St Lucia datasets, we show that our SNN achieves comparable VPR performance to state-of-the-art and classical techniques, and degrades gracefully in performance with an increasing number of reference places. Our results provide a significant milestone towards SNNs that can provide robust, energy-efficient, and low latency robot localization.

Index Terms—Neurorobotics, Localization, Autonomous Vehicle Navigation

I. INTRODUCTION

SPIKING neural networks (SNNs) differ in several aspects from conventional artificial neural networks (ANNs) [1]. A neuron’s output within an ANN is a real number computed by a non-linear function of the sum of the inputs. This computation is different from the spikes (also called events) in SNNs, which carry a single bit of information [1]–[3]. Another difference is the learning algorithm: ANNs rely on the backpropagation of error signals, which in the standard implementation does not have an analogue in the brain [3], [4]. In contrast, the brain uses unsupervised learning methods like Spike Time Dependent Plasticity (STDP), which are also used to train SNNs [5]. Although biological insights or fidelity are not the primary focus of this paper, the premise is that SNNs are better suited to study the operation of biological systems and for (in-)validating hypotheses [1], [3].

Recently, SNNs have gained popularity in the robotics community [6]–[13], due to potential advantages over ANNs including low power consumption and low latencies. The community has access to a growing neuromorphic hardware set, such as Intel’s Loihi [2]. SNNs have e.g. been used in robot grasping [6], robot navigation [7], and motion planning [8]. They are however not yet widely competitive; the resulting accuracy is generally lower compared to conventional non-spiking approaches. Current practical use (beyond scientific curiosity) is limited to narrow cases like high-speed or low power scenarios, or where closed-loop control with low latencies is required [2].

In this paper, we take a significant step towards bridging the performance divide between conventional and SNN-based techniques for the Visual Place Recognition task (VPR). VPR is critical in enabling long-term mobile robot autonomy in challenging environments: it can aid in the global re-localization of mobile robots, minimize model errors by acting as a closed-loop detection algorithm, and can be used as a component of robust localization and mapping algorithms [14], [15].

Our research here makes the following contributions:

1) We introduce the first SNN for the place recognition task, combining leaky integrate and fire neurons within a lightweight, biologically plausible SNN to represent place templates using unsupervised learning via STDP.
2) We introduce multiple novel neuronal assignment schemes that decode all neurons’ spike responses and weight them based on the specificity or ambiguity of a single neuron’s response to learning places.
3) We demonstrate competitive performance in terms of recall at 100% precision when comparing our SNN to
conventional VPR baselines, namely NetVLAD [16] and the Sum-of-Absolute Differences (SAD) [17], on the Nordland [18], Oxford RobotCar [19], SPEDTest [20], Synthia [21], and St Lucia [22] datasets.

4) We present ablation studies showing graceful performance degradation for a fixed network size as the number of places to be learned increases.

To foster future research, we make the code to replicate our experiments available with additional supporting materials and ablation studies: https://github.com/QVPR/VPRSNN.

II. RELATED WORKS

Here we review recent Spiking Neural Network research with an emphasis on robotic-related applications in Section II-A and an overview of Visual Place Recognition in Section II-B.

A. Spiking Neural Networks

The field of neuromorphic computing concerns the development and advancement of neuromorphic hardware, sensors, and associated algorithms. Diehl and Cook [23]’s two-layer Spiking Neural Network (SNN) model demonstrated the potential capability of SNNs in solving pattern recognition tasks such as digit recognition. More broadly, SNNs have been evaluated in various robotics applications [6–13], 24, including head pose tracking and scene understanding [11], prey tracking [12], and control of high-speed unmanned aerial vehicles [13].

