
Article:
Ozdemir, A. orcid.org/0000-0003-0014-4699, Scerri, M. orcid.org/0000-0001-5740-037X, Barron, A.B. orcid.org/0000-0002-8135-6628 et al. (4 more authors) (2022) EchoVPR: Echo state networks for visual place recognition. IEEE Robotics and Automation Letters, 7 (2). pp. 4520-4527. ISSN 2377-3766

https://doi.org/10.1109/lra.2022.3150505

EchoVPR: Echo State Networks for Visual Place Recognition

Anıl Özdemir[1,†], Mark Scerri[1,†], Andrew B. Barron[2], Andrew Philippides[3], Michael Mangan[1,*], Eleni Vasilaki[1,4,*], and Luca Manneschi[1,*]

***Abstract*—Recognising previously visited locations is an important, but unsolved, task in autonomous navigation. Current visual place recognition (VPR) benchmarks typically challenge models to recover the position of a query image (or images) from sequential datasets that include both spatial and temporal components. Recently, Echo State Network (ESN) varieties have proven particularly powerful at solving machine learning tasks that require spatio-temporal modelling. These networks are simple, yet powerful neural architectures that—exhibiting memory over multiple time-scales and non-linear high-dimensional representations—can discover temporal relations in the data while still maintaining linearity in the learning time. In this paper, we present a series of ESNs and analyse their applicability to the VPR problem. We report that the addition of ESNs to pre-processed convolutional neural networks led to a dramatic boost in performance in comparison to non-recurrent networks in five out of six standard benchmarks (GardensPoint, SPEDTest, ESSEX3IN1, Oxford RobotCar, and Nordland), demonstrating that ESNs are able to capture the temporal structure inherent in VPR problems. Moreover, we show that models that include ESNs can outperform class-leading VPR models which also exploit the sequential dynamics of the data. Finally, our results demonstrate that ESNs improve generalisation abilities, robustness, and accuracy further supporting their suitability to VPR applications.**

***Index Terms*—Vision-Based Navigation; Deep Learning for Visual Perception; Visual Learning**

I. Introduction

VISUAL Place Recognition (VPR) challenges algorithms to recognise previously visited places despite changes in appearance caused by illuminance, viewpoint, and weather conditions (for reviews see [1], [2], [3]). Unlike in many machine learning tasks, typical VPR benchmarks challenge models to learn image locations following a single route traversal, which are then compared with data during another route traversal. Thus, there are very few examples to learn from (typically only the images within a few metres of the correct location) making the task particularly challenging.

One approach is to recognise places based on matching single views using image processing methods to remove the variance between datasets. For instance, models have been developed that use different image descriptors to obtain meaningful image representations that are robust to visual change (e.g. DenseVLAD [4], NetVLAD [5], AMOSNet [6], SuperGlue [7], DELG [8], Patch-NetVLAD [9], and HEAPUtil [10]). While matching single images is successful in many scenarios, it can suffer from the effects of aliasing, individual image corruption, or sampling mismatches between datasets (e.g. it is challenging to ensure that images sampled along the same route precisely overlap).

Milford and Wyeth [11] were the first to demonstrate that such issues could be overcome by matching sequences of images using a global search to overcome individual image mismatches. This has led to a family of models that improve VPR performance by exploiting the temporal relationships inherent in images sampled along routes [11], [12], [13], [14], [15], [16], [17]. While achieving state-of-the-art performance on challenging real-world datasets (e.g. Oxford RobotCar [18], Extended CMU Seasons [19], Pittsburgh [20], Tokyo24/7 [4], Nordland [21], Mapillary Street Level Sequences [22]), such models often include an explicit encoding of non-visual information to limit the image search space, require a cache of input images for global searches, and can be computationally expensive in both learning and deployment phases. Neither of these features are desirable for autonomous robots that may have limited computational resources and external sensing capabilities.

Echo State Networks (ESN) [23] are a class of computationally efficient recurrent neural networks, ideally suited to addressing VPR problems without the need for additional support cues or input data caching (see Fig. 1). ESNs are a subset of reservoir computing models in which the reservoir neurons possess fixed, random and recurrent interconnections that sustain recent memories, i.e. *echoes* [24], with the practical benefit that only the output layer weights require training. ESNs thus act as a temporal kernel [25] over a variety of time-scales, creating a form of working memory dispensing of the need for input caching. ESNs have excelled when applied to problems that involve sequential data including dynamical system predictions [26], [27], robotic motion and navigation tasks [28], [29], [30]. Hence, they appear well-suited to the VPR problem but this hypothesis is yet to be tested. Recently

Manuscript received: September 9, 2021; Revised December 3, 2021; Accepted January 20, 2022.

This paper was recommended for publication by Editor Eric Marchand upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by Engineering and Physical Sciences Research Council under Grant EP/P006094/1, EP/S030964/1, EP/S009647/1 and EP/V006339/1, and by Templeton World Charity Foundation Grant № 0539.

[1]Anıl Özdemir, Mark Scerri, Michael Mangan, Eleni Vasilaki, and Luca Manneschi are with Department of Computer Science, The University of Sheffield, UK e.vasilaki@sheffield.ac.uk

[2]Andrew B. Barron is with Department of Biological Sciences, Macquarie University, Australia

[3]Andrew Philippides is with School of Engineering and Informatics, University of Sussex, UK

[4]Eleni Vasilaki is Institute for Neuroinformatics, University of Zurich and ETH Zurich, Switzerland

[†]These authors have equally contributed to the work.

[*]Joint senior authors.

ESNs have been augmented by novel techniques e.g., [31], [32]. Among these, in the present work we focus on SPARCE, which stands for Sparse Reservoir Computing.

The contributions of this work are:

- a novel approach that combines state-of-the-art image descriptors augmented with ESNs to exploit temporal dynamics of image sequences;
- validation of the proposed models in four evaluation datasets that present different input variances (e.g. indoor/outdoor, day/night, and viewpoint);
- benchmarking against contemporary single- and sequential-view based models in challenging, long-range VPR datasets sampled from moving vehicles across illumination and seasonal changes;
- assessment of the models' capacity to recognise routes from a random start-point (akin to kidnapped robot problem).

II. Methods

A. Problem Formulation

VPR algorithms are provided with a sequence of places (in the form of images) sampled along a route. They are then asked to correctly match (within an acceptable threshold) the places by the image key-frames along the same route at a different time. The input data is composed of videos where the network has to correctly infer the location, i.e. the image key-frame that is processed at the considered time. In all the tasks there are at least two sequences of images, one used as a training set (i.e. reference) and the other used as a test set (i.e. query), acquired by visiting the same locations and following the same path twice. Even though there is a one-to-one mapping between training and test samples, the latter is acquired by visiting the locations at different times, leading to differences in visual appearances, such as seasonal or illuminance as well as viewpoint changes. Perfect matching is not always possible, hence, there can be a tolerance term that allows close matches to be accepted. A match is considered successful, if $||\text{reference} - \text{query}|| \leq \text{tolerance}$.

