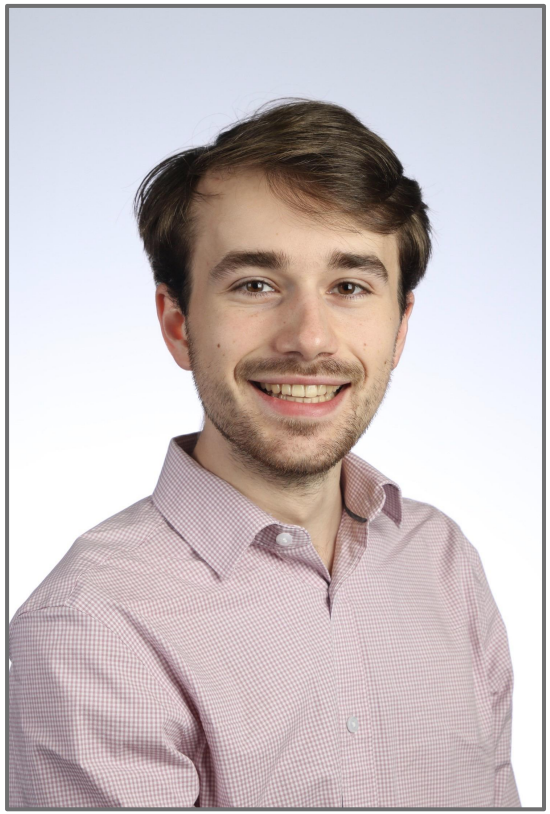
Applications and Performance of Echo State Networks

Jacob Pilawa, Uttam Paudel, Marta Luengo-Kovac, George Valley, Justin Shaw, Hannah Doyle, Matthew Ashner

6 August 2020

Who am I?

A little bit about me…



Jacob Pilawa

- PhD student in astrophysics, UC Berkeley
- 2nd summer as an Aerospace intern
- Lover of all things Cleveland

Academic Interests and Hobbies

- Observational astrophysics of galaxies
- Machine learning
- Astrophotography, skateboarding

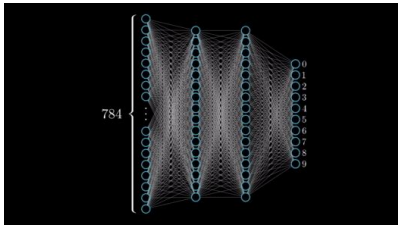
Time lapse of the Milky Way from Mt. Jelm, Wyoming.

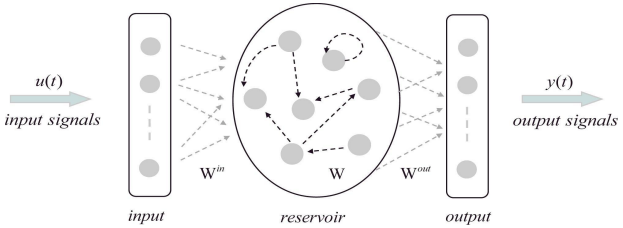
Overview

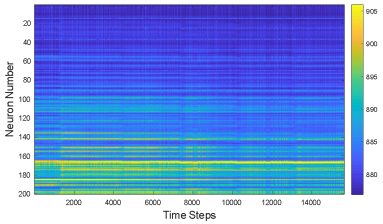
Background of Neural Networks

Echo State Networks/Reservoir Computing

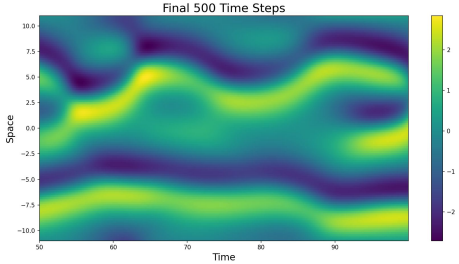
ESNs and their Successes



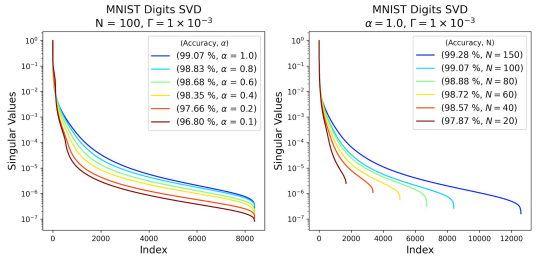




Additional Applications

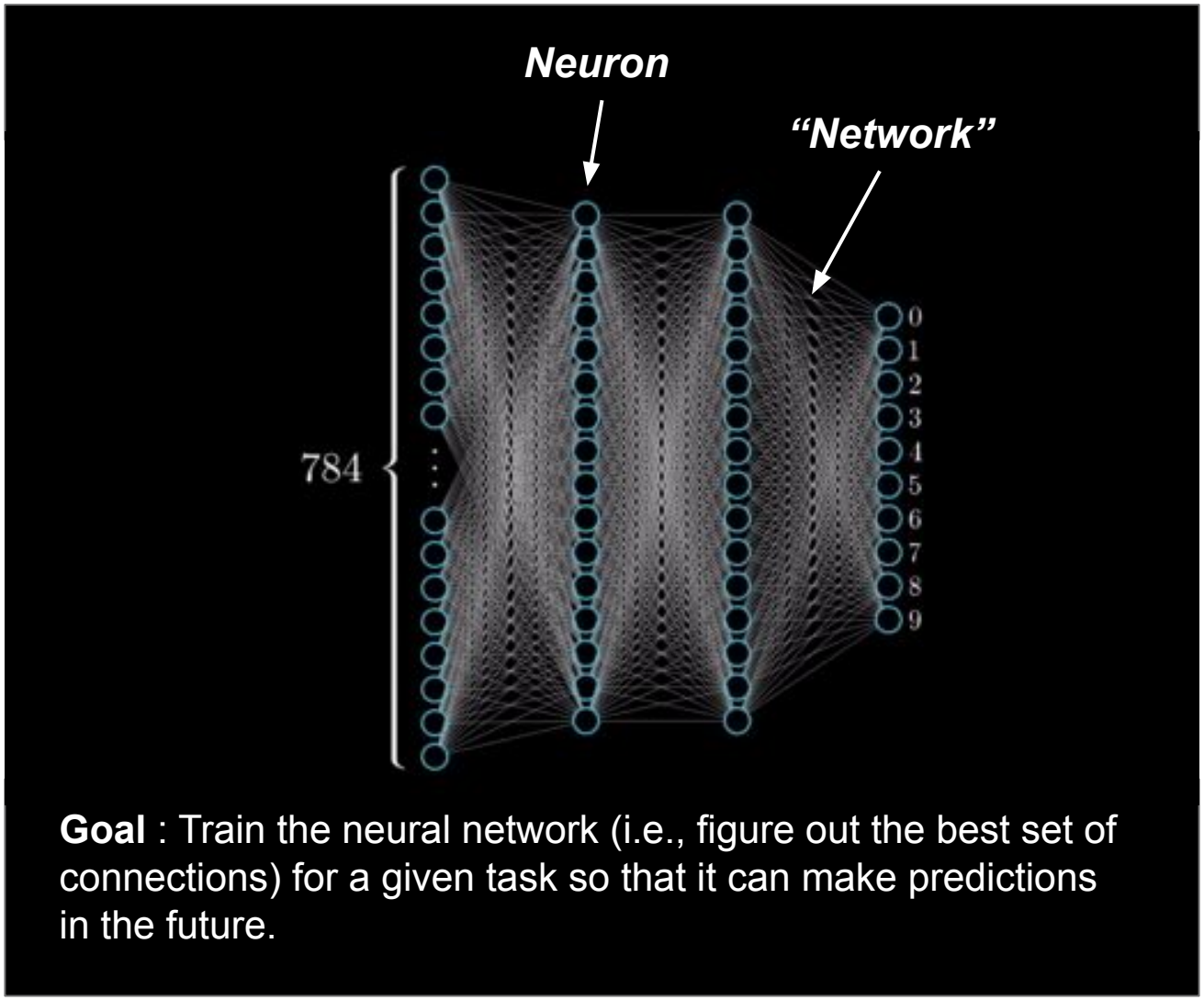


Evaluating ESN Performance



What's still to be done?

Neural Networks



Goal : Train the neural network (i.e., figure out the best set of connections) for a given task so that it can make predictions in the future.

Animation: 3Blue1Brown, YouTube

TAKEAWAY: *A neural network, plain and simply, is a function that takes inputs and maps them to a desired output after it has been "trained" to do so.*

Neural Networks

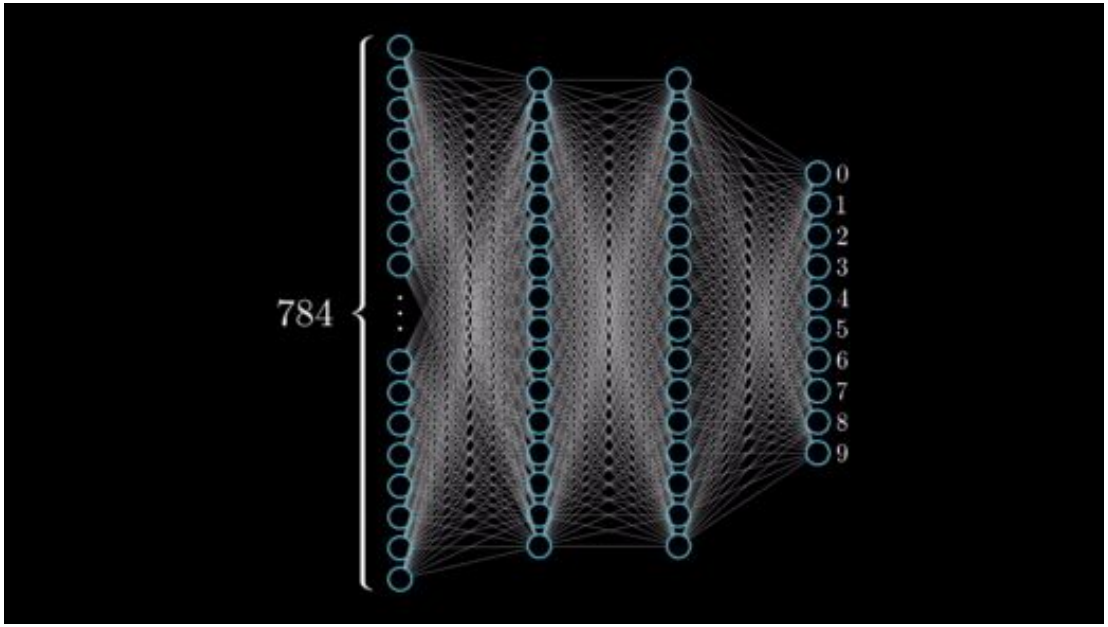
Advantages and Disadvantages

Advantages

- Complex, non-linear relationships
- Generalizable
- Adaptable -- new architectures