Biologically-inspired SNNs have in particular been developed for robot navigation-related tasks including Simultaneous Localization and Mapping (SLAM) [25]. These works include RatSLAM, originally proposed by [26], on a neuromorphic hardware using place, grid and border cells [27], place cells for 3-dimensional path planning [28], spatio-temporal memory [24], and pose estimation and mapping [10] demonstrated on simulated environments. Similar works consider mapping constraint indoor environments which include an energy-efficient uni-dimensional SLAM [9], and a neuromorphic-based navigation method on an autonomous robot [7]. Our work is thus complementary to these works on SLAM by providing a loop closure detection algorithm in long-term and large-scale mapping of real-world environments.

Recently, event cameras have gained popularity as a sensing modality to use for SLAM. Event cameras output pixel-level intensity changes, which can result in significant information reduction compared to conventional images [29]. Event-based input has been used for robotic navigation [30], 31, and in combination with other sensors for a quadrotor navigation system [32]. Other works for robot navigation have used event-based data as input to SNNs. These works tackled estimation of a robot’s orientation for path integration [33], and map formation and pose estimation tasks [34] which are deployed on neuromorphic devices and are applicable to SLAM solutions.

While SNNs and related neuromorphic hardware have promising properties, there is often still a performance gap when compared to modern conventional learning- and algorithmic-based techniques. Due to the non-differentiable neuronal dynamics of spiking neurons, supervised techniques such as back-propagation cannot be directly applied. Recent works [35], [36] have proposed algorithms that approximate backpropagation, and demonstrated use on Intel’s Loihi.

Another common approach to implement high-performing SNNs is through converting Artificial Neural Networks (ANNs) to their equivalent SNNs [37], [38]. These models can achieve comparable performance to their ANN predecessors, but may not necessarily be an ideal form of SNNs in the long term due to not fully exploiting the advantages of computational SNNs; in particular, unsupervised methods and methods that approximate backpropagation can lead to spike-timing codes with more optimized latency and energy efficiency [2].

B. Visual Place Recognition

Visual place recognition (VPR) is the task of recognizing whether a place has been already visited given a current image of the place, and prior knowledge of previously visited places [14], 15. The VPR problem can be considered as a template matching problem, where the closest match of a query image is selected from a reference set of places that can be captured from significantly different viewpoints and appearances.

Early works for the VPR problem implemented hand-crafted feature representations – either from local points or by aggregating features over the whole image [17], [39], [40]. Many of those hand-crafted approaches have subsequently been turned into deep neural networks, enabling feature representation learning.

The most popular of these approaches is arguably NetVLAD [16], based on the Vector of Locally Aggregated Descriptors (VLAD) [40], which implements a pooling layer with differentiable operations to allow end-to-end training. NetVLAD has formed the foundation for a substantial number of follow-on systems [41], [42], but is still competitive performance-wise, and is used as a baseline in our experiments (Section VI-B).

One conceptually relevant method is Bayesian Selective Fusion [33], which finds the best match using several sets of reference images, similar to the use of multiple reference sets here to train the SNN. Other relevant work includes that of Frady et al. [44] who leverage the advantages of Intel’s neuromorphic Loihi processor [2] with a scalable k-Nearest Neighbor (kNN) search algorithm. This work is of particular importance, as it shows that SNNs, solving similar tasks to ours, can achieve similar accuracy as CPU implementations, while having low latency (≈3ms) and high throughput (>300 queries/s) and low power (≈10W).

III. METHODOLOGY: PRELIMINARIES

Implementation of SNN-based models can pose a significant technical learning curve, especially for researchers coming from a conventional neural network background. In the following sections, we attempt to provide sufficient detail to facilitate replication. We introduce the SNN model by Diehl and Cook for unsupervised digit recognition [23], upon which we build our new system. The SNN contains biologically plausible mechanisms including Leaky-Integrate-and-Fire (LIF) neurons, Spike-Time Dependent Plasticity (STDP), lateral inhibition, and homeostasis, which enable unsupervised learning.
The model consists of two layers, the input layer and the processing layer (Figure 1). The input layer converts input images into spike trains using rate coding. The input layer contains one neuron per (resized) image pixel; i.e. the number of neurons is \( K_I = W \times H \), where \( W \) and \( H \) are the width and height of the input image, respectively. The input image is converted into Poisson-distributed spike trains by using each input pixel intensity as a firing rate (frequency of spikes) [45].

