Generalized Learning with Reservoir Computing

Sanjukta Krishnagopal[1], Yiannis Aloimonos[2], and Michelle Girvan[1,3,4]

[1,*] Department of Physics, University of Maryland, College Park, MD 20740, USA
[2] Department of Computer Science, University of Maryland, College Park, MD 20740, USA
[3]Santa Fe Institute, New Mexico 87501, USA
[4]London Mathematical Laboratory, London WC2N 6DF, UK
[*]Corresponding author: Sanjukta Krishnagopal sanjukta@umd.edu

Abstract

We investigate how machine learning schemes known as a Reservoir Computers (RCs) learn concepts such as 'similar' and 'different',and other relationships between pairs of inputs and generalize these concepts to previously unseen types of data. RCs work by feeding input data into a high-dimensional dynamical system of neuron-like units called a 'reservoir' and using regression to train 'output weights' to produce the desired response. We study two RC architectures, that broadly resemble neural dynamics. We show that an RC that is trained to identify relationships between image-pairs drawn from a subset of handwritten digits (0-5) from the MNIST database *generalizes* the learned relationships to images of handwritten digits (6-9) *unseen* during training. We consider simple relationships between the input image pair such as: same digits (digits from the same class), same digits but one is rotated 90°, same digits but one is blurred, different digits, etc. In this dataset, digits that are marked the 'same' may have substantial variation because they come from different handwriting samples. Additionally, using a database of depth maps of images taken from a moving camera, we show that an RC trained to learn relationships such as 'similar' (e.g., same scene, different camera perspectives) and 'different' (different scenes) is able to generalize its learning to visual scenes that are very different from those used in training. RC being a dynamical system, lends itself to easy interpretation through clustering and analysis of the underlying dynamics that allows for generalization. We show that in response to different inputs, the high-dimensional reservoir state can reach different attractors (i.e. patterns), with different attractors representative of corresponding input-pair relationships. We investigate the attractor structure by clustering the high dimensional reservoir states using dimensionality reduction techniques such as Principal Component Analysis (PCA). Thus, as opposed to training for the entire high dimensional

reservoir state, the reservoir only needs to learn these attractors (patterns), allowing it to perform well with very few training examples as compared to conventional machine learning techniques such as deep learning. We find that RCs can not only identify and generalize linear as well as non-linear relationships, but also combinations of relationships, providing robust and effective image-pair classification. We find that RCs perform significantly better than state-of-the-art neural network classification techniques such as convolutional and deep Siamese Neural Networks (SNNs) in generalization tasks both on the MNIST dataset and scenes from a moving camera dataset. Using small datasets, our work helps bridge the gap between explainable machine learning and biologically-inspired learning through analogies and points to new directions in the investigation of learning processes.

1 Introduction

Different types of Artificial Neural Networks (ANNs) have been used for the task of feature recognition and image classification. Feedforward machine learning architectures such as convolutional neural networks (CNNs)[1], deep neural networks [2], stacked auto encoders[3] etc. and recurrent architectures such as recurrent neural networks [4], Long Short-Term Memory (LSTM)[5] etc. have been immensely successful for several tasks from speech recognition [6] to playing GO [2].

There have also been a number of rapid advances in other machine learning architectures such as Echo State Networks (ESN) (originally proposed in the field of machine learning) [7] and Liquid State Machines (LSMs) (originally proposed in the field of computational neuroscience) [8] , commonly falling under Reservoir Computing (RCs) [9]. Compared to deep neural networks, ANN-based RCs are a brain-inspired machine learning framework, and have been show to be a pertinent framework to model local cortical dynamics and their contribution to higher cognitive function [10].

The goal of this work is to demonstrate the unreasonable efficiency of Reserving Computers (RCs) in learning the relationships between images with very little training data and consequently being able to generalize the learned relationships to types of images it hasn't seen before. We concede that other machine learning techniques such as deep learning [11] and CNNs may also be useful for this task and have proven to be extremely successful at image classification. However, we speculate that, because of their complex dynamical character, RCs may inherently be better suited for learning from a small training set and generalization of this learning [12].

RCs are dynamical systems which non-linearly transform the input and have reproducibility to a repeated input signal that can serve as a resource for information processing. They appealing because of their dynamical properties and easy scalability since the recurrent connections in the network aren't trained. Applications of RC include many real world phenomena such as weather or stock market prediction, self driven cars, speech processing and language interpretation, gait generation and motion control in robots etc, several of which are inher-

ently non-linear. RCs are also appealing because of their biologically-inspired underpinnings. Biological systems such as the visual cortex are known to have primarily (70 %) recurrent connections with less than 1 % of the connections being feedforward [13]. RCs (or closely related models) provide insights into how biological brains can carry out accurate computations with an 'inaccurate' and noisy physical substrate [14], especially accurate timing of the way in which visual spatiotemporal information is super-imposed and processed in primary visual cortex [15]. In addition, biological systems are known to learn visual concepts through analogies, using only a handful of examples [16]. In particular, in [17], bees were trained to fly towards the image in an image pair that looked very similar to a previously displayed base image. On training bees to fly towards the visually similar image, the bees were presented with two scents, one very similar and one different from a base scent. As a consequence of the visual training that induced preference to the very similar category, the bees flew towards the very similar scent. Recent work has also been done on the phenomenon of 'peak shift', where animals not only respond to entrained stimuli, but respond even more strongly to similar ones that are farther away from non-rewarding stimuli [18]. Thus, biological systems have been found to translate learning of concepts of similarity across sensory inputs, leading us to believe that the brain has a common and fundamental mechanism that comprehends through analogies or through concepts of 'similarity'.

In our framework, we refer to generalization as the ability of a system to learn the relationships or transformations, both linear and non-linear, between a pair of images and be able to recognize the same relationship in unseen image-pairs. Learning through analogies is a recurring biologically phenomenon, which allow seems to allow for easy generalization of the learned relationships in biological systems. Compared to machine learning approaches, humans learn much richer information using very few training examples. Moreover, people learn more than how to do pattern or object recognition: they learn a concept – that is, a model of the class that allows their acquired knowledge to be flexibly applied in new ways [19]. While many machine learning approaches can effectively classify images with human-like accuracy with sufficient data, these approaches often require large datasets and hence increasingly powerful GPUs do not scale well. Despite the fact that research in learning from very few images, one shot learning [20] etc., has gained momentum recently, integrating it with generalization of learning is a relatively unexplored area. In our framework, the RC not only requires very few training examples compared to techniques such as deep learning, but can also effectively use analogies to learn relationships, leading to easy generalization.