In our specific implementation, we consider supervised learning with the ESNs as a predictor, hence, forming a classification problem. The number of read-out nodes is equal to the number of places, and therefore, specific to the given dataset. The read-out nodes (the final and only learnable layer) output a probability distribution, $\mathcal{P}_{\text{query}}$, for each given query image. The prediction (i.e. key-frame of the query) is the number of the read-out node, i.e. $\arg\max \mathcal{P}_{\text{query}}$.

B. Standard Echo State Networks

An ESN is a reservoir of recurrently connected nodes, whose temporal dynamics $\mathbf{x}(t)$ evolves following [23]:

$$\mathbf{x}(t+\delta t) = (1-\alpha)\mathbf{x}(t) + \alpha f\left(\mathbf{h}(t)\right), \tag{1}$$

$$\mathbf{h}(t) = \gamma \mathbf{W}_{\text{in}}\mathbf{s}(t) + \rho \mathbf{W}\mathbf{x}(t), \tag{2}$$

where α is the leakage term and defines the rate of integration of information, f is a non-linear activation function (usually $\tanh$), $\mathbf{s}(t)$ is the input signal, $\mathbf{W}_{\text{in}}$ is the input connectivity matrix, which is commonly drawn from a random Gaussian distribution, and γ is a multiplicative factor of the external signal. The recurrent connectivity $\mathbf{W}$ is a sparse, random and fixed matrix whose eigenvalues are constrained inside the unit circle of the imaginary plane, with a hyper-parameter ρ (usually in the range of $[0, 1]$) set to further control the spectral radius. Learning occurs via minimisation of a cost function. Only the read-out weights $\mathbf{W}_{\text{out}}$ that connect the reservoir neurons $\mathbf{x}$ to the output change. Optimisation of $\mathbf{W}_{\text{out}}$ can be accomplished through different techniques such as ridge regression or iterative gradient descent methods [33]. In our work, we use exclusively gradient decent methods.

C. Sparse Reservoir Computing

The definition of sparse representations through the SPARCE model [31] can enhance the capacity of the reservoir to learn associations by introducing specialised neurons through the definition of learnable thresholds. Considering the representation $\mathbf{x}$ from which the read-out is defined, as in Eq. (1), SPARCE consists of the following normalisation operation:

$$x'_i = \text{sign}\left(x_i\right) \text{ReLU}\left(|x_i| - \theta_i\right) \tag{3}$$

$$\theta_i = P_n\left(|\mathbf{x}_i|\right) + \bar{\theta}_i \tag{4}$$

where i is the i-th dimension, sign is the sign function and ReLU is the rectified linear unit. Of course, the new read-out is defined from x'_i, that is after the transformation given in Eq. (3) and (4), which leaves the dynamics of the system unaltered and can be easily applied to any reservoir representation. The threshold θ_i is composed of two factors: $P_n\left(|\mathbf{x}_i|\right)$, i.e. the n-th percentile of $\mathbf{x}_i$, which stands for the distribution of activities of dimension i after the presentation of a number of samples with sufficient statistics, and a learnable part $\bar{\theta}_i$, which is adapted through gradient descent and is initialised to arbitrarily small values at the beginning of training. The percentile n can be considered as an additional *interpretable* hyper-parameter that controls the sparsity level of the network at the start of the training phase[1].

D. Image Pre-processing

Convolutional neural networks (CNN) are the best performing architectures for processing images and discovering high-level features from visual data. However, they are static and lack temporal dynamics. Thus, after a pre-processing module composed of NetVLAD [5], a pre-trained CNN, we adopted a system composed by ESNs. Considering that the reservoir computing paradigm is more effective when the reservoir expands the dimensionality of its corresponding input, we first decreased the dimensionality of NetVLAD output (original dimension is 4096) by training a feedforward network composed of one hidden layer (with 500 nodes) on the considered classification task. The hidden layer representation is then considered as the input to the reservoir computing system. With NV we denote the full model consisting of NetVLAD, a

[1]For different methodologies to estimate the percentile operation, see [31]

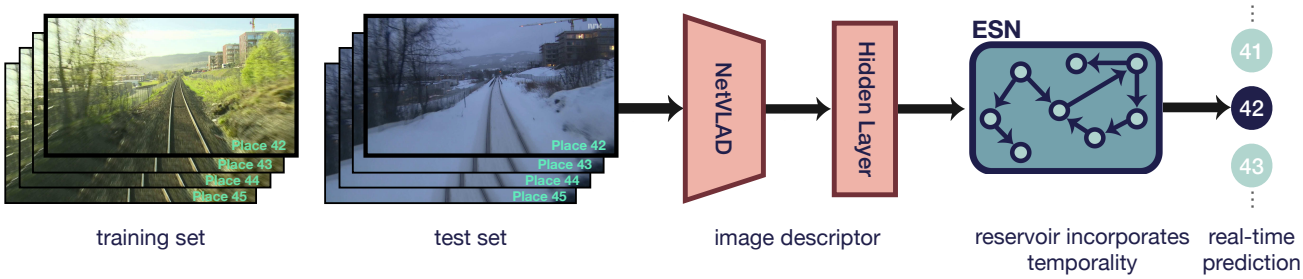

Fig. 1: **An illustration of EchoVPR framework.** Echo State Networks (ESN) incorporate temporality while still maintaining real-time prediction capability, which is a key feature for a robotic system in real-world applications. Given an input image at a time (from snowy Nordland [21] in this example), an image descriptor (class-leading NetVLAD [5] via a hidden layer, see Section II-D) provides a meaningful representation to the ESN to update the *fixed* reservoir.

hidden layer and an output layer. When NV is combined with another system, e.g., NV-ESN, it means that we have used the trained hidden layer as an input representation to the ESN.

E. Proposed Architecture

The reservoir is then trained to distinguish the different locations, which are processed successively in the natural order of acquisition by the overall architecture. The two reservoir computing models we study are summarised below[2]:

- **Echo State Network (`ESN`)**, where learning happens on the output weights only. The critical hyper-parameters of the system for the cases studied are α, γ, η (leakage term, input factor, learning rate). The overall architecture that exploits this reservoir will be named `NV-ESN` in the rest of the work.
- **Echo State Network with SPARCE (`SPARCE-ESN`)**, where thresholds are applied to the reservoir following Eq. (3) and learning occurs on $\bar{\theta}$ and $\mathbf{W}_{\text{out}}$. The hyper-parameters are the same as the standard `ESN` with the addition of the starting percentile P_n of Eq. (4). For this case, the overall architecture will be named `NV-SPARCE-ESN`.

F. Error Functions

Learning of $\mathbf{W}_{\text{out}}$ and $\bar{\theta}$ is accomplished through mini-batches and by minimisation of softmax cross-entropy loss via gradient descent:

$$E = \sum_{j}^{\mathrm{N_{batch}}} \sum_{i} y_{ij}^{\text{target}} \log \left(\frac{\exp\left(y_{ij}\right)}{\sum_i \exp(y_{ij})} \right), \tag{5}$$

where $\mathrm{N_{batch}}$ is the minibatch size, y the output of the neural network, y^{target} the target output, and the indices i and j correspond to the sample number and to the output node considered. The models are trained for up to 60 epochs, i.e. each training image is seen 60 times.