The neurons in the input layer are fully connected (all-to-all connection) to all \( K_p \) excitatory neurons in the processing layer. The processing layer also contains \( K_p \) inhibitory neurons. The \( k \)-th inhibitory neuron receives input from the corresponding excitatory neuron \( k \), and in turn, inhibits all excitatory neurons except the one that it receives the connection from. These inhibitory connections provide lateral inhibition enabling a model behavior similar to a winner-takes-all system.

The neuronal dynamics for all neurons are modeled using the LIF model, which is a common representation of biological spiking neuronal dynamics [8], [35], [46]. The differential equation describing the LIF dynamics (membrane voltage) is:

\[
\frac{dV}{dt} = (E_{\text{rest}} - V) + g_e(E_{\text{exc}} - V) + g_i(E_{\text{inh}} - V),
\]

where \( E_{\text{rest}} \) is the resting membrane potential, \( E_{\text{exc}} \) and \( E_{\text{inh}} \) are the excitatory and inhibitory synapses’ equilibrium potentials with associated synaptic conductances \( g_e \) and \( g_i \).

The synaptic weights (representing synaptic strength) of the connections between the excitatory and inhibitory neurons are constant. The neuronal learning is done by adapting the synapses (connections) from the input layer to the excitatory neurons using STDP, which models the behavior of biological neurons where the magnitude and direction of change in synaptic strength (weight) is affected by the precise timing of presynaptic and postsynaptic spikes. Over a period of iterating over reference images in training, the synapses’ weights increase the sensitivity of neurons to respond to different stimuli. In particular, all synaptic weights are modeled by the change in synaptic conductance. After a presynaptic input neuron sends a spike to the excitatory postsynaptic neuron, the synaptic connection is increased, otherwise, the conductance decays exponentially:

\[
\tau_{ge} \frac{dg_e}{dt} = -g_e, \tag{2}
\]

where the excitatory postsynaptic potential time constant is denoted by \( \tau_{ge} \). For synapses with an inhibitory presynaptic neuron, the same model is used to update conductance \( g_i \), with inhibitory postsynaptic time constant \( \tau_{gi} \).

Neuronal learning with STDP takes place by adapting the synaptic weights between the input layer and the excitatory neurons in the processing layer. The weights are modeled using presynaptic traces \( s_{\text{pre}} \), which are counters that record the number of presynaptic spikes. The change \( \Delta w \) in synaptic weights after receiving a postsynaptic spike is modeled by:

\[
\Delta w = \eta(x_{\text{pre}} - x_{\text{tar}})(w_{\text{max}} - w)\mu, \tag{3}
\]

where \( \eta \) is the learning rate, \( x_{\text{tar}} \) is the value of the presynaptic trace when a postsynaptic spike is received, \( w_{\text{max}} \) is the maximum weight, and \( \mu \) is a ratio to represent the dependence of update on previous weight.

To ensure all output neurons respond to different patterns, and that the number of spikes fired by neurons for different labels remains approximately within a limited number, a bio-inspired mechanism, homeostasis, is deployed. Homeostasis is implemented by an adaptive neuronal threshold for excitatory neurons. The inner voltage threshold to raise a spike is increased by a constant \( \theta \) after a neuron fires and decreases exponentially. For details about these mechanisms, we refer the interested reader to [23] and our open-source code.

IV. METHODOLOGY: NEURONAL ASSIGNMENTS

Diehl and Cook [23] considered the digit classification task, where each of the excitatory neurons unambiguously learned to represent one of ten digits (Section IV-A). This assumption does not hold for the visual place recognition task, where each place is considered as a separate class, resulting in a much larger number of classes [23]. Therefore, a significant number of neurons learn to represent more than one single place (i.e. they fire to more than one place at training time). Section IV-B introduces our novel weighted neuronal assignment strategy, taking a fine-grained account of the firing patterns at training time. Finally, Section IV-C defines a probability-based re-weighting scheme that increases the consistency between spike rates of the neuronal population.