RCs are built on several prior successful approaches that emphasize the use of a dynamical system, i.e., existence of attractors, for successful, neuro-inspired learning. In the ground-breaking work of Hopfield in [21], the success of Recurrent Neural Networks (RNNs) depend on the existence of attractors. In training, the dynamical system of the RNN is left running until it ends up in one of its several attractors. Similarly, in [22], a unique conceptor is found for each input pattern in a driven RNN. However, training of RNNs is difficult due to training problems like exploding or vanishing gradient. RCs overcome this problem by

training only the output weights. Models of spontaneously active cortical circuits typically exhibit chaotic dynamics, as in RCs [23, 24]. RC offers a convenient solution to some the problems with an RNN, while offering the same advantages. In this work, we train RCs on both the MNIST handwritten digit database as proof of concept as well as depth maps of visual scenes from a moving camera, to study generalization of the learned relationships between pairs of images. The reservoir activity is then studied to reveal the underlying dynamical features of the activity that classification can be attributed to. We find that the same type of relationship cluster in reservoir space, i.e., the reservoir space is made of several local attractors corresponding to the relationships. This allows for generalization of the learned relationships to all image pairs, seen and unseen by the reservoir. Additionally we compare its performance for a generalization task to a pair-based deep siamese neural network (SNN) and a convolutional siamese neural network (CSNN) and show that the reservoir performs significantly better, both for simpler MNIST images as well as for depth maps . We also show that the reservoir is able to recognize linear combinations of the individuals relationships it has learned. This work can useful in the field of computer vision to classify relationships between images, even if they are non-linear as in a moving camera, in a biological plausible and computationally efficient way.

2 Data and Methods

We use two datasets for this work: 1. The handwritten digit database MNIST: the MNIST database consists of 70000 images, each 28×28 pixels in size, of handwritten digits 0-9. 2. Depth maps from a moving camera: consists of depth maps from 6 different visual scenes recorded indoors in an office setting (refer supplementary material for complete dataset). Each visual scene has depth maps from at least 300 images, each compressed to 100×100 pixels in size, recorded as the camera is moved within a small distance (∼30cm) and rotated within a small angle (∼30°). A sample of three RBG images from one of the 6 classes is shown in Fig. 1.

In our framework, images are always considered in pairs (image 1 and image 2). We study five relationships- noise, rotated, zoomed, blurred, and different. We are interested in exploring relationships between images through concepts of 'similarity' and 'difference'. Such relationships are a natural extension of these concepts. Examples of the image pair relationships applied to the MNIST dataset is shown in Fig. 2. We create the image pairs as follows:

Figure 1: Examples of images taken from a moving camera from the same class. A pair of these would be classified under the category 'similar'

1. Noise: Two different images from the same class are taken directly from

the MNIST database (Ex. Fig 2(a)). One of the images in the pair (image 1) remains untransformed, whereas the other (transformed) image is superimposed with random noise with peak value given by 20 % of the peak value of image 1.

2. Rotated: Two different images from the same class are taken. Image 2s is 90° rotated (Ex. Fig 2(b))

3. Zoomed: Image 2 is zoomed with a magnification of 2 (Ex. Fig 2(c)).

4. Blurred: Image 2 is blurred (Ex. Fig 2(d)) by convolving every pixel of the image by a 6×6 convolution matrix with all values 1/36:

5. Different: Two different images from different classes (Ex. Fig 2(e)).

All pairs are characterized by the relationship between the image-pairs. For instance, we call a pair rotated if we start from two different handwritten images of the same digit and rotate the second image 90^{circ} with respect to the first. Since two different handwritten images of the same digit are used, the image pair involves an initial non-linear transformation in addition to the applied transformation.

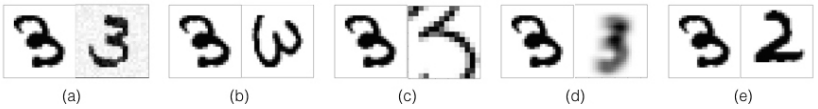


Figure 2: Pairs of images that are representative of the transformations classified into five labels: (a) very similar, (b) rotated by 90^{circ}, (c) zoomed, (d) blurred and (e) different.

2.1 Network Architecture

In this work we use the Echo State Network (ESN) class of RCs for training and classification. Our RCs are neural network with two layers: a hidden layer of recurrently interconnected non-linear nodes, driven both by inputs as well as by feed-backs from other nodes in the reservoir layer and an output or readout layer. Only the output weights of the reservoir are trained. The reservoir, being a dynamical system, works particularly well for analyzing time-series input data due to its memory and high-dimensional projection of the input [25, 26]. The input images are hence converted into a 'time-series' by feeding the reservoir a column of the input image at each time point (as in [27]). The method of 'temporalization' of the input (row-wise, column-wise etc.) simply changes the input representation and doesn't affect the analysis. While there is limited understanding of the actual processes through which the brain processes analogies, we explore two models that represent cortical processing of relationships between inputs. There has also been some evidence [28] of integrated processing, particularly in the visual cortex. To mimic an integrated processing system more closely, we study the Single Reservoir architecture (Fig. 3(a)). However, there is some evidence that analogy processing involves two steps:

1) the brain generated individual mental representations of the different inputs and 2) brain mapping based on structural similarity, or relationship, between them [29]. We create the Dual Reservoir architecture (Fig. 4)in an attempt to mimic this process of parallel processing of signals, followed by mapping based on the differences between the processed signal in the cortex. Since, there isn't a consensus in the neuroscience community about the details of cortical processing, we present both the single and dual reservoir architecture here:

2.1.1 Single Reservoir Architecture

Input Layer As discussed above, in order to exploit the dynamical system properties of RCs, the input is converted to a time series. We vertically concatenate the image pair to form the combined image. Then we input the combined image column by column (shown in Fig. 3(b) for the MNIST database) into the reservoir, allowing the time axis to run across the rows of the image. While this 'temporalization' may seem artificial, there's a unique reproducible reservoir state corresponding to each image causing the results to be independent of order of temporalization, as long as all images are temporalized the same way.