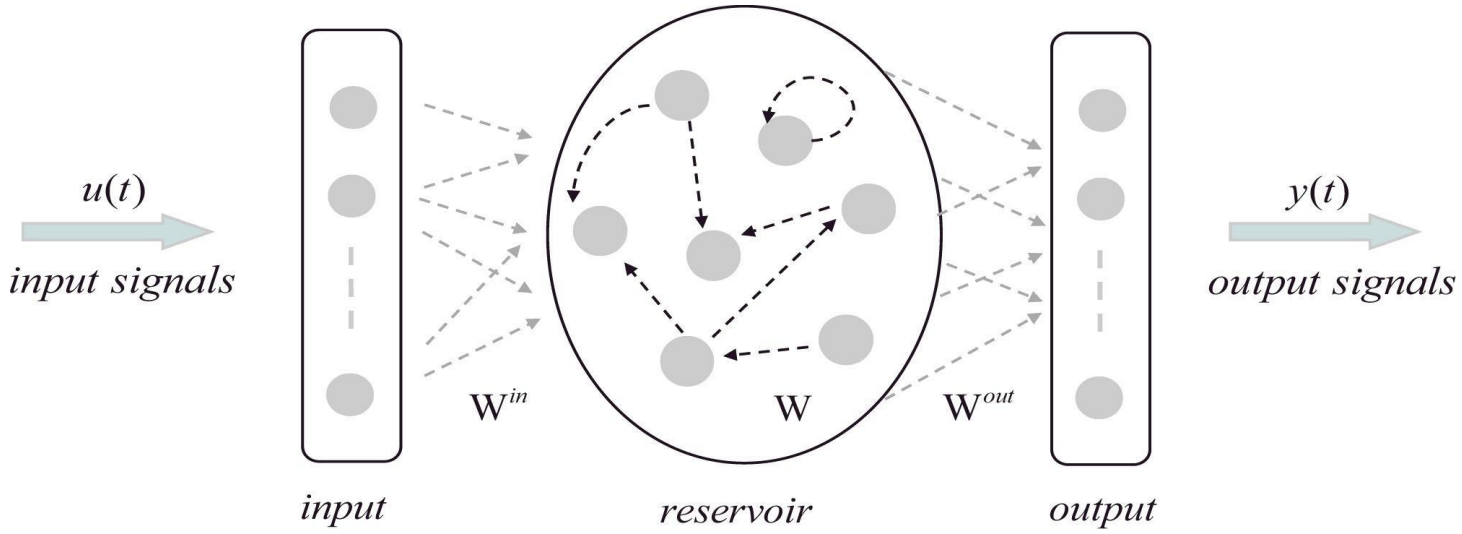
Disadvantages

- Training time
- Data set size
- No recurrent connections to allow for “memory” in time



Reservoir Computing / Echo State Networks

- Echo State Networks (ESNs) are a type of recurrent neural network
- **Key Differences**
  - Connections between neurons are **random** and **fixed** whereas traditional neural networks train all connections
  - Huge reservoir size + non-linear transfer function (connections) + Feedback connections = Captures non-linear, temporal dynamics of “real” systems
  - **Training is performed on a single layer of neurons using simple regression techniques (and thus computationally efficient)**

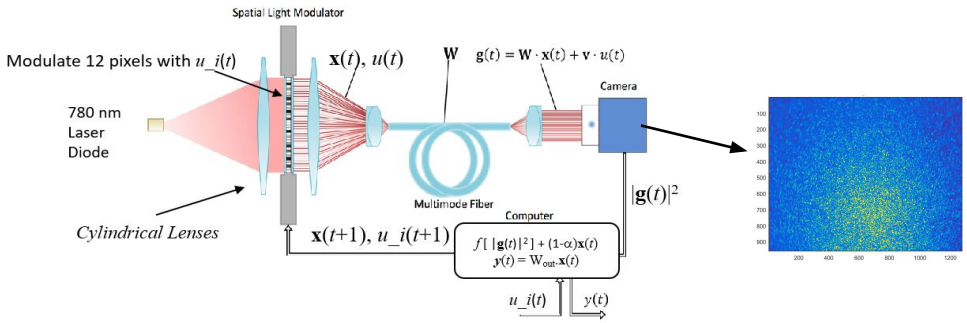


APPLICATIONS: *Speech recognition, time-series predictions, signal classification, denoising, demodulation, on-the-fly signals intelligence, at higher computational efficiency.*

Figure: Ma et al., 2016

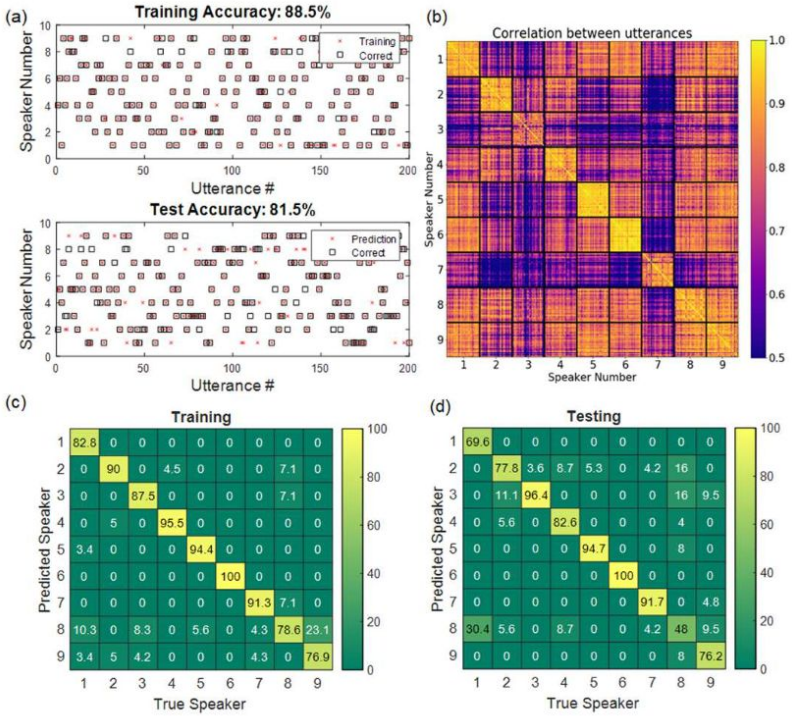
Last Summer's Work

Optical Echo State Network/Reservoir Computing Results

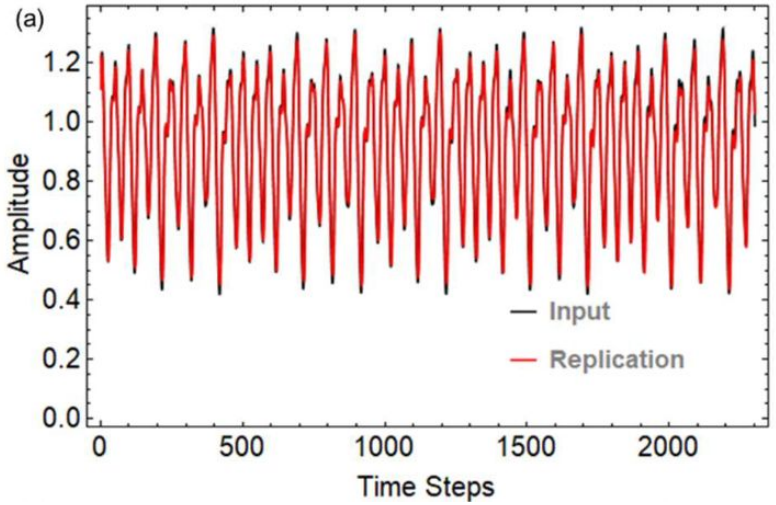


→ Input on SLM Output of SLM to MMF to make speckle
→Apply non-linearity and feedback to speckle
→ Reinject and Calculate result