[2]The supplementary material and source-code for the ESN implementations can be found in https://anilozdemir.github.io/EchoVPR/

For larger datasets, we used the sigmoid cross-entropy loss as the error function, which led to better performance:

$$E = \sum_{j}^{\mathrm{N_{batch}}} \sum_{i} y_{ij}^{target} \log \left[\sigma(y_{ij})\right] + (1 - y_{ij}^{target}) \log \left[1 - \sigma(y_{ij})\right], \tag{6}$$

where σ is the sigmoid function and the remaining terms correspond to the ones of Eq. (5). A description of the datasets used is provided in Section III-A.

III. Experiments

A. Datasets And Metrics

To allow benchmarking of different models across datasets, the VPR community is converging upon a standardised methodology [34], [35], [36]. We therefore evaluate model performance using the recommended metrics: prediction *accuracy*, *recall@n* and precision-recall *area-under-curve* (AUC).

Initial analysis was performed on four datasets that challenge models in a variety of conditions:

- **GardensPoint** [37] consists of 200 indoor, outdoor and natural environments with both viewpoint and conditional changes throughout the dataset. A tolerance of 2 is accepted.
- **ESSEX3IN1** [38] consists of 210 images taken at the university campus and surroundings, focusing on perceptual aliasing and confusing places. There is no tolerance for this dataset.
- **SPEDTest** [39] consists of 607 low-quality but high-depth images collected from CCTV cameras around the World; it includes environmental changes including variations in weather, seasonal, and illumination conditions. There is no tolerance for this dataset.
- **Corridor** [12] consists of 111 images of an indoor environment - three traverses of the same corridor.

Models were then evaluated in two larger datasets captured from moving vehicles that are more representative of the data that would be presented to an autonomous robot:

- **Nordland** [9] comprises of images taken at train traversals in four different seasons in Norway; the viewpoint angle is fixed although there is a high weather, seasonal and illumination variability. Each traversal consists of

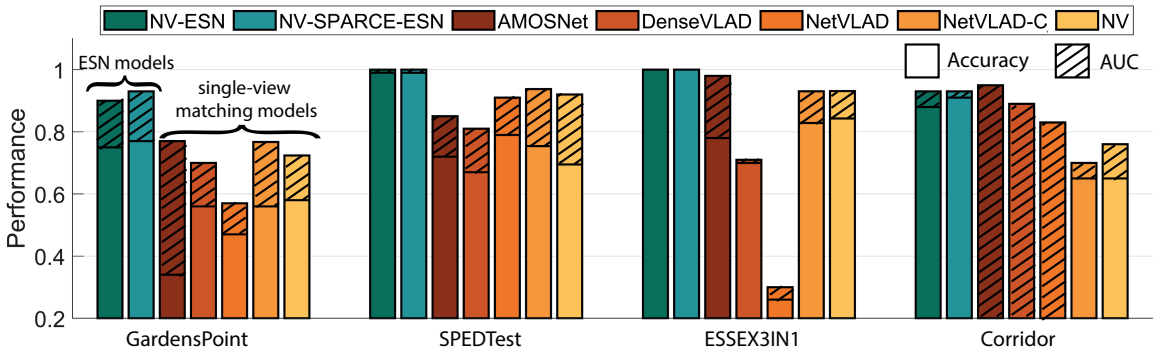

Fig. 2: **ESN performance compared with single view matching algoirthms in evaluation datasets.** The utilisation of reservoir computing models permits the temporal dynamics of the problem to be captured and improves the performance of CNNs. `NV-ESN` and `NV-SPARCE-ESN` are shown in blue-green colours, while the performance of static neural networks is reported in red-yellow colours. The performance of AMOSNet, DenseVLAD and NetVLAD were taken from [34], where image matching was achieved by computing distances among the representation. NetVLAD-C and NV correspond to models in which a simple read-out or a hidden layer with a read-out were trained from the representation of the convolutional network respectively. This was achieved through the minimisation of Eq. (5) on the specific task considered, similar to the approach used for ESNs. The bar plots for our method shows average performance over 20 trials.

27592 images. A tolerance of 10 is acceptable—the same as the models [16], [9], [17] we compare against (see Section IV-E for more details).

- **Oxford RobotCar** [40] comprises of images taken from a car travelling a fixed 10 km route around Oxford, UK at different times of the day across seasons. Each traversal consists of roughly 30000 images.

To allow direct comparison of ESNs with different published results, we also consider a subset of the **Nordland** dataset composed of 1000 images as in [16], and then the full dataset as in [9]. For the **Oxford RobotCar** dataset [40], we used 4599 and 4550 images captured during the day (2015-03-17-11-08-44) and at night (2014-12-16-18-44-24) sampled at 2m apart for training and testing respectively, with a tolerance of 20m, in a similar manner to [17].

B. Benchmarking Process

We compared `NV-ESN` and `NV-SPARCE-ESN` (see Section II-E) on the smaller datasets (GardensPoint, SPEDTest, ESSEX3IN1) to the following models, taken from the literature: AMOSNet [6], DenseVLAD [4] and NetVLAD [5]. We also used our own NetVLAD variants, NetVLAD-C, which consists of NetVLAD with an output layer trained as a classifier, and NV (see Section II-D).

For the Nordland dataset, we compare our methods with NetVLAD [5], SuperGlue [7], DELG global [8], DELG local [8] and Patch-NetVLAD [9]. The latter uses NetVLAD in combination with local features (Patch). We were able to combine our ESNs variants with local features similar to Patch-NetVLAD. In particular, after classification of the top 100 locations performed by `NV-ESN` or `NV-SPARCE-ESN`, we used such local features to select the final classified location. For simplicity, we named the resulting models `PatchL-ESN` and `PatchL-SPARCE-ESN` for the case with or without SPARCE respectively, where `L` stands for "Light". The "Light" model refers to the variant Single-Spatial-Patch-NetVLAD as defined in [9] which employs a simple spatial verification method applied on a single patch of size 5 and 512 PCA dimensions. In contrast the results that are reported from Patch-NetVLAD [9] utilises the variant Multi-RANSAC-Patch-NetVLAD which employs a more complex spatial scoring method applied on a fusion of multiple patches of sizes (2, 5 and 8) and 4096 PCA dimensions. This selection process among the top 100 images is analogous to the one in the original paper [9].

Again on Nordland, we compared `NV-ESN` to sequential models. More specifically, we considered three NetVLAD variants (NetVLAD+Smoothing, NetVLAD+Delta and Netvlad+SeqMatch), SeqNet(S_5) and HVPR(S_5 to S_1). For the description of all these models see [17]. SeqNet, in particular, is a recently formulated sequential model where authors used the images of the Nordland dataset captured during summer and winter as training set, and then tested the model on previously unseen locations during Spring and Fall. In [17], the generalisation on unseen locations is possible because of the computation of distances among images and of the minimisation of a triplet loss function. Considering that we are training a classifier, where each image corresponds to a different output node similarly to [16], it is difficult to make a direct comparison. Instead, we formulated a methodology inspired by the one adopted for SeqNet [17]. We selected single images or pairs of consecutive images randomly from the summer and winter datasets and excluded them from the training. We used these images for validation and images captured at the same locations, but during Spring and Fall, for testing. In this way, our models are tested on images for which training was absent.