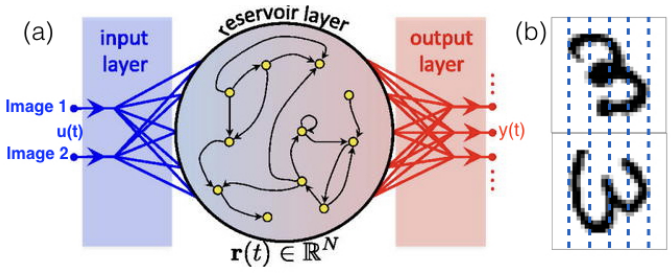


Figure 3: (a) Reservoir architecture with input state of the two images at time t denoted by $\vec{u}(t)$, reservoir state at a single time by $\vec{r}(t)$ and output state by $\vec{y}(t)$. (b) shows one image pair from the rotated 90° category of the MNIST dataset split vertically and fed into the reservoir in columns of 1 pixel width, shown to be larger here for ease of visualization.

Reservoir Layer The reservoir can be thought of as a dynamical system described by a reservoir state vector $\vec{r}(t)$ which describes the states of the reservoir nodes as a function of time t. The reservoir state $\vec{r}(t)$ is given by:

$$\vec{r}(t+1) = \tanh\left(W^{\text{in}} \cdot \vec{u}(t) + W^{\text{res}} \cdot \vec{r}(t) + b\right) \tag{1}$$

The input weights matrix $W^{\text{in}} \in \mathbb{R}^{N_R \times N_u}$, where N_R is number of nodes in the reservoir and N_u is the dimension of the input vector $\vec{u}(t)$; here N_u is the number of rows of the concatenated image. The activity of the reservoir at time t is given by $\vec{r}(t)$, of size N_R. The recurrent connection weights $W^{\text{res}} \in \mathbb{R}^{N_R \times N_R}$

are set randomly between -1 and 1. b is a scalar bias. We use hyperbolic tangent as the non-linear activation function. We set the spectral radius γ (maximal absolute eigenvalue of W^{res} at 0.5, but we observe performance to be insensitive to this choice 10. The reservoir is a dynamical system that transforms the low dimensional input into a much higher dimensional reservoir space and reaches its optimal performance even when the W^{out} and W^{res} are sparse. Matrix sparsity is 0.9 unless otherwise stated.

Output Layer The composite output reservoir state of one reservoir for one image $\tilde{X}$ is formed by concatenating the reservoir state (the state of all reservoir nodes) at every timestep $\vec{r}(t)$ as follows:

$$\tilde{X} = \vec{r}(0) \oplus \vec{r}(t=1) \oplus \ldots \oplus \vec{r}(t=T). \tag{2}$$

$\tilde{X}$ is a matrix of size $N_R \times c$ where c is the number of columns in the image (number of time steps (T)through which the entire image is input).
The output/readout layer representation (Y_i) for a very similar pair is $(1,0,0,0,0)$, rotated pair is $(0,1,0,0,0)$, zoomed pair is $(0,0,1,0,0)$, blurred pair is $(0,0,0,1,0)$ and different pair is $(0,0,0,0,1)$. The output weights convert the output reservoir states $\tilde{X}_k$ into the reservoir output y_i. Ridge regression (refer A) is then used to train the output weights of the reservoir. While testing, the reservoir computer allots a fractional probability to each output label, and the image pair is classified into the label with the highest probability.

2.1.2 Dual Reservoir Architecture

Input Layer In order to exploit the dynamical system properties of RCs, the input is converted to a time series. However unlike the single reservoir architecture, we input each image (image 1 and image 2) column by column into two identical reservoirs, allowing the time axis to run across the rows of the image.

Reservoir Layer The reservoir states for the two images, $\vec{r_1}(t)$ and $\vec{r_2}(t)$, are given by :

$$\begin{aligned} \vec{r_1}(t+1) &= \tanh\left(W^{\text{in}} \cdot \vec{u}(t) + W^{\text{res}} \cdot \vec{r_1}(t) + b\right) \\ \vec{r_2}(t+1) &= \tanh\left(W^{\text{in}} \cdot \vec{v}(t) + W^{\text{res}} \cdot \vec{r_2}(t) + b\right) \end{aligned} \tag{3}$$

The properties of the internal dynamics of the reservoir are the same as the single reservoir. The two reservoirs used in the Dual architecture are identical.

Output Layer The total reservoir state X of one reservoir for one image is then formed by concatenating the reservoir state (the state of all reservoir nodes) at every timestep $\vec{r}(t)$ as follows:

$$X = \vec{r}(0) \oplus \vec{r}(t=1) \oplus \ldots \oplus \vec{r}(t=T). \tag{4}$$

X is a matrix of size $N_R \times c$ where c is the number of columns in the image (number of time steps (T)through which the entire image is input).
The $k^{th}z$ output reservoir state is given by $\tilde{X}_k = \Delta X_k = \Delta X_{k(i,j)} = |X_i - X_j|$, where X_i is the reservoir state of the k^{th} image corresponding to the image 1 and X_j corresponds to the image 2. The readout layer representations for different transformations are the same as that in the single reservoir case. Ridge regression (refer Appendix A) is then used to train the output weights of the reservoir.

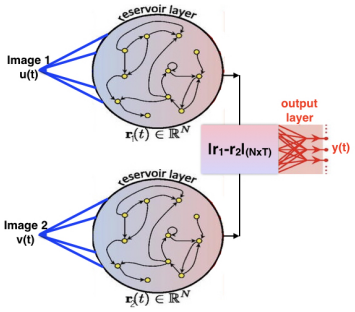


Figure 4: (a) Dual reservoir architecture with input state of the two images at time t denoted by $\vec{u}(t)$ and $\vec{v}(t)$, reservoir state by $\overrightarrow{r_{1,2}}(t)$ and output state by $\vec{y}(t)$.