Japanese Vowels

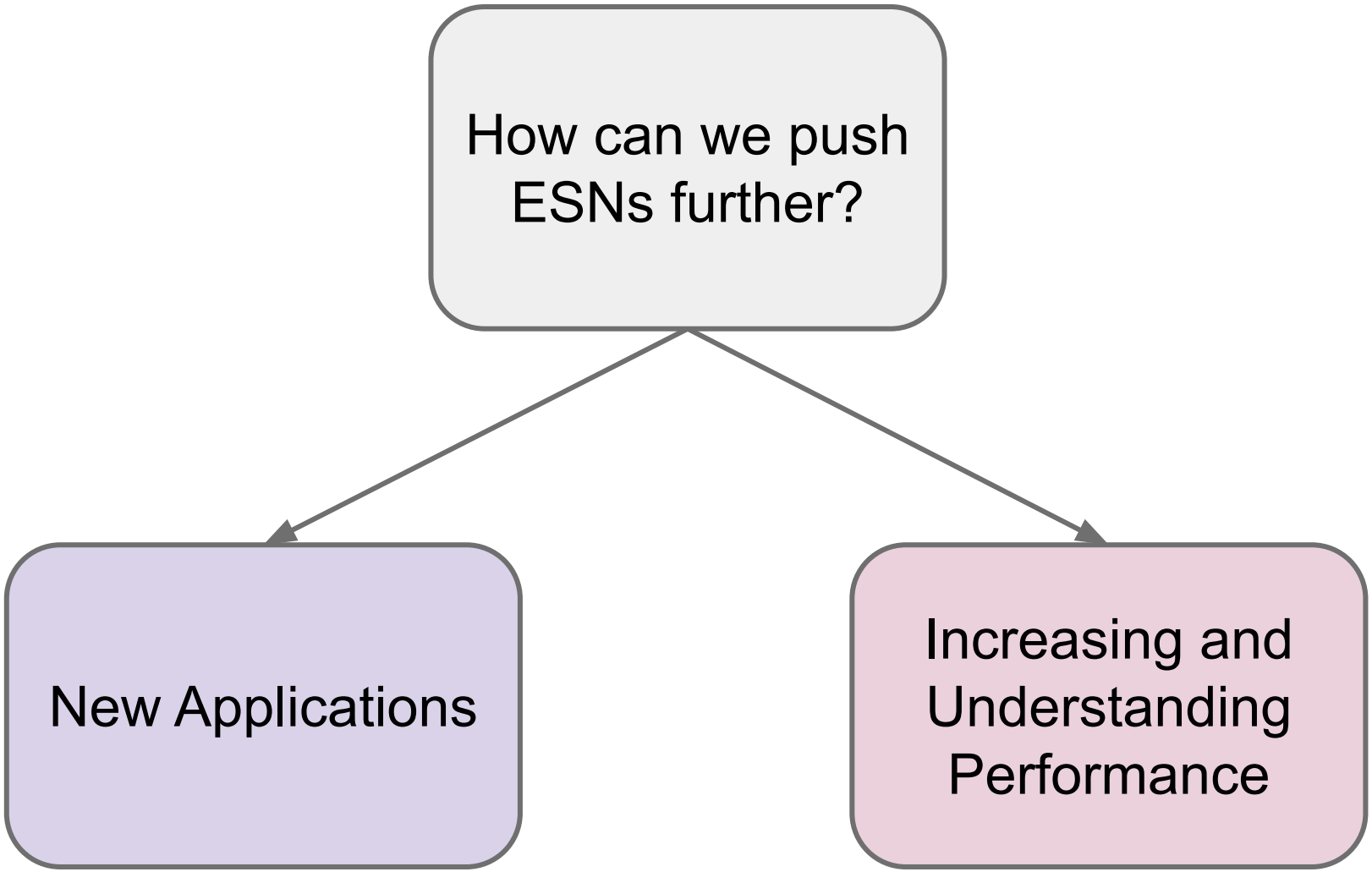


Mackey-Glass Time Series



Paudel, Luengo-Kovac, Pilawa, Shaw, and Valley 2019 (Optics Express)

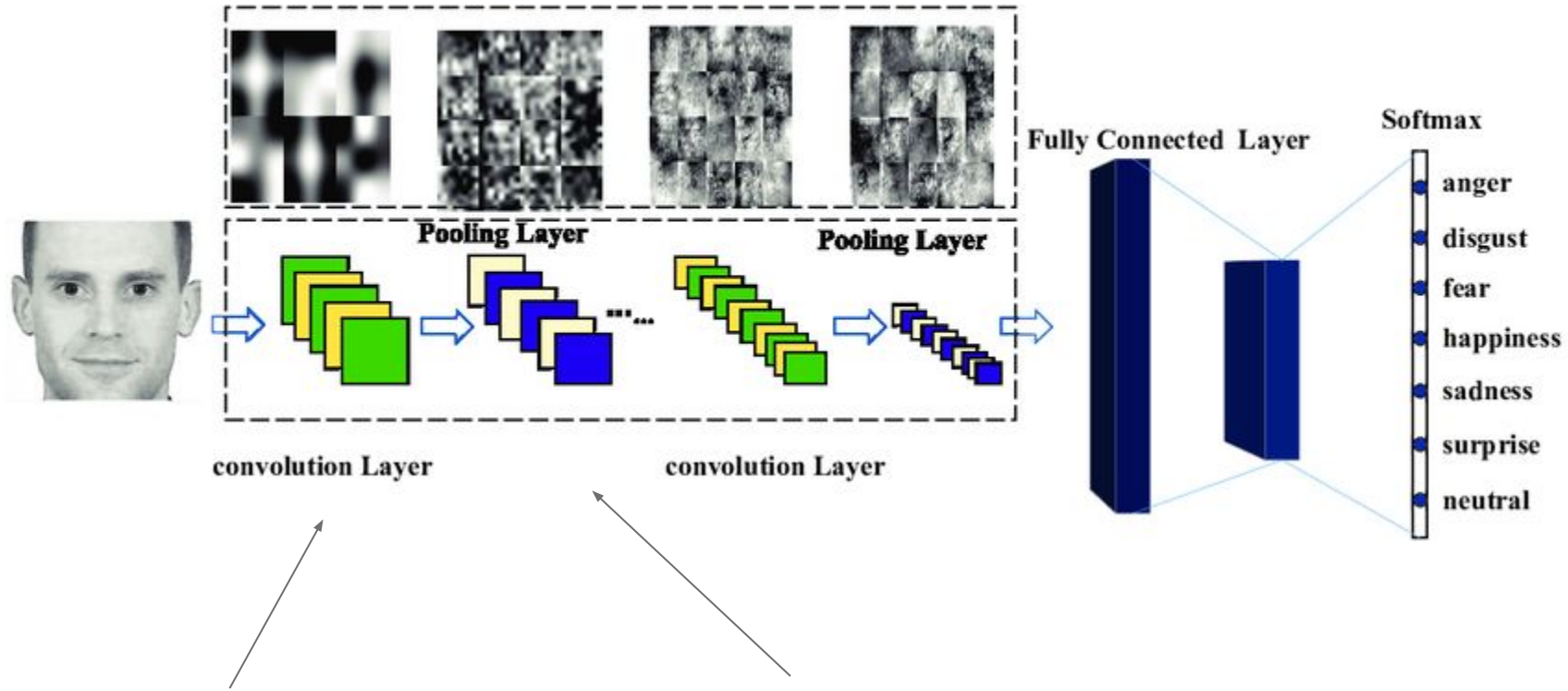
Summer 2020


How can we push ESNs further?
New Applications
Increasing and Understanding Performance

Application #1: Convolutional RCs

New architectures to deal with images

Convolutional Neural Networks



"Feature Extraction" Layer

"Feature Selection" Layer

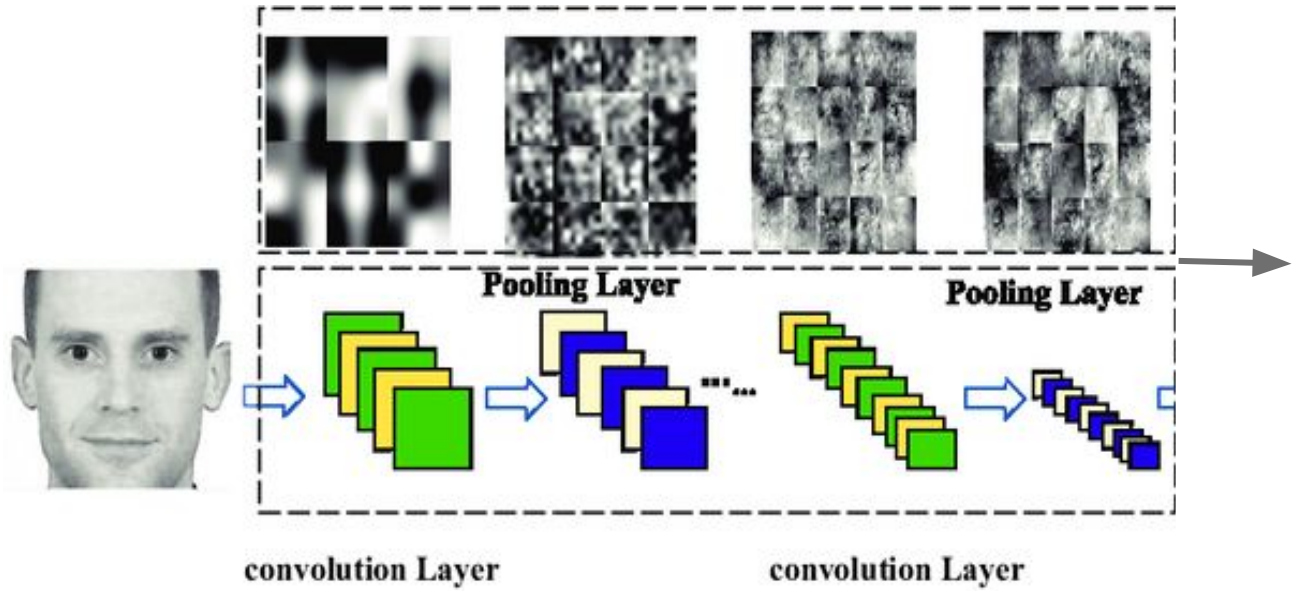
Classification Layers

Figure: Wang et al., 2019

Application #1: Convolutional RCs

New architectures to deal with images

Convolutional Neural Networks

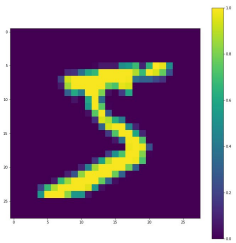


Echo State Network Classifier

APPLICATIONS: *Time domain image and video recognition and classification at low SWAP with optical convolutions. Specifics of this setup can be found in the backup slides if there are additional questions!*