For the reduced version of the Nordland dataset, we compared our methods `NV-ESN` and `NV-SPARCE-ESN` to FlyNet+CANN and FlyNet+RNN. In that case the comparison was straightforward as all networks are classifiers presuming

each place is a class.

We also compared NV-ESN, PatchL-ESN, NV-SPARCE-ESN and PatchL-SPARCE-ESN on the Oxford RobotCar dataset with NetVLAD and Patch-NetVLAD, which were not available from the literature. For these two models, we produced results using the source-code provided by the authors.

C. Hyper-parameter Tuning On ESNs

The lack of a validation set for the considered tasks makes hyper-parameter selection challenging. This difficulty is increased by the small number of samples in the training set (i.e. one sample per place) and by the major statistical differences between the training and test data. In particular, the seasonal differences in the acquisition of reference and query data lead to the possible presence or absence of snow and shifts in colour intensity. In our preliminary experiments, different hyper-parameters often reached perfect accuracy (i.e. 100%) on the training set and had reduced performance on the test set. For this reason, a validation test was required.

We believe that there is a lack of clarity in previous research works regarding the definition of a methodology to overcome the problem of hyper-parameter selection. Wetherefore tuned the hyper-parameters of the reservoir by using a small percentage (i.e. 10%) of samples of the test set as validation. In other words, while the read-out was always optimised from training samples, hyper-parameters were optimised through grid search over the performance achieved on 10% of the test data. We did not estimate performance on validation data when computing scores on the test dataset. Being aware of the limitations of this methodology, we show how it is possible to use the test set of one task as validation for another task with little performance loss, demonstrating that the model performs well if the hyper-parameters were selected to be robust to non-excessive statistical changes (see Section IV-B).

IV. Results

A. Assessing ESNs On Visual Place Recognition

The performance of the NV-ESN and NV-SPARCE-ESN models were first evaluated in four datasets (GardensPoint, SPEDTest, ESSEX3IN1 and Corridor). Fig. 2 shows that both ESN variants outperform single-view matching models (including NetVLAD with read-out and hidden layers) in all four conditions (with the exception of AMOSNet in Corridor). The NV-ESN achieves mean accuracy scores of 0.75, 0.99, 1.0 and 0.88 and mean AUC scores of 0.9, 1.0, 1.0 and 0.93. The addition of the NV-SPARCE layer provides additional improvement with accuracy scores of 0.77, 0.99, 1.0, and 0.91 and mean AUC scores of 0.93, 1.0, 1.0 and 0.93. The ESNs used $N = 1000$ reservoir neurons.

B. Robustness Of Hyper-parameters

We also analysed the sensitivity of the ESNs with respect to hyper-parameter selection. Fig. 3 shows accuracy scores for hyper-parameters tuned by training the models on GardensPoint and maintaining them when training in SPEDTest, ESSEX3IN1 and Corridor. The reason we chose the hyper-parameters for GardensPoint is that generalisation is more likely to occur when the baseline task is more complex than the new tasks to which it is applied. Indeed, richer and more difficult datasets can lead neural networks to discover high-level features that are transferable to simpler datasets, while the contrary is difficult. Fig. 3 demonstrates how, even with sub-optimal hyper-parameters, the introduction of ESNs leads to higher performance in comparison to single-view matching models: NetVLAD-C and NV. Moreover, the performance remains above 86% for both accuracy and AUC compared to the virtually perfect scores achieved when hyper-parameters were tuned using the same dataset (see Fig 2).

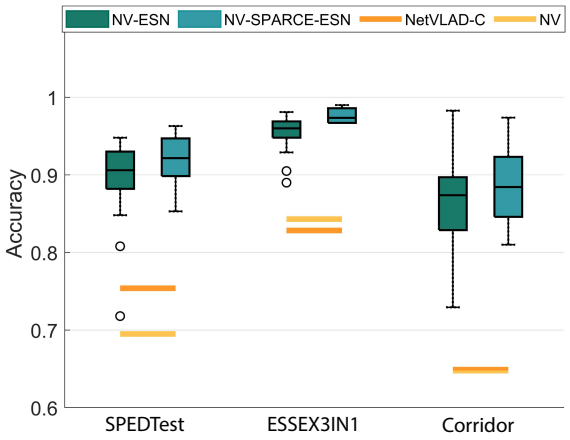

Fig. 3: **Robustness of hyper-parameters.** Performance is maintained despite using the hyper-parameters optimised for a different dataset (GardensPoint). The two variants of ESN are well above the accuracy achieved by NetVLAD-C and NV (horizontal lines). The box plots represent the distribution of 20 trials.

TABLE I: **ESNs outperform state-of-the-art single view matching models in the challenging Nordland and Oxford RobotCar benchmarks.** The larger datasets pose a greater challenge for algorithms with an order of magnitude more images and extreme variance between training and test sets (seasonal and day/night). Best results are found for PatchL-ESN in Nordland, and PatchL-SPARCE-ESN in Oxford RobotCar.

	Nordland Summer vs Winter			Oxford RobotCar Day vs Night		
Method	R@1	R@5	R@10	R@1	R@5	R@10
NetVLAD [5]	13.0	20.6	25.0	31.2	48.6	58.3
SuperGlue [7]	29.1	33.4	35.0	–	–	–
DELG global [8]	23.4	35.4	41.7	–	–	–
DELG local [8]	60.1	63.5	64.6	–	–	–
Patch-NetVLAD [9]	44.9	50.2	52.2	80.7	84.8	86.4
NV-ESN	47.5	61.0	66.4	44.0	58.1	65.5
NV-SPARCE-ESN	46.7	60.3	65.9	80.4	87.9	91.5
PatchL-ESN	**66.4**	**76.0**	**78.7**	75.9	85.2	87.7
PatchL-SPARCE-ESN	66.3	75.7	78.6	**88.8**	**97.0**	**98.2**

C. Performance On Larger Datasets

To assess the scalability of the ESNs, and to compare their performance to state-of-the-art models, performance was analysed in the full Nordland and Oxford RobotCar datasets

as in [17]. The methodology adopted is the same as in Section IV, but the number of reservoir nodes in the ESN is increased to 8000 and 6000 for the Nordland and the Oxford RobotCar datasets, respectively. Results presented in Table I, show that best performance was achieved for PatchL-ESN in the Nordland dataset, when compared with state-of-the-art single view matching models. We would like to note that, in our models we have used exclusively the "light" version of Patch-NetVLAD for computational efficiency, while the performance reported in Table I of the original Patch-NetVLAD exploited its most complex version. Considering the results for Nordland (summer vs winter) and Oxford RobotCar (day vs night), it is evident that the temporal features captured by the ESNs cause it to reach competitive performance. `NV-ESN` reports recall@n that are higher than the original Patch-NetVLAD for the Nordland dataset, while `NV-SPARCE-ESN` is surprisingly successful on the Oxford RobotCar dataset, with performance that is inferior only to its more complex variant `PatchL-SPARCE-ESN` and comparable to Patch-NetVLAD. The utilisation of the local features from [9] further improve these results, and the model `PatchL+SPARCE-ESN` is the best performing network overall. Indeed, `PatchL+SPARCE-ESN` reports the highest score for Oxford RobotCar and it is only slightly worse than `PatchL-ESN` for Nordland. It is likely that these results might improve by using the complete variant of Patch-NetVLAD. Here, we chose to keep the model as simple as possible, given that it still outperforms the other techniques. Thus, ESNs constitute an addition to other neural architectures that can lead to considerable performance improvements and to state-of-the-art results.