3 Results

3.1 Generalization to Untrained Image Classes

In this section we discuss the performance of the single and dual reservoir set-up for the task of generalization of learned relationships. We present the results obtained on the MNIST dataset as proof of concept. The systems were trained on the five relationships – noise added, 90° rotation, blur, zoom, different (i.e. no relationship), on image-pairs of handwritten digits 0-5. Then they were tested on identifying the relationships between image pairs of handwritten digits 6-9 (digits they have never seen before). We use fraction correct (1- error rate) as a metric of performance.

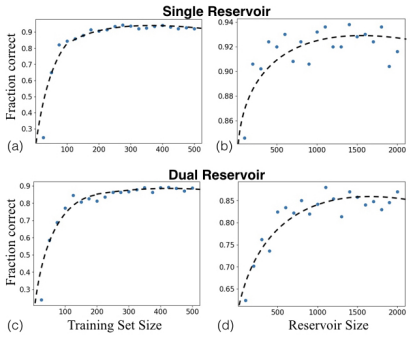


Figure 5: Fraction of image-pairs correctly classified versus training set size (a&c) and reservoir size (b&d). Single reservoir results show in (a&b). Dual reservoir results shown in (c&d). Reservoir size=1000 nodes for (a&c); training size=250 pairs for (b&d). Spectral radius $\gamma = 0.5$, sparsity = 0.9.

In Fig. 5(a&c), we see that the reservoir performance increases rapidly with training set size and plateaus at around 200 training pairs. A training set size of ∼250 image pairs gives a reasonable trade-off between performance and computational efficiency. This is significantly lower than the training set sizes typically used in deep learning. A biologically reasonable system is expected to train with relatively few training examples, as our system does. Fig. 5(b&d) shows that for a constant training data size (250 pairs) the performances increase as expected with reservoir size up to around 750 nodes after which it saturates. The overall optimal performance of

the single reservoir appears to be better than that of the dual reservoir. Further, we also examine the reservoir performance as a function of the spectral radius γ in Fig. 10; we see no clear dependence on the spectral radius for the range investigated. For reference, reservoir activity as well as single node activity and output weights are shown in the Appendix (B).

3.2 Comparison with Deep Siamese Network

The topic of generalized learning has, to the the best of our knowledge, not been satisfyingly addressed using a dynamical-systems-based machine learning approach. To assess the effectiveness of our approach, we compare the performance of RCs with variants of a Siamese Neural Network (SNN), a successful pair-based machine learning technique (architecture shown in Fig. 6). Specifically, we compare the single and dual reservoir model to three other architectures: a base SNN perceptron with 4 fully connected layers of 128 neurons each, a deep SNN perceptron with 8 fully connected layers of 128 neurons each, and a convolutional SNN (convolutional layer with 32 filters, 3X3 kernel and a rectified linear non-linearity, followed by 4 fully connected layers with 128, 64,32 and 2 neurons each). We compared performance for two binary classification tasks (Fig. 7(c)): 1. Learning the 90^{circ} rotation operator on MNIST image pairs 2. Learning to detect depth maps that come from the same visual scene class for the dataset of depth maps from a moving camera.

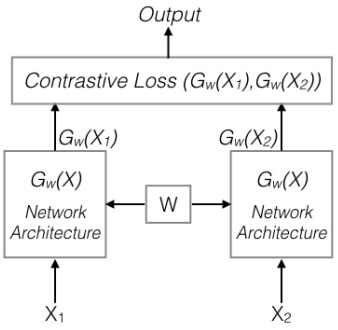
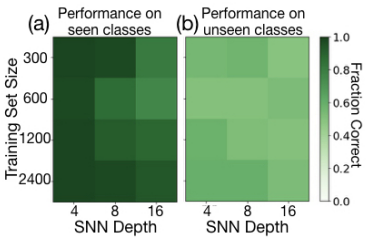


Figure 6: Siamese Network Architecture. Two inputs $X1$ and $X2$ are fed into two identical networks. $G_w(X)$ is the network transformation of the input X. W is the shared weights between the two heads of the siamese architecture.

All SNN architectures were trained using contrastive loss (following [30]) and we use the optimizer Adadelta with a self adjusting learning rate. The single and dual reservoirs have 1000 nodes with $\gamma = 0.5$ and sparsity 0.9. Training is done for a 100 (40) epochs on the base and deep SNN perceptrons, 40 (20) epochs on the convSNN for MNIST (visual scenes) data respectively and once on the reservoirs on 500 image pairs. While we present a select few SNN architectures here (and selected choices of parameters), we tried several other SNN architectures including VGG16-SNN and deep convSNN and found their performance to be comparable to the representative SNN performances we have shown. We also show SNN perceptron performance on varying depth (number of layers) and varying training data size (varied in the lower range compared to traditional deep network training sizes for comparison with the RCs and to motivate the question of biological plausibility) while testing on seen (trained) classes and unseen (test) classes (Fig. 7(a&b) respectively) and find that while the network performs fairly well on the trained classes, it performs consistently poorly on

the unseen classes. The loss and accuracy plots for the SNN architectures for both tasks are in appendix C.



(c) Fraction correct for different network architectures

	Single Reservoir	Dual Reservoir	Base SNN	Deep SNN	ConvSNN
Digits 6-9 (unseen classes)	**0.91**	**0.89**	0.54	0.51	0.53
Digits 0-5 (seen classes)	0.94	0.91	0.93	0.91	0.89
Scenes 4-6 (unseen classes)	**0.85**	**0.95**	0.53	0.51	0.78
Scenes 0-3 (seen classes)	0.87	0.95	0.89	0.89	0.93

Figure 7: SNN perceptron performance on trained (seen) classes (a) and test (unseen) classes (b) of MNIST data as a function of training dataset size and SNN perceptron depth. (b) Classification accuracy (fraction correct) of the single and dual reservoir, base SNN, deep SNN and convSNN on seen (trained) classes and unseen (test) classes, on (1) identifying rotation transformation in MNIST images, and (2) identifying similar visual scenes from a moving camera. Training size: 500 images.