A. Standard Assignments

The neuronal assignment scheme implemented by Diehl and Cook [23] for digit recognition, which we refer to as standard assignments, is introduced here as follows as it serves as a baseline. The excitatory neurons were assigned to place labels based on the highest average response of the neurons to a place label in all training images \( r \in R \). The place label \( l \) is assigned to neuron \( i \) if the sum of all spikes at training time was the highest for that label:

\[
N_i = \arg \max_l S_{i,l}^R \tag{4}
\]

where \( N_i \) is the assignment of neuron \( i \) and \( S_{i,l}^R \) is the total number of spikes of neuron \( i \) when given images with label \( l \) as input. We also define the set \( M_l \) that contains all neurons that are assigned to label \( l \), i.e. \( M_l = \{ i \mid N_i = l \} \).

At inference time, the predicted place label \( l^* \) for a given query image \( q \) is the place label \( l \) that, summed over all neurons that were assigned to that label (i.e. \( m \in M_l \)), had the highest spike rate:

\[
l^* = \arg \max_l \sum_{m \in M_l} S_{m,l}^q \tag{5}
\]

B. Weighted Assignments

This section introduces our novel weighted neuronal assignment scheme. We considered each input frame as a new place, such that \( R \) images of a traverse are considered as \( R \) distinct

\footnote{Note that we use the terms visual place recognition and place classification synonymously.}
places. With $K_P$ excitatory neurons to distinguish between $R$ places, experimental observation revealed that the majority of these output neurons can fire spikes in response to multiple similar places at different intervals in training. Of responsive neurons, there are three groups: 1) neurons that continuously only raise spikes to one particular place label, 2) neurons that raise spikes to two labels, and 3) neurons that fire spikes for many labels.

However, Diehl and Cook’s standard assignment scheme does not consider the responsiveness of neurons to multiple places. This led to our motivation for normalizing the neurons’ spike rates according to the number of places that they have been assigned to at training time, in order to improve performance. Our proposed weighted assignments comprises of the following three strategies that work in conjunction. The strategies are applied one after the other and introduce just a single parameter $\gamma$ that is kept constant across all experiments.

a) Regularization by involvement: The first normalization step lessens the contribution of neurons that have learned many place labels. Specifically, the spike rates of neurons that have learned more than $\gamma\%$ of place labels are regularized by the number of place labels that they have learned:

$$S_{i,l}^{Q'} = \begin{cases} 
S_{i,l}^{Q} & \text{if } \omega_i \leq \gamma \cdot R \\
\frac{1}{\omega_i}S_{i,l}^{Q} & \text{otherwise}
\end{cases} \quad (6)$$

where $S_{i,l}^{Q}$ are the regularized spike rates and $\omega_i = \text{card}(\{i | N_i \in M_l\})$ is the number of place labels represented by neuron $i$.

b) Normalization by response strength: The next normalization step considers how strongly neuron $i$ has learned label $l$. This normalization step accounts for the strength of the responses; effectively measuring to what percentage label $l$ was learned compared to all labels $m$ that were learned.

$$S_{i,l}^{Q} = S_{i,l}^{Q'} \times \frac{S_{i,l}^{R}}{\sum_m S_{i,m}^{R}} \quad (7)$$

c) Penalize relevant neurons that did not spike: The last normalization step down-weights place labels $l$ when not all neurons $m \in M_l$ that have learned that place label, fired when seeing the query image. The normalization factor is the number of spikes (at training time) of the neurons that also fired at query time over the number of spikes of all neurons that have learned this label at training time:

$$\tilde{S}_{i,l}^{Q} = S_{i,l}^{Q'} \times \frac{\sum_m S_{i,m}^{R} \cdot S_{i,m}^{Q,Q',>0}}{\sum_m S_{i,m}^{R}} \quad (8)$$

In our code repository, we show an ablation study that demonstrates that the combination of all three strategies leads to the best results, outperforming the use of single strategies or pairs of strategies.