Figure: Wang et al., 2019

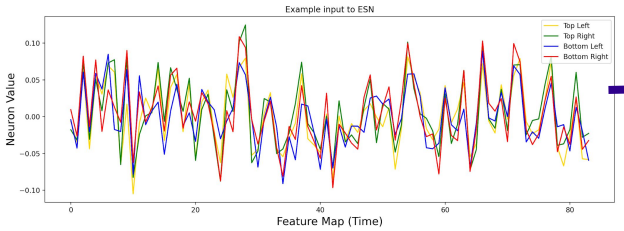
Block Diagram of Code

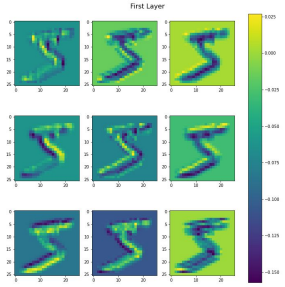


INPUT

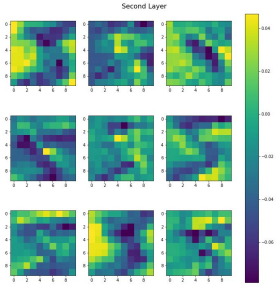
Convolutional Neural Network

Time Series to ESN

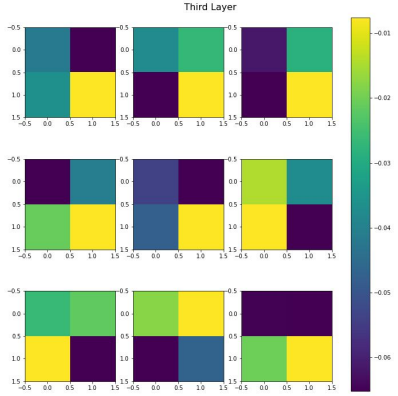




First Convolution Layer

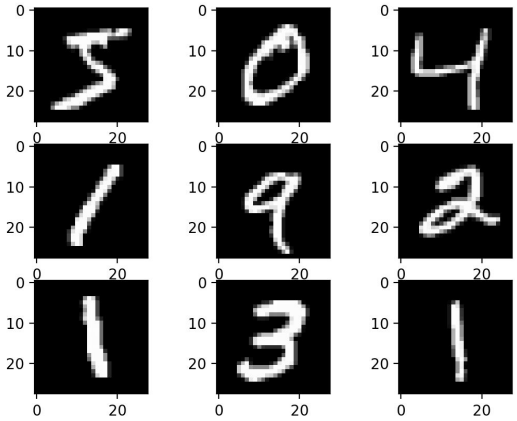


Second Convolution Layer

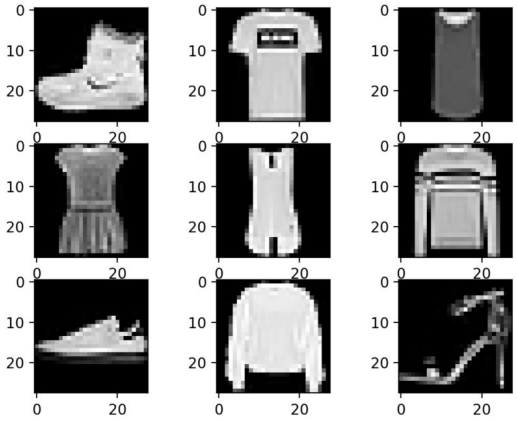


Third Convolution Layer

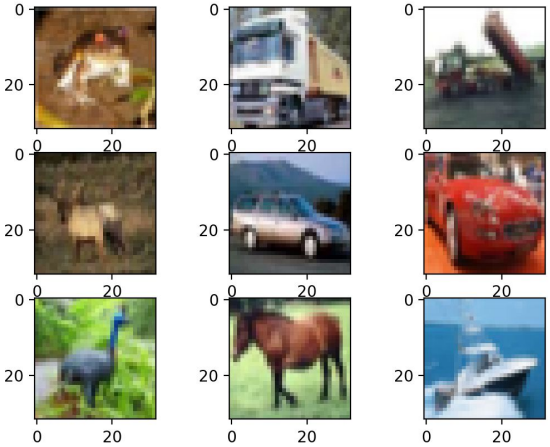
Benchmark Classification Tasks



MNIST Digits



MNIST Fashion

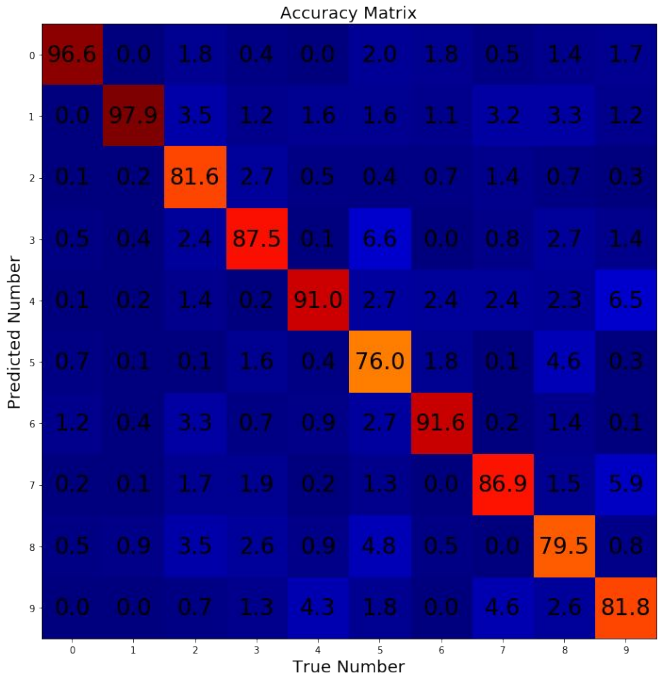


CIFAR10

Case Study #1

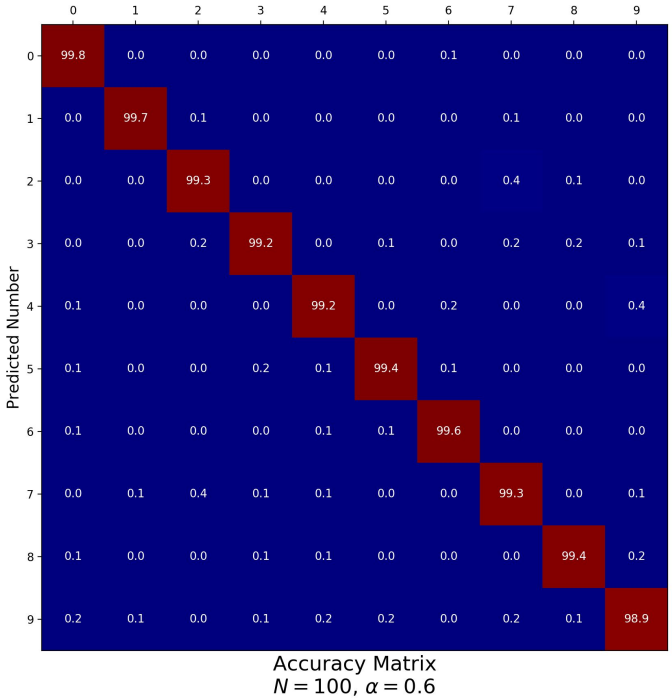
MNIST DIGITS Dataset (N = 100, α = 0.6, γ = 0.001)

ESN Alone



Error Rate: 12.74%
Time: 5.3 s

CNN + ESN



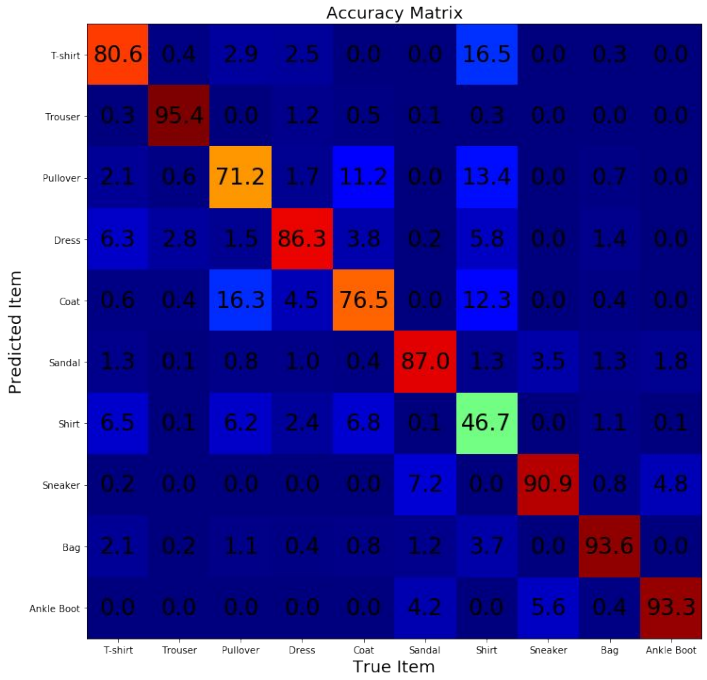
Error Rate: 0.63%
Time: 729.3 s

20x better performance

Case Study #2

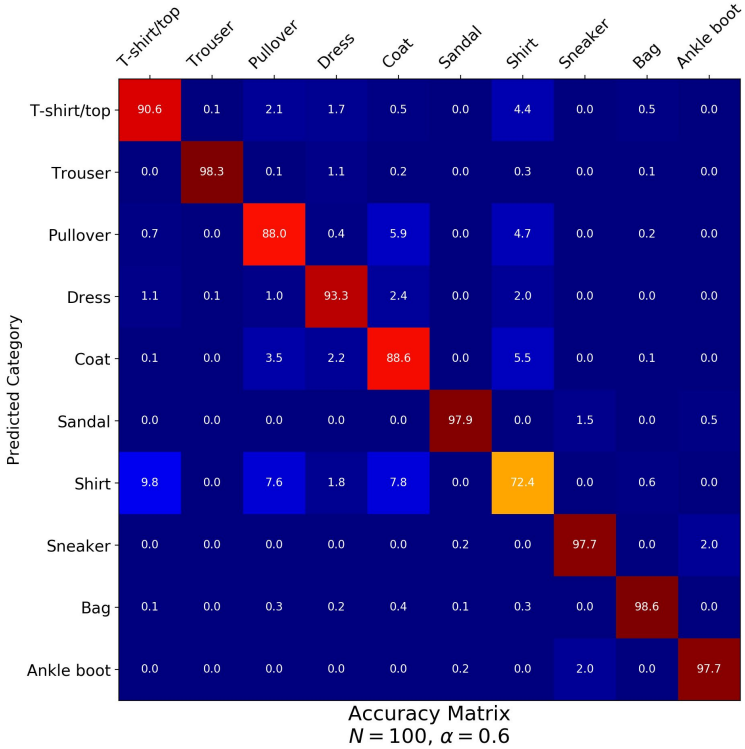
MNIST FASHION Dataset (N = 100, α = 0.6, γ = 0.001)