D. Robustness With Respect To Start-Point

We investigate if the proposed models satisfy two conditions that are necessary for a potential implementation on a robot. First, the model should maintain high performance if we start from different locations of the input sequence. Second, the amount of temporal information that needs to be processed before the system can reach satisfactory results should be low. Fig. 4 shows the recall@n of an ESN as the number of images being processed increases from the starting location. The recall@n are averaged across different possible initial locations and computed on the test set of the Nordland dataset. The average performance increases quickly and steadily, showing that the model needs to process in the order of 50 images before the trends converge. The fact that the performance of Fig. 4 is obtained by averaging from various starting points further demonstrates the robustness of the model with respect to variation of the initial location.

E. Comparing ESN With Sequential VPR Models

In this section, we benchmark the performance of ESNs against sequence matching VPR models. The majority of sequential models for VPR benchmarks adopt different learning and testing methodologies, and it is consequently difficult to make direct comparisons to previously published results. However, it is possible to directly compare our networks with

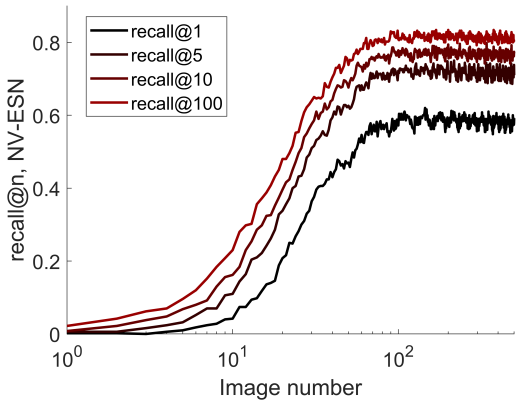

Fig. 4: **Robustness with respect to start-point.** Performance of `NV-ESN` as the image number from a starting location increases. The trends are averaged across 500 different starting locations of the Winter Nordland datasets.

with two models recently reported to achieve great performance [16] in a subset of the Nordland dataset [21] composed by 1000 images. Both previous models use a bio-inspired feed-forward neural network (FlyNet) to encode visual information and either a recurrent neural network (RNN) or a continuous attractor neural network (CANN) to introduce temporality. Fig. 5 shows accuracy scores of 0.72 and 0.92 for the standard `NV-ESN` and `NV-SPARCE-ESN` respectively. For the AUC test, `NV-ESN` achieves scores of 0.95, with `NV-SPARCE-ESN` improving results to 0.98. This compares favourably to both static view matching models (e.g. NetVLAD+HL) which score 0.24, and sequential models which score 0.21 (FlyNet+RNN) and 0.91 (FlyNet+CANN). While it is not possible to directly and fairly compare with models such as SeqNet [17] that use distances between images, we have devised a methodology that allows us to provide scores of our methods in classifying previously unseen data.

The performance obtained for the larger version of the Nordland dataset are shown in Table II (*summer-winter vs spring-fall*). The two results reported for each recall@n correspond to the case where single, or consecutive image pairs, were randomly selected for testing respectively (see Section III-B for more details). In the latter case, because the connections to adjacent pairs of output neurons are not trained, the performance degrades. However, even in a learning and testing paradigm not completely suited to our model, the proposed networks are able to generalise. This additional comparison between ESNs and previously published results (Table II) confirms that the former can achieve high performance in relation to sequential techniques. Moreover, the memory of ESNs is solely reliant on the internal dynamics, while the vast majority of other sequential models rely on a direct comparison among sequences, which need to be stored in memory.

V. Conclusions

Here we have demonstrated the viability of ESNs as a solution to the VPR problem. Both ESN variants achieve

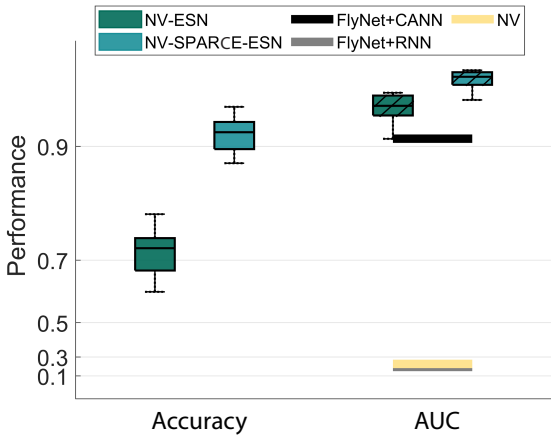

Fig. 5: **Direct comparison to sequential classification models in the Nordland dataset.** The `NV-ESN` model and in particular `NV-SPARCE-ESN`, outperform FlyNet+RNN and FlyNet+CANN, taken from [16] and NV when compared with reported AUC scores. The horizontal lines report the performance of FlyNet and NV models. The box plots represent the distribution of 20 trials.

TABLE II: **Indicative comparison of ESNs to published results with sequential models in the Nordland dataset.** We note that the NV-ESN cannot be evaluated in the same way as the other models so a direct comparison is not appropriate.

	Nordland Summer-Winter vs Spring-Fall		
Method	R@1	R@5	R@20
NetVLAD+Smoothing [17]	44.0	59.0	72.0
NetVLAD+Delta [17]	56.0	70.0	80.0
NetVLAD+SeqMatch [17]	61.0	71.0	78.0
SeqNet (S_5) [17]	79.0	90.0	94.0
HVPR (S_5 to S_1) [17]	79.0	89.0	94.0
NV-ESN	82.7/77.6	91.2/85.9	95.1/90.9

comparable or better performance than single-view matching models in four evaluation datasets. On larger datasets such as full Nordland and Oxford RobotCar, the two variants, combined with local features, were the best models. With respect to sequential models, where a fair comparison was permitted, two ESN variants achieved performance above/equal to the class-leading results on the reduced Nordland dataset. While performance is comparable, we note that FlyNet [16] has many fewer parameters. However, the ESN does not require images to be cached during multiple comparisons or any external cues. A comparison with sequential models is, in general, difficult due to their use of distances that allow unseen images to be compared. In this sense, these sequential models have an advantage at the cost that caching images is required in order to recognise a new place.

The addition of SPARCE to the standard ESN improved performance considerably in almost all cases, showing how the introduction of sparse representations can efficiently help the classification process. An exception was the full Nordland dataset, where we did not find any advantage by adding this technique, and the reasons are worthy of further investigation.