3.2.1 Generalized Learning of the Rotation Operator on the MNIST dataset

We trained the reservoir on a simple binary classification task, i.e., classify image pairs from the MNIST dataset as having the relationship 'rotated' or not. Our training set consists of rotated and not rotated images of digits 0-5. Fig. 7(c))shows the fraction of correct classification of the RCs and the SNNs on the training classes (seen, digits 0-5) and testing classes (unseen, digits 6-9), as rotated or not rotated. We observe that, while the performance of all the networks is comparable on training set digits (digits 0-5), all the SNN architectures seems to classify randomly for untrained digits (6-9). Performance didn't improve on increasing the depth of the base SNN (Fig. 7(a&b)). The reservoir performance remains equally good over trained digits (0-5) and untrained digits (6-9), indicative of learning of the underlying relationship in the pairs and not the individual digits themselves. As seen in section 3.4, the generalization ability of the reservoir may be attributed to the convergence of parts of the dynamical reservoir state for all rotated image-pairs, a concept analogous to that of an attractor in dynamical systems. In contrast, the SNN isn't a dynamical system, and training occurs explicitly on the images as opposed to the classes of relationships, leading to poorer performance while generalizing. However, we present performance of an fully connected SNN obtained by varying the SNN depth, training data size in Fig 7(a&b).

3.2.2 Generalizing Similarities in Depth Perception from a Moving Camera

Identifying similarities in scenes, properties of scenes such as depth, style etc. from a moving camera is an important problem in the field of computer vision [31, 32]. We are interested in studying how the reservoir could learn and generalize relationships between images from a moving camera, frames of which may be non-linearly transformed with respect to each other. To demonstrate the practicality of our method, we implement it on depth maps from 6 different visual scenes recorded indoors in an office setting. Each visual scene has depth maps from 300 images, recorded as the camera is moved within a small distance ($\sim$30cm) and rotated within a small angle ($\sim 30^\circ$). We then train the networks to identify pairs of depth-maps as very similar (same visual scene) or different (different visual scenes), learning to capture small spatial and rotational invariance. Training is done on 500 images each from the first three visual scenes. We study whether the systems are able to generalize, i.e., identify relationships between depth maps from the other three visual scenes. Fig. 7(c) shows the reservoir performs significantly better on untrained scenes than the SNN, which classifies randomly. Both systems have a comparable and very high performance on the trained scenes. Thus, the reservoir is able to identify frames with similar depth maps from scenes it hasn't seen before. This has potential applications in scene or object recognition using a moving camera.

3.3 Combining Relationships

In the section we train the reservoir independently on the five relationships as before. However our test input images have a linear combination of multiple relationships applied on them simultaneously (e.g., rotated as well as blurred). We then study the ability of the reservoir to recognize all the separate relationships applied to the test input pair.

Training is done on the five individual relationships (noise, rotated, blurred, zoomed and different) for digits 0-5. Testing is done on a combination relationships (90° rotation and blurring) as well as only 90° rotation for digits 6-9. For testing image-pairs with n relationships applied simultaneously, we consider the reservoir to have classified correctly if the n highest probabilities correspond to the n applied relationships. In Fig. 8 we observe that both the reservoirs perform very well (in terms of classification percent correct) at identifying combined relationships in images that they have never seen before. The single reservoir, on average, performs slightly better than the dual reservoir. While there may be some inherent biases (ex. in Fig. 8(f), the dual reservoir shows a bias towards the zoomed category), in spite of the biases, the reservoirs are able to not only generalize the learned relationships, but also identify and separate linear combinations of these relationships in previously unseen images. We speculate that this ability to generalize combinations of multiple relationships is a result of overlap of regions in reservoir space that correspond to the separate relationships. While we only present a few combinations here, we also ran several tests on

other subsets/combinations of relationships and the RC consistently performs very well.

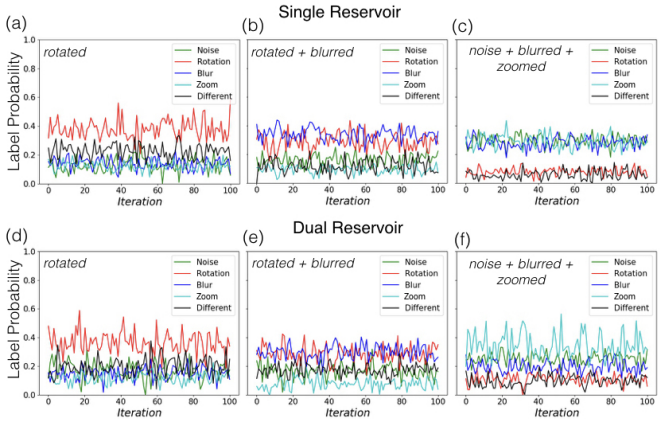


Figure 8: Label probability for images that are rotated (a,d) 2 combination: rotated and blurred (b,e), 3 combination: noise, blurred and zoomed (c,f), for single and dual reservoir respectively. The fraction correct, where classification is considered to be correct if the n predicted maximum probability labels are the n transformations applied to the test image-pair (shown on top left of each panel), are $0.97, 0.97, 1.0, 0.93, 0.84, 0.93$ for (a,b,c,d,e,f) respectively. γ=0.5, reservoir size=1000. Training digits: 0-5, testing digits: 6-9. Training size: 250 pairs.

3.4 Dimensionality Reduction of Reservoir Space

A possible explanation for the ability of the reservoir to learn relationships between pairs of images and generalize to unseen images comes from dynamical systems analysis. In order to generalize, for a given relationship between the input image pairs, there must be a corresponding relationship between the reservoir activity, dependent only on the relationship between the input images and not on the input images themselves. From a non-linear dynamical perspective, this relates to the attractor structure of the reservoir dynamics. In this section we show that reservoir states corresponding to a relationship do indeed cluster in reservoir space, allowing for generalization.

In Fig. 9, we plot a representation of 500 total output reservoir states for each relationship (using different input digits) for (a) the single reservoir and (b) the dual reservoir. We show here the five standard relationships for MNIST - noise, rotate, blur, zoom, different, as well as one combined relationship - blur+rotate. A single output reservoir state has a very high dimensionality ($N_R \times T$). We are interesting in viewing this high dimensional data in a reduced dimensional space. Hence, we use the following dimensionality reduction techniques - first, we use Principal Component Analysis (PCA) to extract the

100 largest principal components (PCs) of each reservoir state. We then use the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [33] on the extracted PCs for further dimensionality reduction. t-SNE, being particularly well suited for the visualization of high-dimensional datasets, has been used very successfully in recent years along with PCA.