C. Probability-based Neuronal Assignments

Despite the addition of weighted assignments, the number of spikes of different query images varies across the dataset, complicating the assessment of whether a place match has been made. To account for the variation in spike numbers for different images, we converted the output spike numbers for every query image to a probability-based array. The probability-based spike numbers $\tilde{S}_{i,l}^{Q}$ are achieved by min-max normalization scaled by the total number of spikes across all output neurons, as follows:

$$\tilde{S}_{i,l}^{Q} = \frac{1}{\sum_k \max_k \tilde{S}_{k,l}^{Q} - \min_k \tilde{S}_{k,l}^{Q}} \left( \tilde{S}_{i,l}^{Q} - \min_k \tilde{S}_{k,l}^{Q} \right) \quad (9)$$

This method can be added as an additional step to both standard assignments and weighted assignments to increase consistency of spike rates for each neuron.

V. EXPERIMENTAL SETUP

This section presents implementation details in Section V-A, followed by the datasets and evaluation metrics used in Sections V-B and V-C respectively.

A. Implementation Details

We implemented the SNN model in Python3 using the Brian2 simulator [47] (CPU version) and BindsNET [46] (GPU version). We base our implementation on [23]. All reference and query images were resized to $W \times H = 28 \times 28$ pixels. The images were also patch-normalized [17] with patch sizes of $W_p \times H_p = 7 \times 7$ pixels to produce compact image representations.

The input images are converted from pixel values to Poisson spike trains using rate coding. This encoding method defines a firing rate for each input pixel based on its intensity. The pixel intensities are scaled to cap the frequencies of input spikes to a range of 0 - 63.75 Hz. The $K_f = 784$ neurons in the first layer of the network each receive a spike train corresponding to the intensity of a single pixel in the input image. For the regularization term in the weighted assignments, we used a $\gamma$ value equal to 2% of the number of reference places.

We trained the SNN model by presenting each input image for 350 ms and defined a resting period of 150 ms to allow neuronal dynamics to decay to their initial values. For a given image, the output of the model is the number of spikes raised by the $K_P = 400$ excitatory neurons in the output layer. The model output is recorded after presenting each image. We trained the models on the reference dataset for 60 epochs.

B. Datasets

To evaluate the model performance, we used five widely established datasets that provide sequential images captured under different appearance conditions: Nordland [18], Oxford RobotCar [19], SPEDTest [20], Synthia [21], and St Lucia [22]. We used the same model configuration to train all datasets and without any dataset-specific fine-tuning.

2Note that most $S_{i,m}^{R}$ are zero.
The Nordland dataset captures a train journey along a 728 km long route in Norway, with one traverse each in spring, summer, fall, and winter. As typically done in the literature in [43], [48], all tunnels, stationary periods, and sections where the train travels below 15 km/h (based on GPS data) were removed. The frames have one-to-one correspondences across seasons, i.e. frame $k$ in one season matches exactly frame $k$ in the other seasons. We used the spring and full traverses as the reference dataset and the summer traverse as the query dataset. To increase independence between each place label, we sampled places every 8 seconds (roughly every 100 meters) from the first quarter of the data.

The Oxford RobotCar dataset contains over 100 traversals around Oxford, with variations in weather conditions, times of the day and seasons, and centimeter-accurate annotations. Following [43], we selected the sun and rain traverses for reference, and the dusk traverse as the query dataset. We sampled places roughly every 10 meters.

The SPEDTest dataset consists of images obtained from surveillance cameras around the world under various weather, season, and illumination conditions with frame-to-frame correspondences [20]. The Synthia dataset is a synthetic dataset in a city-like driving scenario, originally created for semantic scene understanding. As in [49], we used the SEQS-04 fog (reference) and night (query) traverses. We removed the sections where the vehicle is stopped and roughly sampled places at about 1 fps. The St Lucia dataset contains multiple traverses of a route through the St Lucia suburb, Brisbane. We used an early morning traverse (reference), and a late afternoon traverse (query). We removed the sections where the vehicle is stationary and roughly sampled places every 20 meters.