When it comes to ESN advances, an intriguing future course of action is to take inspiration from invertebrate mini-brains. An interesting point being that they possess analogous structural motifs of both deep and shallow ESNs. A simple example is the insect mushroom body. This is considered the cognitive centre of the insect brain [41] and is necessary for learning relationships sequences and patterns in honey bees [41], [42]. Structurally the mushroom body is a three-layer network with a compact input layer, an expanded middle layer of inter-neurons called Kenyon cells (KC), and a small layer of output neurons [43]. The connections between the KC and output neurons are plastic and modified by learning [44], and there are chemical and electrical synapses between the KC [45]. These features are analogous to the recurrent connections in the reservoir layer of an ESN, and it has been hypothesised [31] that these recurrent connections in the KC layer could contribute to the reverberant activity of the mushroom body that supports forms of memory [46]. Further, the technique SPARCE is inspired by the low activity exhibited by the KC. Given the similar structures of ESN and mushroom body, insights gained from neurobiology could help shape the future ESN investigations and in turn, analysis of the optimal structure for VPR could shed light on the function of different brain areas. For instance, in [32], hierarchical structures with fast and slow information processing times are investigated, reminiscent of neural processing observed in insect brains. See [47] (supplementary materials) for a preliminary study of such networks.

ESN efficiently exploit information in sequences but in practice it is also desirable that places are recognised from a single input image allowing robotics to truly solve the kidnapped robot problem. However, in the cases where such methods fail, traversing portions of a familiar path can help to disambiguate input. In this respect, our models require "viewing" of approximately 50 images (for the Nordland set) in order to start correctly identifying their location.

ESNs provide a means to exploit such temporal dynamics using only visual data but more powerful variants require tuning of a large number of parameters which may not be possible when only a small amount of training examples are provided. Other methods [11], [16] have focused on low-parameter models but often require additional cues such as velocity to focus the image search. Ensemble methods [48], [49] that combine these features are emerging that may provide the best of both worlds.

REFERENCES

[1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2015.

[2] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19 516–19 547, 2021.

[3] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, vol. 113, p. 107760, 2021.

[4] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1808–1817.

[5] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[6] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep learning features at scale for visual place recognition," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 3223–3230.

[7] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4938–4947.

[8] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," in *European Conference on Computer Vision*. Springer, 2020, pp. 726–743.

[9] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 141–14 152.

[10] N. V. Keetha, M. Milford, and S. Garg, "A hierarchical dual model of environment-and place-specific utility for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6969–6976, 2021.

[11] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 1643–1649.

[12] M. J. Milford, "Vision-based place recognition: how low can you go?" *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 766–789, 2013.

[13] P. Hansen and B. Browning, "Visual place recognition using HMM sequence matching," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4549–4555.

[14] E. Kagioulis, A. Philippides, P. Graham, J. C. Knight, and T. Nowotny, "Insect inspired view based navigation exploiting temporal information," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2020, pp. 204–216.

[15] L. Zhu, M. Mangan, and B. Webb, "Spatio-temporal memory for navigation in a mushroom body model," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2020, pp. 415–426.

[16] M. Chancán, L. Hernandez-Nunez, A. Narendra, A. B. Barron, and M. Milford, "A hybrid compact neural architecture for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 993–1000, 2020.

[17] S. Garg and M. Milford, "SeqNet: Learning descriptors for sequence-based hierarchical place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4305–4312, 2021.

[18] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford Robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[19] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, *et al.*, "Benchmarking 6DOF outdoor visual localization in changing conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[20] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 883–890.

[21] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons," in *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation*. Citeseer, 2013, p. 2013.

[22] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, and J. Civera, "Mapillary street-level sequences: A dataset for lifelong place recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2626–2635.

[23] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of Echo State Networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.

[24] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[25] M. Hermans and B. Schrauwen, "Recurrent kernel machines: Computing with infinite echo state networks," *Neural Computation*, vol. 24, no. 1, pp. 104–133, 2012.

[26] D. Li, M. Han, and J. Wang, "Chaotic time series prediction based on a novel robust Echo State Network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 787–799, 2012.

[27] A. Deihimi and H. Showkati, "Application of Echo State Networks in short-term electric load forecasting," *Energy*, vol. 39, no. 1, pp. 327–340, 2012.

[28] P. G. Plöger, A. Arghir, T. Günther, and R. Hosseiny, "Echo State Networks for mobile robot modeling and control," in *Robot Soccer World Cup*. Springer, 2003, pp. 157–168.

[29] K. Ishu, T. van Der Zant, V. Becanovic, and P. Ploger, "Identification of motion with Echo State Network," in *MTS/IEEE Techno-Ocean*, vol. 3, 2004, pp. 1205–1210.

[30] C. Hartland and N. Bredeche, "Using Echo State Networks for robot navigation behavior acquisition," in *IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 201–206.

[31] L. Manneschi, A. C. Lin, and E. Vasilaki, "SpaRCe: Improved Learning of Reservoir Computing Systems through Sparse Representations," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[32] L. Manneschi, M. O. Ellis, G. Gigante, A. C. Lin, P. Del Giudice, and E. Vasilaki, "Exploiting Multiple Timescales In Hierarchical Echo State Networks," *Frontiers in Applied Mathematics and Statistics*, 2021.

[33] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 659–686.

[34] M. Zaffar, S. Garg, M. Milford, J. Kooij, D. Flynn, K. McDonald-Maier, and S. Ehsan, "VPR-Bench: Open-Source Visual Place Recognition Evaluation Framework with Quantifiable Viewpoint and Appearance Change," *International Journal of Computer Vision*, pp. 1–39, 2021.

[35] S. Garg, T. Fischer, and M. Milford, "Where is your place, visual place recognition?" *arXiv preprint arXiv:2103.06443*, 2021.

[36] C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla, *et al.*, "Long-term visual localization revisited," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[37] A. Glover, "Day and night, left and right," 2014. [Online]. Available: https://doi.org/10.5281/zenodo.4590133

[38] M. Zaffar, S. Ehsan, M. Milford, and K. D. McDonald-Maier, "Memorable maps: A framework for re-defining places in visual place recognition," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

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

[40] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[41] R. Menzel and M. Giurfa, "Cognitive architecture of a mini-brain: the honeybee," *Trends in Cognitive Sciences*, vol. 5, no. 2, pp. 62–71, 2001.

[42] A. J. Cope, E. Vasilaki, D. Minors, C. Sabo, J. A. Marshall, and A. B. Barron, "Abstract concept learning in a simple neural network inspired by the insect brain," *PLoS Computational Biology*, vol. 14, no. 9, p. e1006435, 2018.

[43] S. E. Fahrbach, "Structure of the mushroom bodies of the insect brain," *Annual Review of Entomology*, vol. 51, pp. 209–232, 2006.

[44] B. Gerber, H. Tanimoto, and M. Heisenberg, "An engram found? evaluating the evidence from fruit flies," *Current Opinion in Neurobiology*, vol. 14, no. 6, pp. 737–744, 2004.