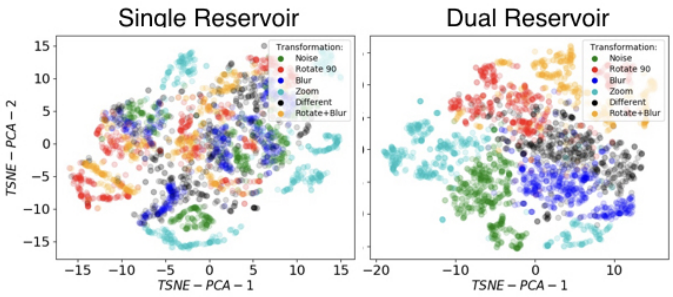


Figure 9: 500 output reservoir states for each relationship in the reduced dimensional space spanned by the two largest components obtained using t-SNE on the 100 largest principal components of the reservoir state for (a) single reservoir and (b) dual reservoir. Input images: digits 0-9 of MNIST dataset. N_R: 1000. t-SNE iterations: 300, perplexity: 40.

We observe from Fig. 9(b), that the relationships separate very well for the dual reservoir in the space of the largest two t-SNE components. Thus, the reservoir encodes features of the *relationship* between the image pair, not only the input image features themselves. We note that the separation isn't as prominent for the case of the single reservoir (Fig. 9(a)) as compared to the double reservoir. Additionally, we also notice that some relationships correspond to multiple clusters in this representation. This may allow for robustness because the RC has multiple ways of representing the same relationship. Lastly, we observe that the 'rotate+blur' relationship clusters partially overlap with (for the dual reservoir) / are in near vicinity of (for the single reservoir) 'rotate' and/or 'blur' relationship clusters. This partially explains the success of the single as well as dual reservoir in being able to identify the individual relationships involved when the input is a combination of relationships (section 3.3). Proximity of the combined cluster to one of the individual transformation clusters over the other could also explain biases induced in the reservoir while identifying combined relationships.

4 Conclusion

In this paper we have used RCs to solve a class of image classification problems that involve generalization of learning of relationships between images using limited training data. While image classification has been studied extensively

before, here we present a biologically-inspired method that not only generalizes learning, but also allows us to interpret the results analytically through a dynamical systems lens. We observe that the output reservoir states obtained from input image-pairs with the same relationships cluster in reservoir space. From a dynamical systems perspective, this can be interpreted as the attractor structure of the reservoir dynamics being associated with image-pair relationships. By reducing dimensionality from the reservoir space to the space mapped by the clusters, we are able to get a well-generalizing reservoir using only a small training dataset, whereas contemporary methods such as deep learning require much larger datasets. The clustering of dynamical reservoir states allows the reservoir to generalize the relationships learned to types of images it hasn't seen during training. Although we see strong performance with a sparse reservoir and few training images in our proof-of-concept study, we predict that for more complex input images, a more powerful (and possibly more sophisticated) reservoir would be required to match performance.
We find that the RC performs significantly better than a deep/conv SNN for the task of generalization. From a computation perspective, the RC has the added advantage of speed since only the output weights are being trained and the reservoir is sparsely connected. Our system is biologically-inspired in two ways. First, the learning mimics biological learning through comparisons and analogies. Second, the internal dynamics of the reservoir are known to resemble neural cortex activity. We conclude that although state of the art machine learning techniques such as SNNs (for pairwise input) work exceedingly well for image classification, they do not work as well for generalization of learning, for which RCs significantly outperform them, due perhaps to their dynamical properties. Thus, we see the strength of our work as lying in not only its demonstration of the utility of RCs for generalization, but also in our ability to explain this in terms of the clustering of reservoir state dynamics -through PCA and t-SNE. This relates to new ideas in explainable Artificial Intelligence (AI), a topic that continues to receive traction. An interesting direction would be to explore different reservoir architectures that model the human brain better. Another promising direction would be to study synchronization patterns in the reservoir and their effects on learning.

Data Availability The visual scenes captured from a moving camera (images and depth maps) dataset used to support the findings of this study are available from the corresponding author upon request. The dataset has not been included in the article/ supplementary material due to its large size.

Conflicts of Interest The authors declare that there are no conflicts of interest regarding the publication of this article.

Funding Statement This research was supported in part by the University of Maryland's COMBINE (Computation and Mathematics for Biological Networks) program through NSF award number 1632976.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, 2012.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[3] Bo Du, Wei Xiong, Jia Wu, Lefei Zhang, Liangpei Zhang, and Dacheng Tao. Stacked convolutional denoising auto-encoders for feature representation. *IEEE transactions on cybernetics*, 47(4):1017–1027, 2017.

[4] Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.

[5] Wenhao Zhu, Tengjun Yao, Jianyue Ni, Baogang Wei, and Zhiguo Lu. Dependency-based siamese long short-term memory network for learning sentence representations. *PloS one*, 13(3):e0193919, 2018.

[6] Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 4845–4849. IEEE, 2017.

[7] Herbert Jaeger. The" echo state" approach to analysing and training recurrent neural networks. 148, 01 2001.

[8] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.*, 14(11), November 2002.

[9] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[10] Pierre Enel, Emmanuel Procyk, René Quilodran, and Peter Ford Dominey. Reservoir computing properties of neural dynamics in prefrontal cortex. *PLOS Computational Biology*, 12(6):1–35, 06 2016.

[11] Yann LeCun, Y Bengio, and Geoffrey Hinton. Deep learning. 521:436–44, 05 2015.

[12] Masanobu Inubushi and Kazuyuki Yoshimura. Reservoir computing beyond memory-nonlinearity trade-off. *Scientific Reports*, 7(1):10199, 2017.

[13] Nuno Maçarico Da Costa and Kevan A.C. Martin. The proportion of synapses formed by the axons of the lateral geniculate nucleus in layer 4 of area 17 of the cat. *The Journal of Comparative Neurology*, 516(4):264–276, 2009.

[14] Stefan Haeusler and Wolfgang Maass. A statistical analysis of information-processing properties of lamina-specific cortical microcircuit models. *Cerebral Cortex*, 17(1):149–162, 2007.

[15] Wolf Singer Danko Nikolic, Stefan Haeusler and Wolfgang Maas. Temporal dynamics of information content carried by neurons in the primary visual cortex. NIPS'06, 2006.