ESN Alone



Error Rate: 17.85%
Time: 5.3 s

CNN + ESN



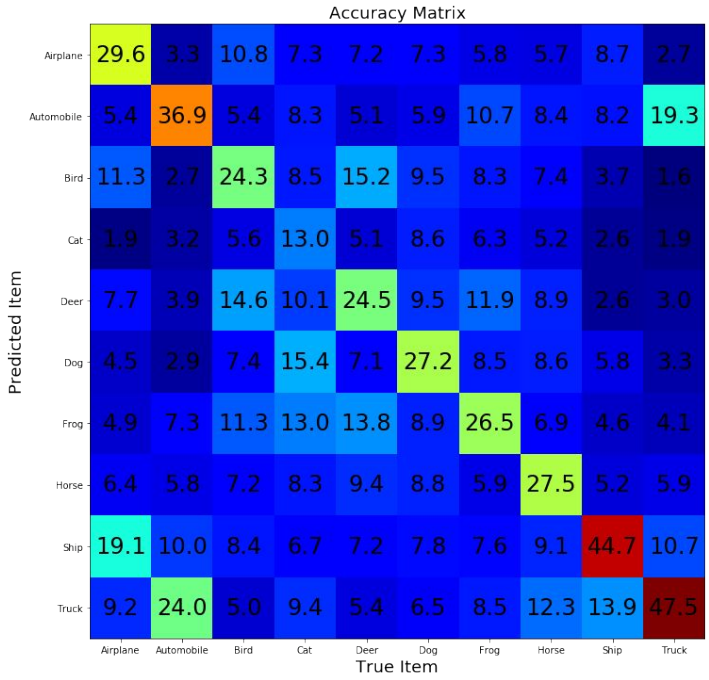
Error Rate: 7.68%
Time: 853.6 s

2.3x better performance

Case Study #3

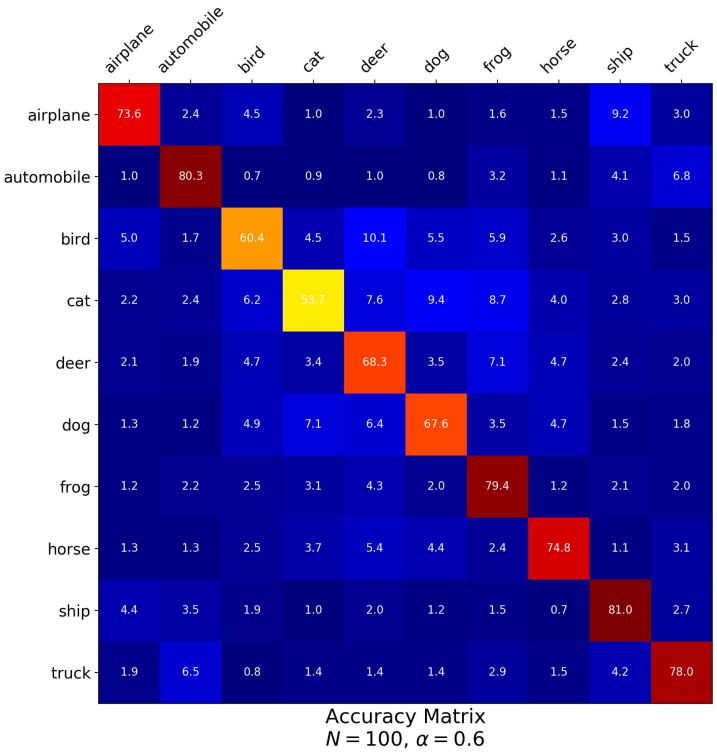
CIFAR10 Dataset (N = 100, α = 0.6, γ = 0.001)

ESN Alone



Accuracy: 30.17%
Time: 6.7 s

CNN + ESN

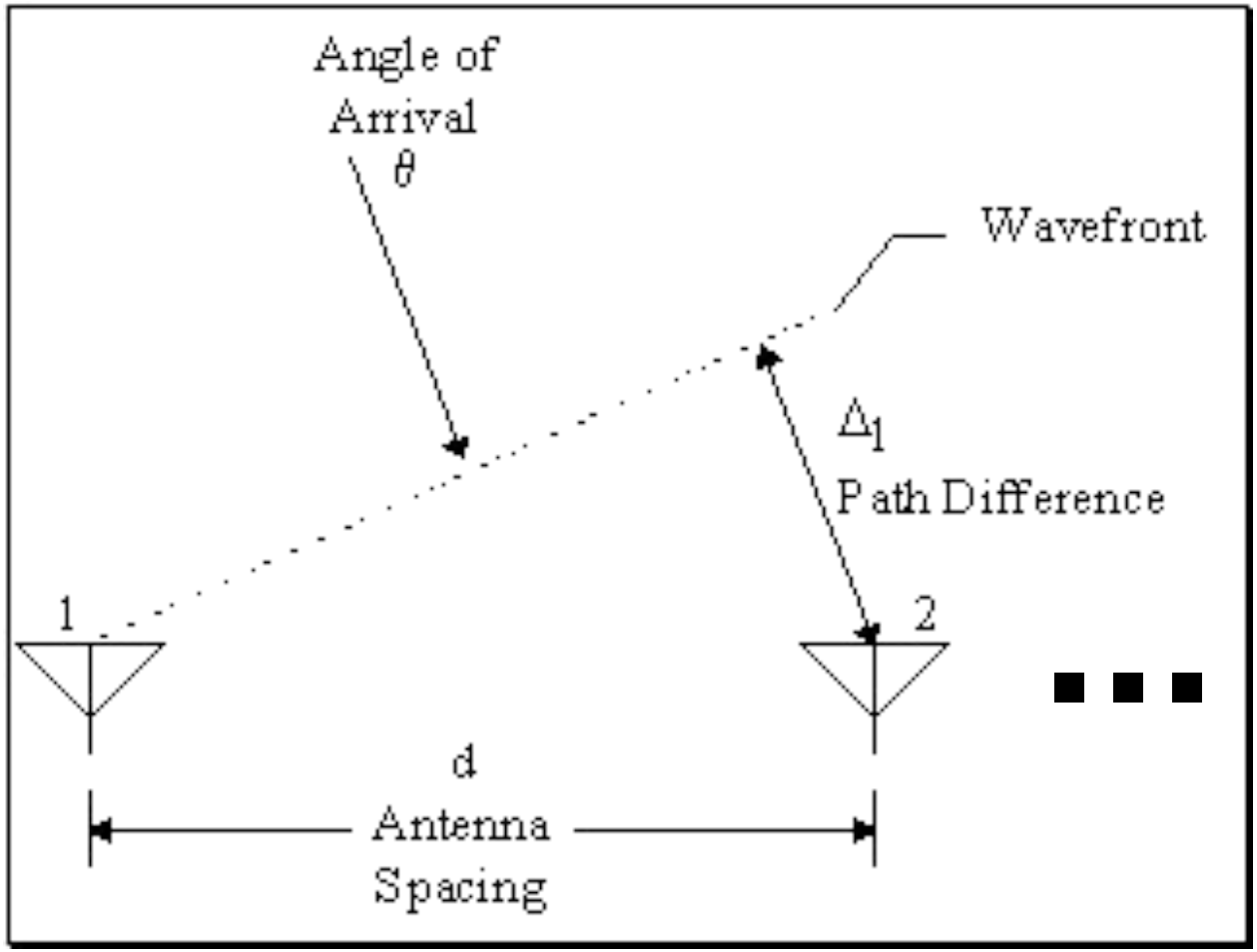


Accuracy: 71.7%
Time: 695.8 s

Accuracy: +41.53%

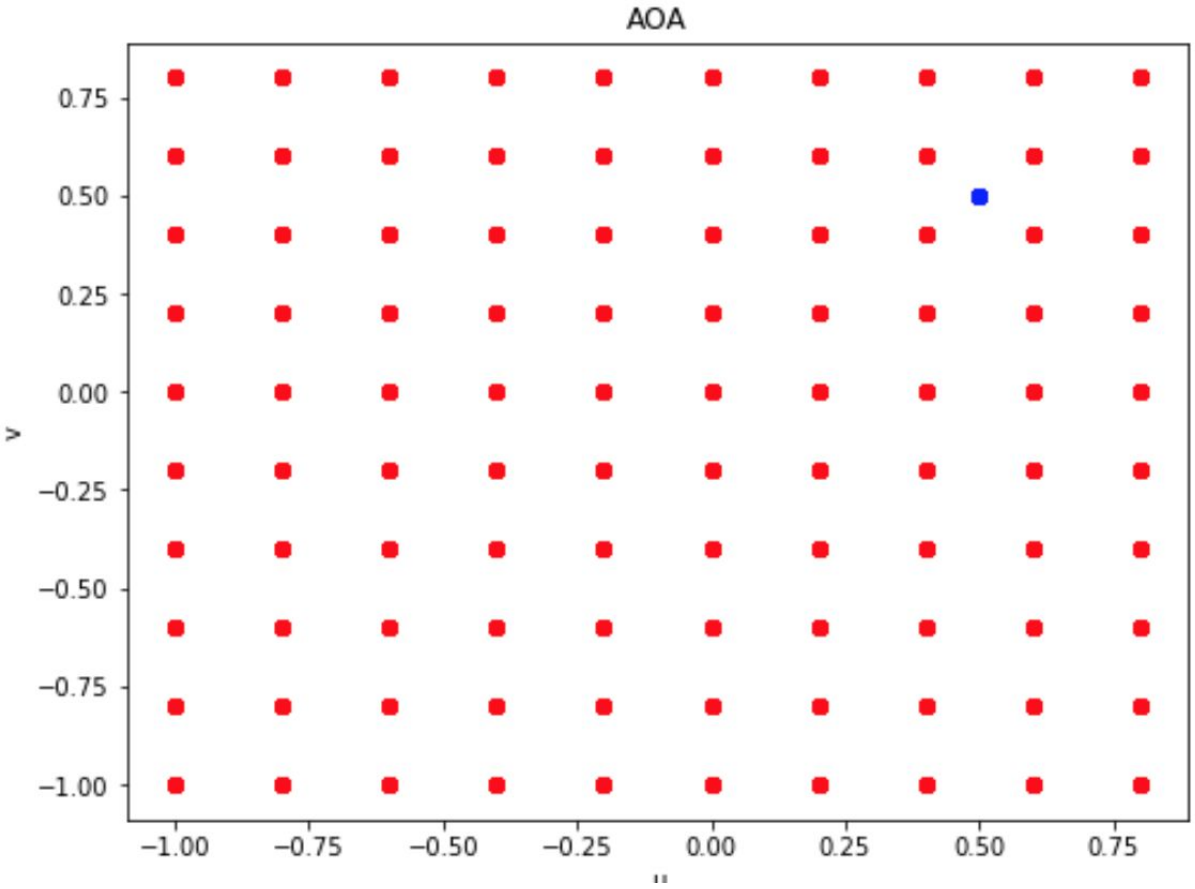
Application #2: Phased Array

Given an array of receivers, can we use a neural network to predict angles of arrival?