[45] Z. Zheng, J. S. Lauritzen, E. Perlman, C. G. Robinson, M. Nichols, D. Milkie, O. Torrens, J. Price, C. B. Fisher, N. Sharifi, *et al.*, "A complete electron microscopy volume of the brain of adult drosophila melanogaster," *Cell*, vol. 174, no. 3, pp. 730–743, 2018.

[46] P. Cognigni, J. Felsenberg, and S. Waddell, "Do the right thing: neural network mechanisms of memory formation, expression and update in drosophila," *Current Opinion in Neurobiology*, vol. 49, pp. 51–58, 2018.

[47] A. Ozdemir, M. Scerri, A. B. Barron, A. Philippides, M. Mangan, E. Vasilaki, and L. Manneschi, "EchoVPR: Echo State Networks for Visual Place Recognition," *arXiv preprint arXiv:2110.05572*, 2021.

[48] S. Hausler, A. Jacobson, and M. Milford, "Multi-process fusion: Visual place recognition using multiple image processing methods," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1924–1931, 2019.

[49] T. Fischer and M. Milford, "Event-based visual place recognition with ensembles of temporal windows," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6924–6931, 2020.

Supplementary Materials for EchoVPR: Echo State Networks for Visual Place Recognition

Anıl Özdemir[1,†], Mark Scerri[1,†], Andrew B. Barron[2], Andrew Philippides[3],
Michael Mangan[1,*], Eleni Vasilaki[1,4,*], and Luca Manneschi[1,*]

Section I introduces hierarchical ESNs for the VPR problem. Section II presents preliminary analysis of hierarchical ESNs for VPR in the evaluation study. Section III contains details of the hyper-parameters used in the main paper.

I. Hierarchical ESNs and SPARCE

Recent works have started to analyse the benefits of reservoir computing systems composed of multiple ESNs. In these architectures, ESNs are connected hierarchically and are tuned differently to exhibit diverse dynamical properties. For instance, the values of the leakage term $\alpha^{(k)}$, where k is the reservoir number, can vary for different networks, allowing them to regulate the time-scales at which diverse reservoirs operate. As a result, the overall system can be characterised by a wider range of time constants that has richer dynamics and improved memory abilities. Following the architecture in Fig. 1(b), the equations that describe a system of hierarchically connected reservoirs can be defined by:

$$\mathbf{x}^{(k)}(t+\delta t) = (1-\alpha^{(k)})\mathbf{x}^{(k)} + \alpha^{(k)} f\left(\mathbf{h}^{(k)}(t)\right), \tag{1}$$

$$\mathbf{h}^{(k)}(t) = \sum_{l}^{\mathrm{N_{ESN}}} \rho^{(kl)} \mathbf{W}^{(kl)} \mathbf{x}^{(l)}(t), \tag{2}$$

where $\alpha^{(k)}$ is the leakage term and defines the rate of integration of information, f is a non-linear activation function (usually $\tanh$), $\mathbf{s}(t)$ is the input signal. The recurrent connectivity $\mathbf{W}$ is a sparse, random and fixed matrix whose eigenvalues are constrained inside the unit circle of the imaginary plane, with a hyper-parameter $\rho^{(kl)}$ (usually in the range of $[0, 1]$) set to further control the spectral radius.

In the hierarchical structure of Fig. 1(b), $\mathbf{W}^{(kl)} \neq 0$ if $k = l$ or $k = l + 1$. In detail, $\mathbf{W}^{(kk)}$ indicates the recurrent connectivity of reservoir k and needs to have a spectral radius smaller than one, while $\mathbf{W}^{(kl)}$, where $k = l + 1$ is the connectivity among different reservoirs and can be drawn from any desirable distribution. In this work, we focus on a hierarchical structure of two ESNs with different values for the two leakage terms.

While the exploitation of multiple ESNs can enrich the dynamics of the system by discovering temporal dependencies over multiple time-scales, the definition of sparse representations through the SPARCE model [1] can enhance the capacity of the reservoir to learn associations by introducing specialised neurons through the definition of learnable thresholds.

- **Hierarchical ESN (`H-NV-ESN`)**, composed by two reservoir connected unidirectionally. The read-out is defined from both reservoirs and, as for the case of a single ESN, $\mathbf{W}_{\mathrm{out}}$ is subject to training. In this case, the number of hyper-parameters is theoretically more than doubled in comparison to a single ESN and it is practically challenging to perform an exhaustive tuning procedure of all of them. We selected the value of γ as the optimal one found for the single `ESN` and fixed $\alpha^{(1)} \approx 1$, focusing on the tuning of $\alpha^{(22)}, \rho^{(21)}, \eta$. The constraint $\alpha^{(1)} \approx 1$ is justified by considering that the second reservoir would lose information that lives on fast time-scales if $\alpha^{(1)} \ll 1$, leading to an overall system with slow reacting dynamics. On the contrary, if $\alpha^{(1)} \approx 1$ and $\alpha^{(2)} < \alpha^{(1)}$, the first reservoir can react to rapid changes of the input and the second can maintain past temporal information, leading to a system that is robust to signals with both short and long temporal dependencies.
- **Hierarchical ESN and SpaRCe (`H-NV-SPARCE-ESN`)**, which is the same as a hierarchical reservoir, but with the addition of `SPARCE` (for more details, see Section II.C in the main paper).

II. Results

We assessed if a hierarchical ESN architecture would produce improved results in the GardensPoint dataset. For the initial analysis, we kept the reservoir size $N = 1000$, leading to `NV-ESN` and `NV-SPARCE-ESN` having 1000 neurons (as in the main paper experiments), and `H-NV-ESN` and `H-NV-SPARCE-ESN` having 2000 neurons. Then, to ensure a fair

This work was supported by the Engineering and Physical Sciences Research Council under Grant EP/P006094/1, EP/S030964/1, EP/S009647/1 and EP/V006339/1, and by Templeton World Charity Foundation Grant № 0539.

[1]Department of Computer Science, The University of Sheffield, UK e.vasilaki@sheffield.ac.uk
[2]Department of Biological Sciences, Macquarie University, Australia
[3]School of Engineering and Informatics, University of Sussex, UK
[4]Institute for Neuroinformatics, University of Zurich and ETH Zurich, Switzerland
[†]These authors have equally contributed to the work.
[*]Joint senior authors.