[16] Peter J. Urcuioli, Edward A. Wasserman, and Thomas R. Zentall. Associative concept learning in animals: Issues and opportunities. *Journal of the Experimental Analysis of Behavior*, 101(1):165–170, 2014.

[17] Martin Giurfa, Shaowu Zhang, Arnim Jenett, Randolf Menzel, and Mandyam V. Srinivasan. The concepts of 'sameness' and 'difference' in an insect. 410:930–933, 03 2001.

[18] Samuel C. Andrew, Clint J Perry, Andrew B Barron, Katherine Berthon, Verónica Peralta, and Ken Cheng. Peak shift in honey bee olfactory learning. *Animal Cognition*, 17:1177–1186, 2014.

[19] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

[20] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in neural information processing systems*, pages 1087–1098, 2017.

[21] J. J. Hopfield. Neurocomputing: Foundations of research. chapter Neural Networks and Physical Systems with Emergent Collective Computational Abilities. 1988.

[22] H. Jaeger. Controlling recurrent neural networks by conceptors. 2014.

[23] C. van Vreeswijk and H. Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274(5293):1724–1726, 1996.

[24] Rune Rasmussen, Mogens H Jensen, and Mathias L Heltberg. Chaotic dynamics mediates brain state transitions, driven by changes in extracellular ion concentrations. *Cell Systems*, 5(6):1–13, 2017.

[25] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, 120(2):024102, 2018.

[26] Joni Dambre, David Verstraeten, Benjamin Schrauwen, and Serge Massar. Information processing capacity of dynamical systems. *Scientific reports*, 2:514, 2012.

[27] Nils Schaetti, Michel Salomon, and Raphaël Couturier. Echo state networks-based reservoir computing for mnist handwritten digits recognition. *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, pages 484–491, 2016.

[28] Jean Bullier. Integrated model of visual processing. *Brain research. Brain research reviews*, 36 2-3:96–107, 2001.

[29] Emmanuelle Volle, Sam J. Gilbert, Roland G. Benoit, and Paul W. Burgess. Specialization of the rostral prefrontal cortex for distinct analogy processes. *Cerebral Cortex*, 20(11):2647–2659, 2010.

[30] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 2006.

[31] Hossein Rahmani, Ajmal Mian, and Mubarak Shah. Learning a deep model for human action recognition from novel viewpoints. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):667–681, 2018.

[32] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1386–1383. IEEE, 2017.

[33] Laurens van der Maaten and Geoffrey Hinton. Viualizing data using t-sne. 9:2579–2605, 11 2008.

[34] Francis Wyffels, Benjamin Schrauwen, and Dirk Stroobandt. Stable output feedback in reservoir computing using ridge regression. In *Proceedings of the 18th International Conference on Artificial Neural Networks*, ICANN '08, 2008.

[35] Shishir B. Vengamoorthy G.K. Effects of spectral radius and settling time in the performance of echo state networks. *Neural Networks*, 2009.

A Ridge Regression and Training

Only the output weight matrix W^{out} is optimized during training such that it minimizes the mean squared error $E(y, Y)$ between the output of the reservoir y and the target signal Y. The reservoir output is:

$$Y = W^{\text{out}} \Delta X \tag{5}$$

$W^{\text{out}} \in \mathbb{R}^{N_y \times N_R}$ where N_y is the dimensionality of the readout layer.
ΔX or the concatenated reservoir state is the matrix containing all total reservoir states during training phase, $\Delta X = \tilde{X}_0 \oplus \tilde{X}_1 \oplus \ldots \oplus \tilde{X}_M$ where M is the total number of training image-pairs, input one after the other, and $Y = Y_0 \oplus Y_1 \oplus \ldots \oplus Y_M$ is the matrix containing the corresponding readout layer for all images. The most common way to compute W^{out} is to use Ridge Regression (or Thikonov regularization) [34], which adds an additional small cost to least square error, thus making the system robust to overfitting and noise. Ridge regression calculates W^{out} by minimizing squared error $J(W^{\text{out}})$ while regularizing the norm of the weights as follows:

$$J(W^{\text{out}}) = \eta |W^{\text{out}}|^2 + \sum_i ((W^{\text{out}})^T \Delta X_i - Y_i)^2. \tag{6}$$

where ΔX is the concatenated reservoir state over input image pairs, Y contains the corresponding label representations and the summation is over all training

image pairs. The stationary condition is

$$\frac{\partial J}{\partial W^{\text{out}}} = \eta W^{\text{out}} + \sum_i ((W^{\text{out}})^T \tilde{X}_i - Y_i)\Delta X = 0. \tag{7}$$

$$(\Delta X \Delta X^T + \eta I) W^{\text{out}} = \Delta X Y. \tag{8}$$

$$W^{\text{out}} = (\Delta X \Delta X^T + \eta I)^{-1} \Delta X Y. \tag{9}$$

where η is a regularization constant and I is the identity matrix.

B Reservoir Dynamics and Performance

We present the performance of the single and dual reservoir as a function of spectral radius γ. γ is varied from 0 to 1 while looking for the optimal performance region where the reservoir has memory or is in the 'echo state' (edge of charos) [35], however we find no indicative pattern (Fig. 10).
Performance with Spectral Radius: Fig. 10 shows fraction correct as a function of reservoir dynamics for (a) single and (b) dual reservoir.

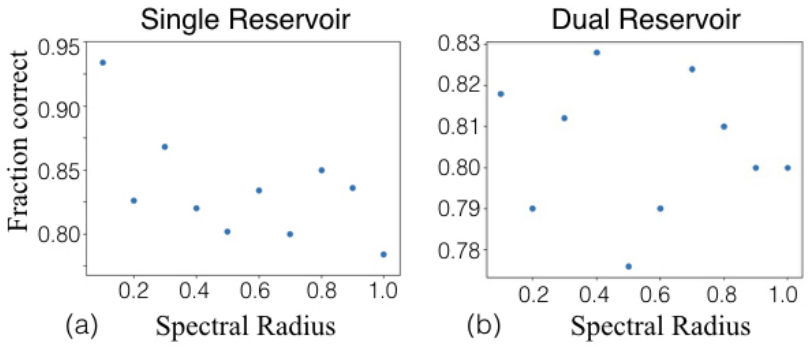


Figure 10: Fraction correct as a function of spectral radius for (a) single reservoir (b) dual reservoir. N_R=1000, training size=250 pairs, $\gamma = 0.5$, sparsity = 0.9.