Since the performance of different methods varies significantly depending on the particular section of the dataset that is being used [50], we performed comparisons on 15 non-overlapping sections on Nordland, 4 non-overlapping sections on Oxford RobotCar, 6 non-overlapping sections of SPEDTest, 1 section of Synthia, and 1 section of St Lucia.

### C. Evaluation Metrics

We use recall at 100% precision (R@100P) and area under the precision-recall curve (AUC) to evaluate the performance of the SNN model, which is consistent with many works on VPR [14], [39], [51]. R@100P is an important metric when

---

**TABLE I**

| Dataset       | Standard | Weighted | Probability-based | Weighted probability-based | NetVLAD | SAD |
|---------------|----------|----------|-------------------|---------------------------|---------|-----|
| Nordland      | 12.8%    | 29.7%    | 38.2%             | 47.5%                     | 15.3%   | 51.7%|
| Oxford RobotCar| 2.5%     | 8.8%     | 4.8%              | 12.0%                     | 11.0%   | 11.3%|
| SPEDTest      | 10.2%    | 19.8%    | 29.5%             | 18.7%                     | 15.2%   | 7.7% |
| Synthia       | 1.0%     | 4.0%     | 22.0%             | 16.0%                     | 15.0%   | 19.0%|
| St Lucia      | 12.0%    | 22.0%    | 13.0%             | 22.0%                     | 6.0%    | 0.0% |

---

**VI. RESULTS**

We first compare different neuronal assignment schemes on the loop closure detection task in Section VI-A. This is followed by a comparison of our method with existing approaches for VPR, namely the widely used NetVLAD [16] and Sum-of-Absolute-Differences [17] (Section VI-B). Finally, we evaluate performance for a fixed network size when increasing the number of reference images in Section VI-C.

**A. Loop Closure Detection**

Table I and Fig. 2 present a comparison of the model performance using standard, weighted, probability-based, and weighted probability-based assignments on the Nordland, Oxford RobotCar, SPEDTest, Synthia, and St Lucia datasets. Compared to standard assignments, the weighted and probability-based assignments improve R@100P on average by 9.1% and 13.8% respectively (absolute increase). Our proposed combination of both of these methods, the weighted probability-based assignment, significantly improves the model performance on these datasets ($p < 0.001$; paired $t$-test).

Figure 4 shows the distance matrices of the SNN model using 100 places from the Nordland dataset, obtained by subtracting
the spike rates from the maximum observed spike rate (higher spike rates indicate higher similarities). Thus lower values indicate better matches; an ideal VPR system would have low values on the diagonal and high values elsewhere. In Figure 3, the standard assignment achieved 78.0% precision at 100% recall (P@100R) and a R@100P of 2.1%. The weighted assignments increased P@100R by 4.0% and R@100P by 29.3%.

Figure 4 presents the SNN model performance using standard, weighted, probability-based, and weighted probability-based assignments. Consistent with the results presented in Figure 2 and Table I, the weighted and probability-based assignments increase recall at 100% precision compared to the standard assignments. The weighted probability-based assignments outperform standard assignments and maintains and/or improves performance compared to weighted and probability-based assignments.

Figure 5 shows a qualitative example; in particular, the query place, the place matched by the SNN model with weighted probability-based assignment, and the corresponding reference place for the given query image. The figure includes the templates learned by the neurons assigned to that place label.

B. Comparison to State-of-the-art VPR Approaches

This section provides a comparison of our proposed SNN model with the standard VPR methods, namely the simplistic Sum of Absolute Differences (SAD) [17] and a state-of-the-art approach, NetVLAD [16].