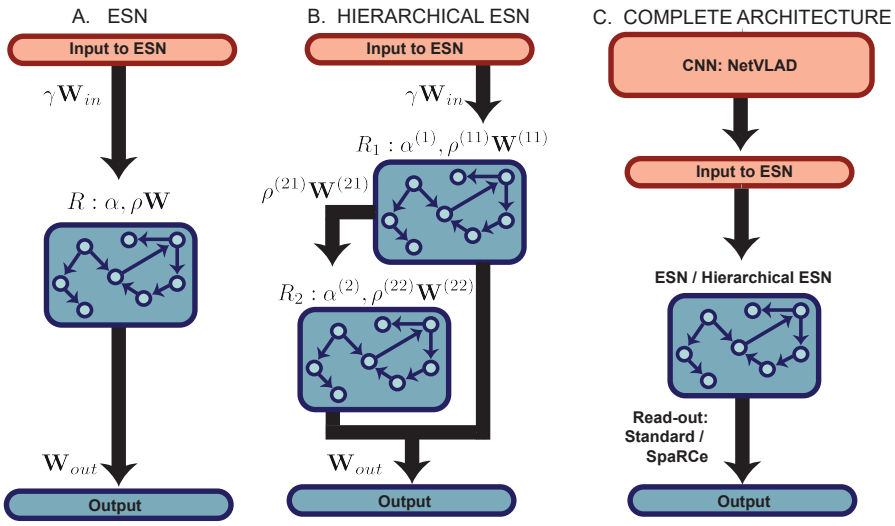

Fig. 1: **Scheme of the ESN models and the overall network architecture. A**: ESN protocol. The input is fed to an ESN and the training process occurs on the read-out $\mathbf{W}_{\text{out}}$ from the network representation. When the SPARCE algorithm is adopted, additional thresholds θ are initialised and adapted through the gradient. **B**: Hierarchical ESN. The input is first processed by the first reservoir (R_1), which is then connected to a second ESN (R_2, tuned with different values of the hyper-parameters to exhibit diverse dynamical properties) unidirectionally. As in **A**, learning occurs on the output weights $\mathbf{W}_{\text{out}}$ defined from the representation of both reservoirs and on the thresholds θ when SPARCE is adopted. **C**: Scheme of the overall model, composed of a pre-processing module (red boxes) and a reservoir model (blue boxes). In the pre-processing, an image is fed through a CNN (i.e. NetVLAD [2]), and through a hidden layer (the input to the ESN), pre-trained to reduce the dimensionality of NetVLAD output (4096 to 500) and to be fed into the reservoir system. The reservoir model can then be a single or hierarchical ESN with or without the SPARCE model. Input images are perceived sequentially as a video, and the network has to correctly classify the location of the current image

comparison, we increased the single reservoirs to match the size of the hierarchical models, $N = 2000$. Comparison of these three settings: (i) single reservoir models with $N = 1000$, (ii) with $N = 2000$ and (iii) hierarchical models with $N = 2000$, is given in Table I.

The results shows that the introduction of hierarchical models, compared to single reservoir $N = 1000$, increased the median accuracy scores while decreasing their variance; `H-NV-ESN` median: 0.80 and std: 0.14 vs `H-NV-SPARCE-ESN` median: 0.87 and std: 0.06; both for 20 trials). AUC scores showed little change, but they were already close to the maximum possible (> 0.93), and thus there was little room for improvement. Their performance was slightly below the single reservoir $N = 2000$ neurons. We would like to highlight that in the hierarchical models only the leakage terms (α) and the learning rate (η) were optimised.

Considering the performance improvement consequent to the utilisation of the hierarchical model, it is evident how the GardensPoint dataset [3] contains longer temporal dependencies among images that a single ESN cannot capture. After an inspection of the datasets, it is clear that data of GardensPoint are captured at a higher frame-rate than the other datasets (ESSEX3IN1, SPEDTest, Corridor), where images appear more static and separated in time across each other. Consequently, GardensPoint has a more complex underlying temporal structure. At this point, we see the benefit of using hierarchical models, noting that further study is needed to assess their utilities for the VPR problem thoroughly.

TABLE I: Comparison of hierarchical models against single reservoir NV−ESN and NV−SPARCE. N is the total number of reservoir neurons. The results are for 20 trials.

Model	Accuracy mean (std)	AUC mean (std)
NV−ESN ($N = 1000$)	0.748 (0.142)	0.904 (0.07)
NV−ESN ($N = 2000$)	0.85 (0.093)	0.932 (0.028)
H−NV−ESN ($N = 2000$)	0.829 (0.094)	0.913 (0.045)
NV−SPARCE−ESN ($N = 1000$)	0.772 (0.106)	0.926 (0.04)
NV−SPARCE−ESN ($N = 2000$)	**0.867(0.059)**	**0.951 (0.024)**
H−NV−SPARCE−ESN ($N = 2000$)	0.858 (0.056)	0.936 (0.036)

III. Hyper-parameters

In this section, we provide the hyper-parameters used for the ESNs. Table II shows the hyper-parameters used for datasets: GardensPoint, SPEDTest, ESSEX3IN1, Corridor, and Nordland (subset). Table III shows the hyper-parameter used for Nordland and Oxford RobotCar datasets.

TABLE II: Hyper-parameters used for experiments in Section IV. A,B,E: GardensPoint, SPEDTest, ESSEX3IN1, Corridor, and (subset) Nordland. For the hierarchical models, two hyper-parameters are for two reservoirs.

Dataset	Model	η	α	γ	P_n
GardensPoint	NV−ESN (N=1000)	0.01	0.678032	0.000316	−
	NV−SPARCE−ESN (N=1000)	0.01	0.739869	0.000316	0.4
	NV−ESN (N=2000)	0.0055	0.707946	0.000888	−
	NV−SPARCE−ESN (N=2000)	0.005	0.723563	0.000888	0.5
	H−NV−ESN (N=2000)	0.001	0.6 / 0.9	0.01 / 0.01	−
	H−NV−SPARCE−ESN (N=2000)	0.0005	0.6 / 0.9	0.01 / 0.01	0.4
SPEDTest	NV−ESN	0.01	0.957745	0.00072	−
	NV−SPARCE−ESN	0.01	0.978873	0.00072	0.1
ESSEX3IN1	NV−ESN	0.01	1	0.003728	−
	NV−SPARCE−ESN	1×10^{-5}	1	0.003728	0
Corridor	NV−ESN	0.001	0.841395	0.003728	−
	NV−SPARCE−ESN	0.005	0.841395	0.003728	0.4
Nordland (subset, 1000 images)	NV−ESN	0.01	1	0.01	−
	NV−SPARCE−ESN	0.0001	1	0.01	0.25

TABLE III: Hyper-parameters used for experiments in Section IV. C,D,E: Nordland and Oxford RobotCar.

	Nordland Summer vs Winter		Oxford Day vs Night	
Parameter	NV−ESN	NV−SPARCE−ESN	NV−ESN	NV−SPARCE−ESN
N	8000	8000	6000	6000
α	1.0	1.0	0.3	0.3
γ	0.01	0.01	0.008	0.008
ρ	0.99	0.99	0.99	0.99
P_n	−	0.0	−	0.7
N_{batch}	200	200	30	30
η	1×10^{-4}	5×10^{-4}	5×10^{-4}	1×10^{-3}
η_θ	−	5×10^{-7}	−	1×10^{-6}

References

[1] L. Manneschi, A. C. Lin, and E. Vasilaki, "SpaRCe: Improved Learning of Reservoir Computing Systems through Sparse Representations," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[3] A. Glover, "Day and night, left and right," 2014. [Online]. Available: https://doi.org/10.5281/zenodo.4590133