Application #2: Phased Array

Given an array of receivers, can we use a neural network to predict angles of arrival?



Train with Red Points (on grid pulses)

Test with Blue Point (off grid pulse)

Input: Simulated signal (with random amplitude and phase) from 11x11 grid of antennae.

Goal: Predict AOA from signal.

Adjusted Update Equation

$$x(t+1) = (1-\alpha)\,x(t) + \alpha \tanh\left(Wx(t) + W_{in}u(t)\right)$$

$$Wx(t) \rightarrow \begin{cases} \text{cycle } x(t) \text{ inside and outside non-linearity} \\ \text{random shuffle } x(t) \text{ inside and outside} \\ \text{relatively prime cycle } x(t) \\ \text{no transformation} \end{cases}$$

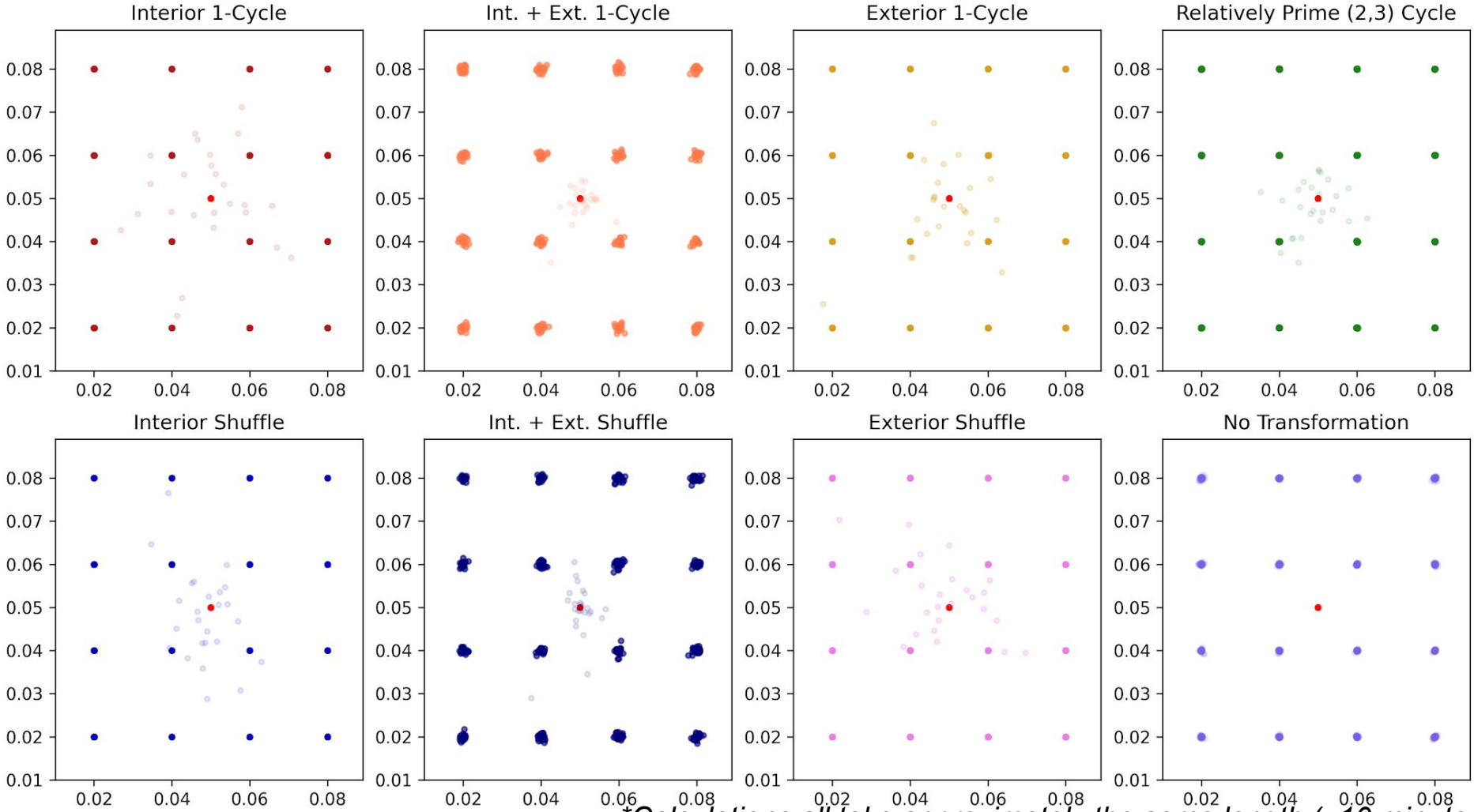
Inside and Outside Cycle, for example:

$$g(x,u) = (1-\alpha)x^{\circlearrowright} + \alpha \tanh\left(x^{\circlearrowright} + W_{in}u\right)$$

Predictions

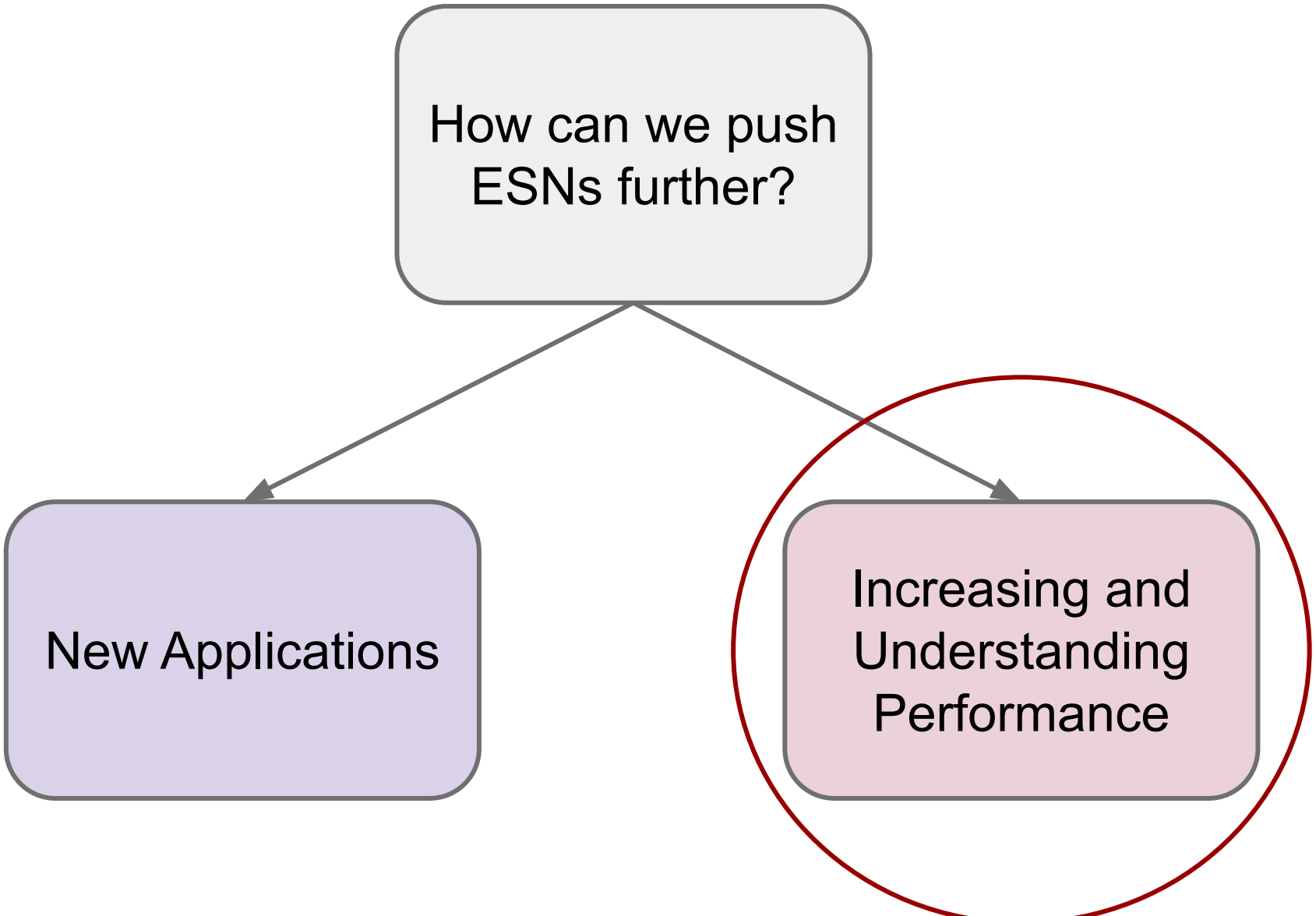
Note: *N = 5000, a = 0.6, No Reg.*

- ***Interior + Exterior 1-Cycle*** *and* ***Random Shuffle*** *perform better than the others.*