Reservoir Dynamics:
For completion, we plot the reservoir activity, i.e., averaged reservoir state corresponding to our five relationships applied to the MNIST dataset, output weights, and single node activity. Fig. 1112 show plots of activity in the single reservoir and dual reservoir architecture respectively. We see that the individual node (f) itself doesn't encode any decipherable information. However each output label (a,b,c,d,e) has a different signature in reservoir space.

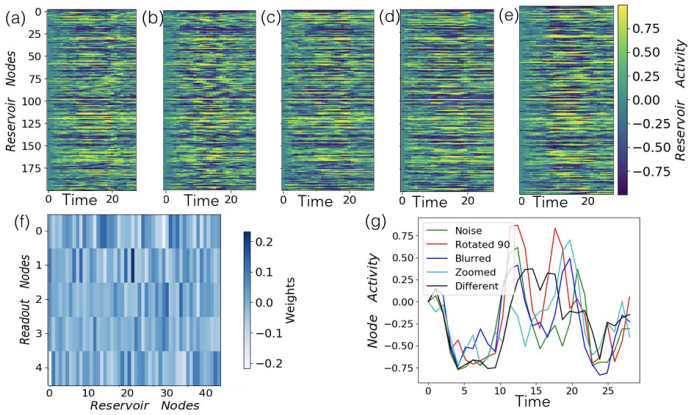


Figure 11: Reservoir activity for the single reservoir architecture. (a), (b), (c), (d), (e) show the differential reservoir activity of 200 nodes over 28 timesteps for input relationships noise, rotated, zoomed, blurred and different respectively. (f) shows the output weight matrix(W^{out}) for 50 reservoir nodes. (g) shows activity of a random node for all output labels over 28 timesteps. N_R: 1000, $\gamma = 0.5$, sparsity= 0.9.

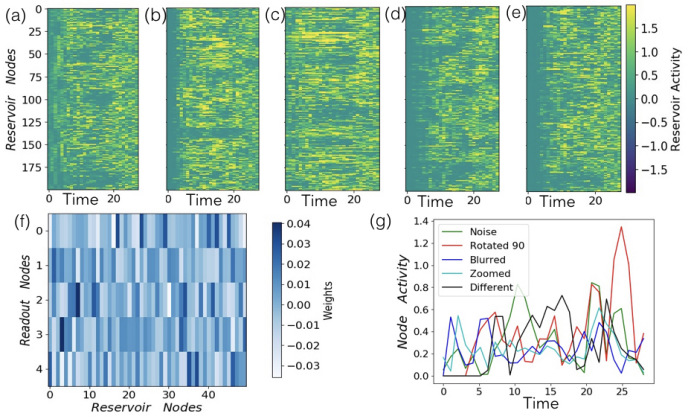


Figure 12: Reservoir activity for the dual reservoir architecture. (a), (b), (c), (d), (e) show the differential reservoir activity of 200 nodes over 28 timesteps for input relationships noise, rotated, zoomed, blurred and different respectively. (f) shows the output weight matrix(W^{out}) for 50 reservoir nodes. (g) shows activity of a random node for all output labels over 28 timesteps. N_R: 1000, $\gamma = 0.5$, sparsity= 0.9.

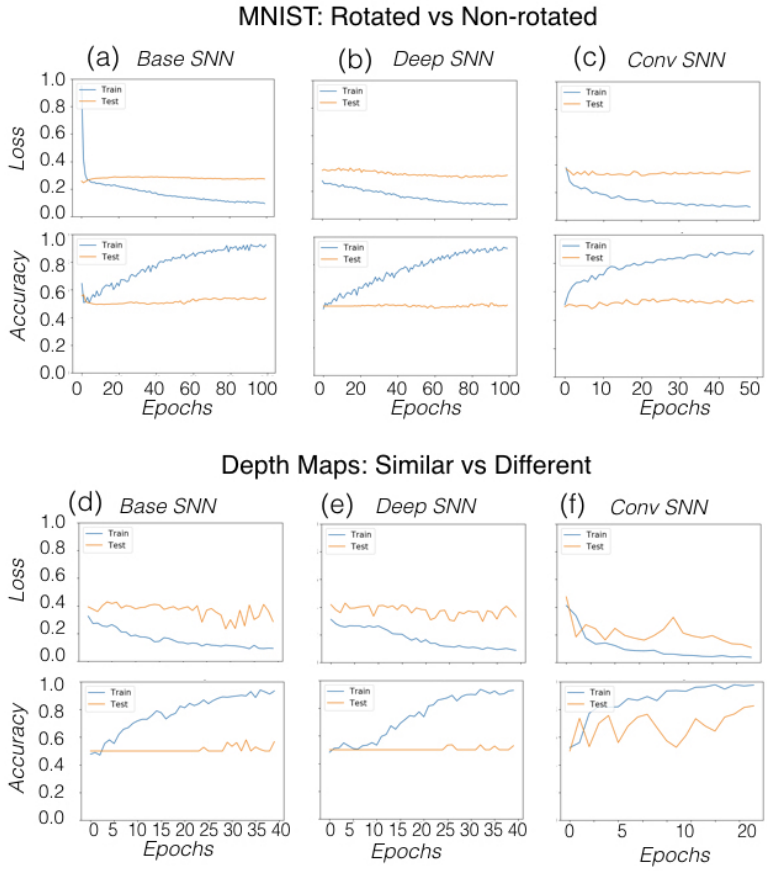


Figure 13: Plot of training loss and accuracy for (a&c) base siamese network, (b&d) deep siamese network, and (c&f) convolutional siamese network.

C Loss and Accuracy of SNNs

In Fig. 13 we plot the training loss and accuracy for the base SNN (4 layers), deep SNN (8 layers), and convolutional SNN for the two tasks of identifying rotation operator in MNIST and identifying similar visual scenes from a moving camera. Since training data is small, losses converge fairly quickly over epochs. The optimizer Adadelta, which employs a variable learning rate, was used in training.