SAD measures the pixel-wise difference between the reference and query images. For a fair comparison, the input images of the SAD model were pre-processed with the same configuration as our SNN model (same image size and patch normalization; see Section V-A). NetVLAD is known to generalize well across different datasets, having high robustness to both viewpoint and appearance changes. Due to the VGG backbone, NetVLAD requires input images to be at least 240x240 pixels, which is significantly higher than the input resolution to our SNN model and SAD.

Table I and Fig. 2 present a comparison of SNN model performance using standard, weighted, probability-based and weighted probability-based assignments as well as NetVLAD and SAD methods. For the Nordland dataset, the mean R@100P of the SNN model with weighted probability-based assignment (47.5%) is comparable to the performance of SAD with an average R@100P of 51.7%. We note that the performance of NetVLAD on the Nordland dataset is known to be relatively low due to the different training dataset [15], [49]. For Oxford RobotCar dataset, the mean R@100P of the SNN model with weighted (8.8%) and weighted probability-based (12.0%) is comparable to that of NetVLAD (11.0%) and SAD (11.3%). The Oxford RobotCar dataset is challenging due to appearance change (sun and rain reference images and dusk queries) and occlusions.

For the SPEDTest dataset, our SNN model achieves a mean R@100P of 18.7%, outperforming both NetVLAD (15.2%) and SAD (7.7%). For the Synthia dataset, our method achieves a R@100P of 16.0%, which is similar to the performance of NetVLAD (15.0% R@100P) and slightly worse than SAD (19.0% R@100P). For the St Lucia dataset, our SNN method outperforms both NetVLAD (6.0% R@100P) and SAD (0.0% R@100P) with a R@100P of 22.0%.

We note that SAD performs well on the Nordland dataset as this dataset has minimum viewpoint change, which makes it well suited for SAD that calculates patch-wise differences to compute similarity (as highlighted in literature, some techniques are better suited than others for some datasets [50]). In comparison, our learning-based approach can potentially generalize better to other datasets compared to SAD.

Getting an SNN-based model to not only mechanistically perform the VPR task, but also achieve broadly comparable performance to SAD and NetVLAD represents a substantial step forward. We note however that these existing techniques have been demonstrated on larger datasets, making further scalability (beyond the studies already presented here) one of the core foci of future work with the SNN-based approach.

C. Ablation Study: Varying number of places

In this section, we analyze the model performance (using our proposed weighted probability-based assignments) with a varying number of place labels and a lower number of epochs when increasing the number of places. This study covers the same section of the Nordland dataset which covers 400 place labels. Therefore, the model learning 25 place labels was repeated 16 times on consecutive sections (16 × 25 = 400),
the model with 50 place labels was repeated 8 times, and so on. For the evaluation points where 400 is not divisible by the number of places, at least two sections were evaluated with varying offsets.

Figure 6 shows that as the number of place labels increases, the model performance gracefully degrades when more than 100 places are learned. This is expected as the number of neurons assigned to a particular place label decreases with increasing number of places to encode, and the number of labels learnt by a neuron increases. Consequently, a single neuron outputs a higher number of spikes for different place labels, resulting in worse performance.

As the number of places grows, a network with an increased number of neurons will be needed to maintain a certain desired level of performance. Characterising this relationship, we found that to achieve an average AUC of 92+/−2%, the model approximately requires 100 output neurons for 50 places, 200 output neurons for 100 places, 450 output neurons for 150 places, and 800 output neurons for 200 places. However, the increase in the number of output neurons results in an approximately linear increase in inference time per query image (see our GitHub repository for figures). In Section VII, we discuss our future work on approaches that can enable the network to maintain its performance for large-scale VPR.

An additional post-processing approach that can improve model performance is sequence matching as in SeqSLAM [17]. The AUC of the model with 400 places is 39.9% (Figure 6) using the weighted probability-based assignment (single-image based). By using SeqSLAM with a sequence length of 5 and 10 frames, the AUC of this model increases to 88.3% and 98.4% respectively, showing that our model is amenable to sequence processing (see our GitHub repository for a figure). We note that sequence matching can thus be used to implicitly encode a larger number of places. In future works, we would like to provide image sequences as input to our model instead of using a post-processing step.