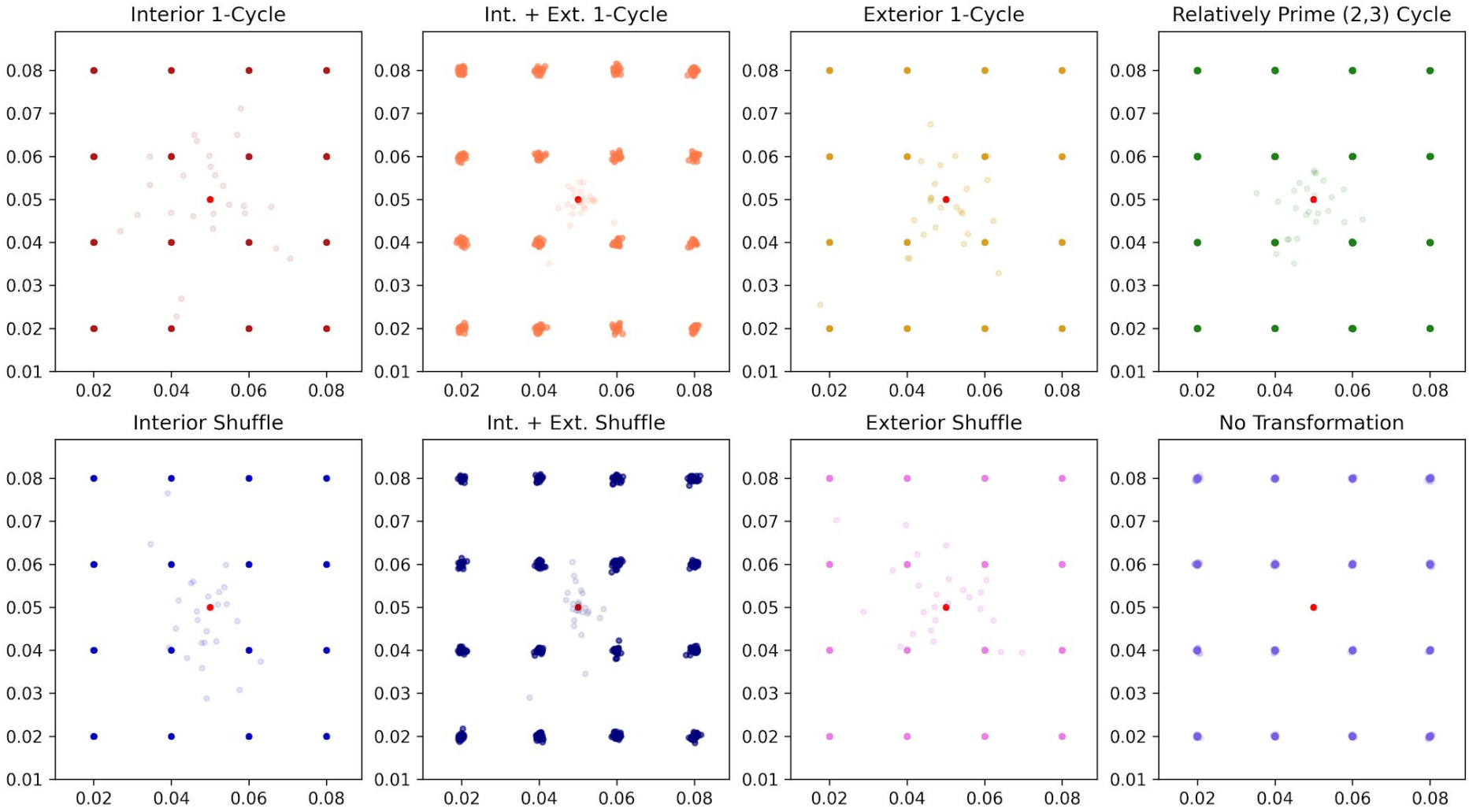
**Calculations all take approximately the same length (~10 minutes).*

Summer 2020


How can we push ESNs further?
New Applications
Increasing and Understanding Performance

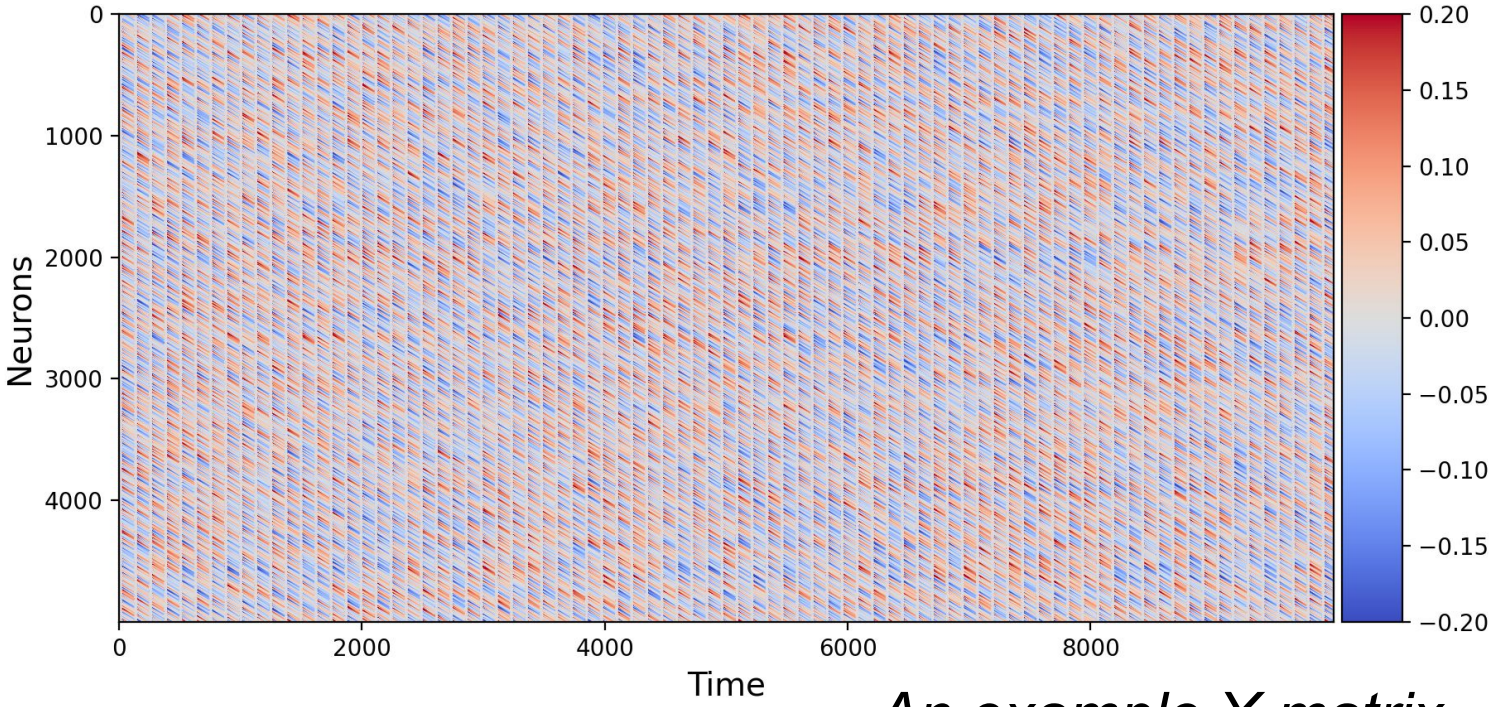
Phased Array Predictions

What is actually causing the echo state network to have better performance?



X and X^2 Matrices

- *Part of ESN calculations is to store the "neuron vector" at each time step → neuron matrix.*

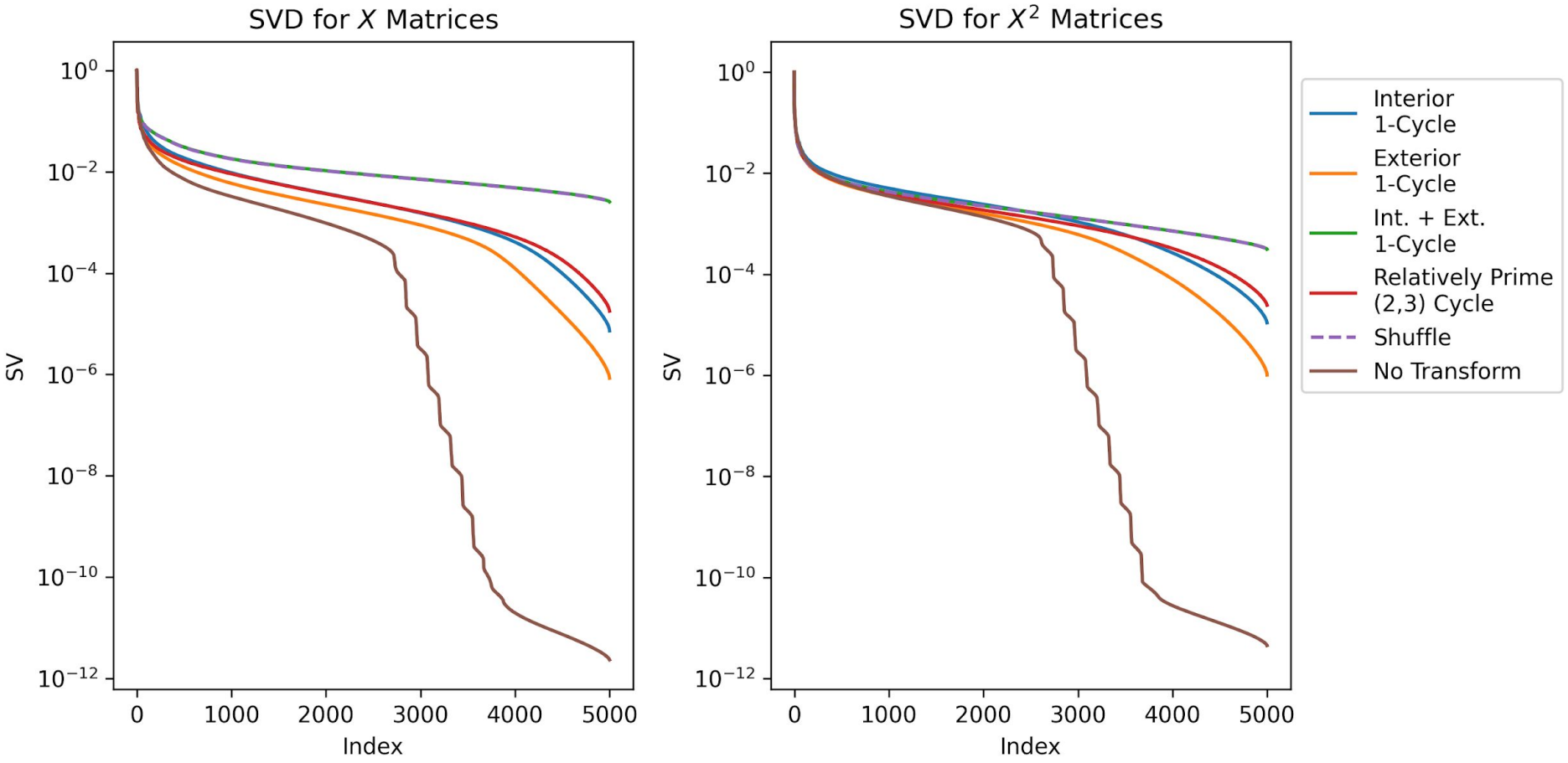


An example X matrix.