Future work will pursue several promising directions. Firstly, we will investigate scaling the model size up to arbitrarily large environments, evaluating in particular how training and compute requirements scale and the resulting effect on performance. We will also investigate approaches to improve the robustness of our method to significant viewpoint changes.

We are working with Intel to deploy this system on their new Loihi neuromorphic hardware. Such deployment has previously shown promise in terms of achieving high energy efficiencies with high throughput and low latencies [44]. A number of recent works on SNNs [35], [44] (which are loosely similar to ours and were deployed on Loihi) have reported inference times of less than 1 ms per query image, which is significantly lower than the inference time of our model on CPU (1 s) and GPU (0.2 s). The network upon which we base our system has been estimated to have less than 1 mW of power consumption on neuromorphic hardware [23], which is significantly lower than the power consumption of our model on GPU (81 W).

Loihi provides support to implement energy-efficient neuronal dynamics by supporting fine-grain parallelism, sparse event-based processing, and effective weight sharing [2]. It will be feasible to deploy our SNN model on Intel’s Kapoho Bay USB device which contains two Loihi chips and interfaces for sensor and event-based cameras [2]. Integrating the VPR capability into a full SNN-based SLAM system will assist in enabling the deployment of this system on low-power, low-cost robots with tight energy constraints.

Another avenue of future work includes optimization techniques for encoding mechanisms and structural synaptic plasticity mechanisms to improve learning speed, optimize computational resource allocation, and improve the encoding efficiency of place labels [22]. Finally, rather than converting intensity images to spike-trains, we are investigating the use of event cameras [29] that directly provide spikes as output, providing the potential for a further reduction in energy consumption and localization latency. Previous works have demonstrated improvement in energy consumption and computational latency in robotic applications [33], [34].

**VII. Discussions and Conclusions**

This research demonstrated that a two-layer SNN can be adapted for the place recognition task, through the introduction of a new weighted assignment scheme, and that it can achieve comparable performance to classical and state-of-the-art techniques on benchmark robotics datasets with significant degrees of appearance change.

Future work will pursue several promising directions. Firstly, we will investigate scaling the model size up to arbitrarily large environments, evaluating in particular how training and compute requirements scale and the resulting effect on performance. We will also investigate approaches to improve the robustness of our method to significant viewpoint changes.

We are working with Intel to deploy this system on their new Loihi neuromorphic hardware. Such deployment has previously shown promise in terms of achieving high energy efficiencies with high throughput and low latencies [44]. A number of recent works on SNNs [35], [44] (which are loosely similar to ours and were deployed on Loihi) have reported inference times of less than 1 ms per query image, which is significantly lower than the inference time of our model on CPU (1 s) and GPU (0.2 s). The network upon which we base our system has been estimated to have less than 1 mW of power consumption on neuromorphic hardware [23], which is significantly lower than the power consumption of our model on GPU (81 W).

Loihi provides support to implement energy-efficient neuronal dynamics by supporting fine-grain parallelism, sparse event-based processing, and effective weight sharing [2]. It will be feasible to deploy our SNN model on Intel’s Kapoho Bay USB device which contains two Loihi chips and interfaces for sensor and event-based cameras [2]. Integrating the VPR capability into a full SNN-based SLAM system will assist in enabling the deployment of this system on low-power, low-cost robots with tight energy constraints.

Another avenue of future work includes optimization techniques for encoding mechanisms and structural synaptic plasticity mechanisms to improve learning speed, optimize computational resource allocation, and improve the encoding efficiency of place labels [22]. Finally, rather than converting intensity images to spike-trains, we are investigating the use of event cameras [29] that directly provide spikes as output, providing the potential for a further reduction in energy consumption and localization latency. Previous works have demonstrated improvement in energy consumption and computational latency in robotic applications [33], [34].