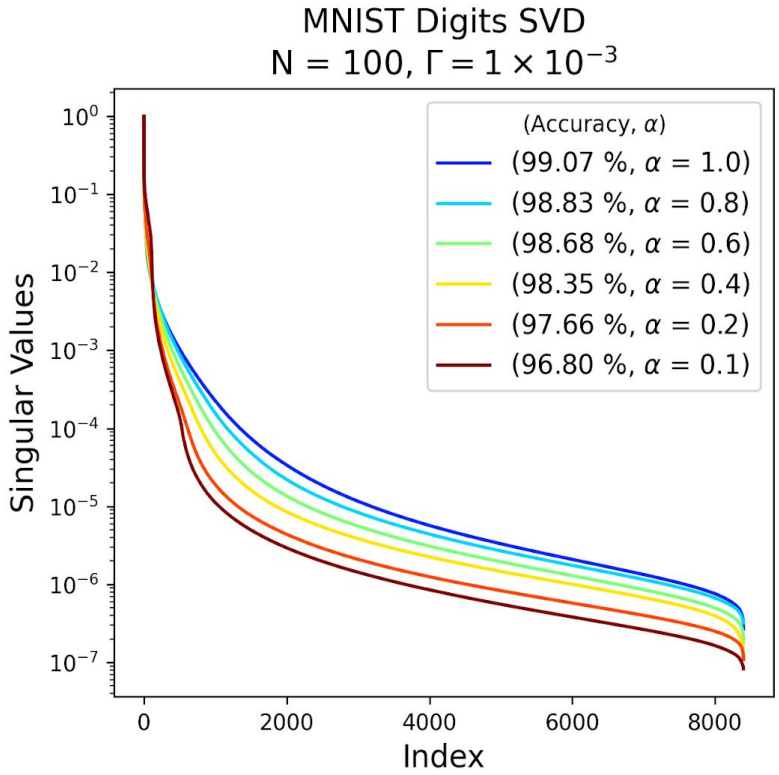
Takeaway: *We can think of the X matrix as storing the information of the calculation. We want to maximize this information carrying capacity.*

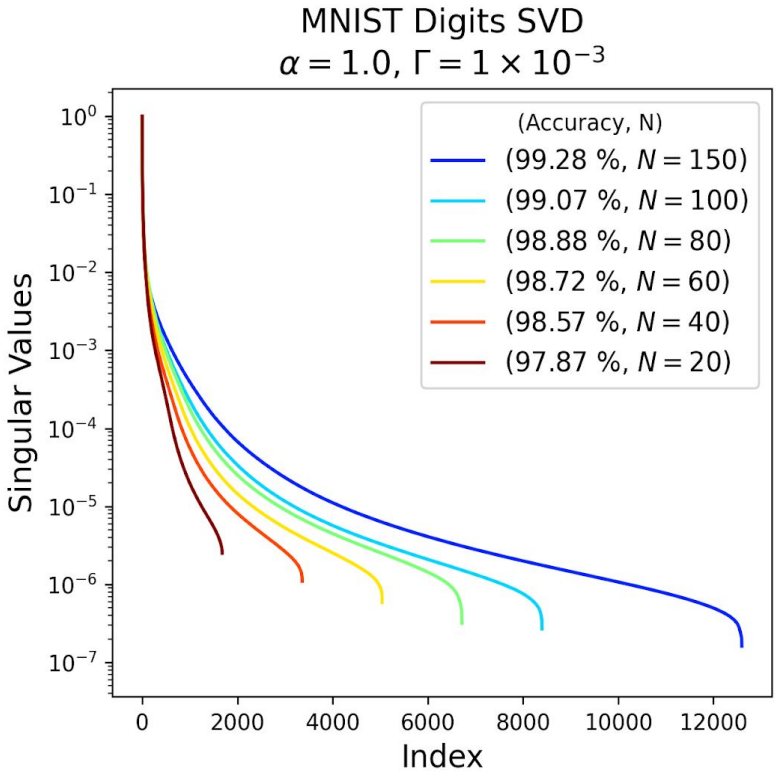
SVD for X and X^2 Matrices

- ***Interior + Exterior 1-Cycle** and **Random Shuffle** SVD values are nearly identical. The two are indistinguishable on both plots.*
- *Correlation between minimum singular value and performance.*

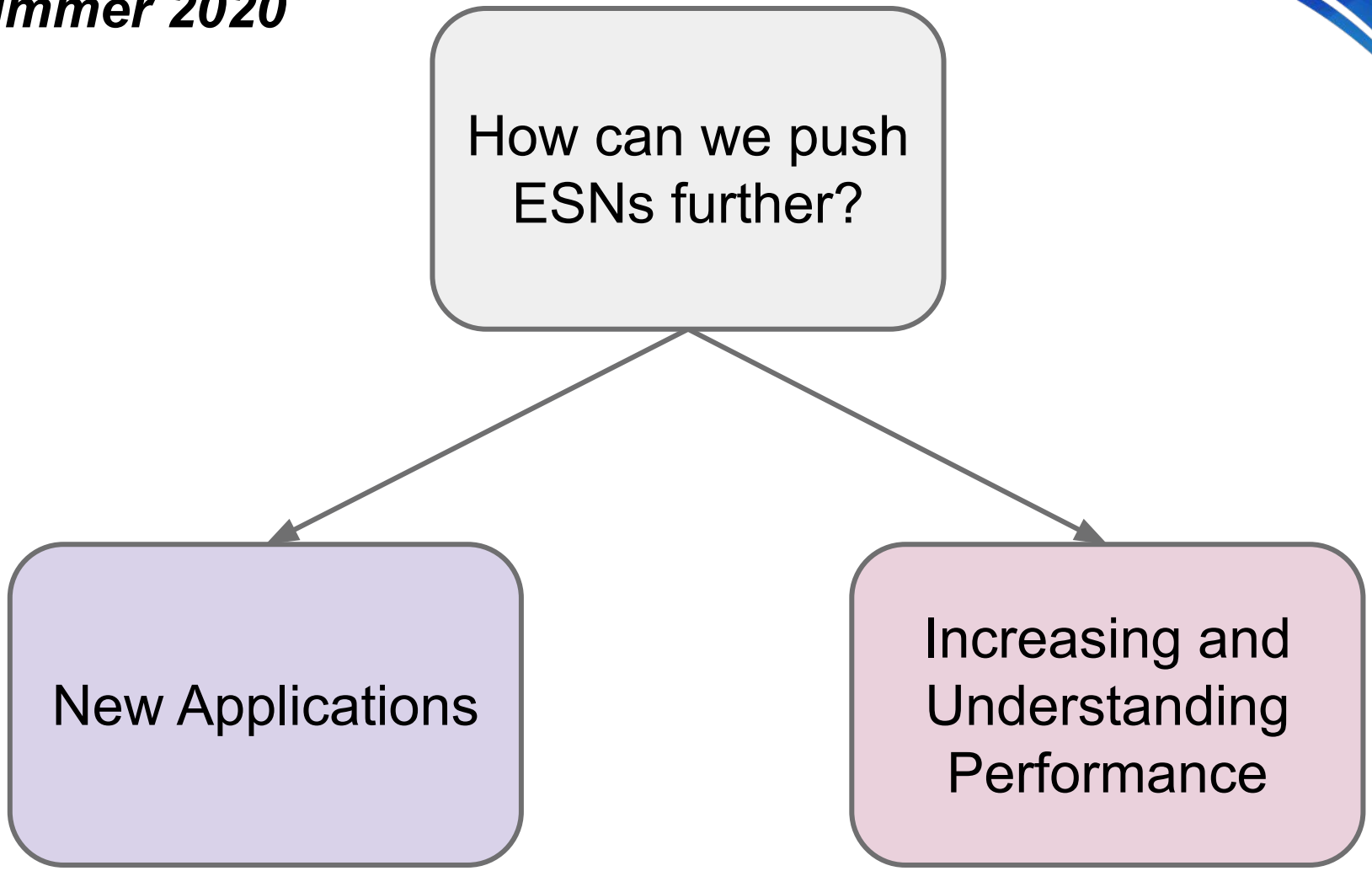


MNIST Digits SVD





Minimal singular values do seem to correspond to higher performance from ESNs.


How can we push ESNs further?
New Applications
Increasing and Understanding Performance

So … now what?

New Application Recap:

Summer 2020 Work	Questions for the Future
New architectures like CNN to RC system.	***How might we implement some of the new architectures optically?***
Calculations on efficiency of optical computer versus GPUs.	***Is the SWAP of large scale optical computing actually lower than traditional computing?****
Exploration of new problems like image classification.	***How can we find what else ESNs are good at?***

Increasing Performance Recap:

Summer 2020 Work	Questions for the Future
Using new ESN update equations.	***Can we understand why/if these increase performance?***
Exploring SVDs as a probe for ESN performance.	***Can we utilize this to increase training and testing time on our neural networks?***

Thank you!

A special thanks to Uttam Paudel, Marta Luengo-Kovac, George Valley, Justin Shaw, Hannah Doyle, and Matthew Ashner for their support.

Backup Slides/Extra Information

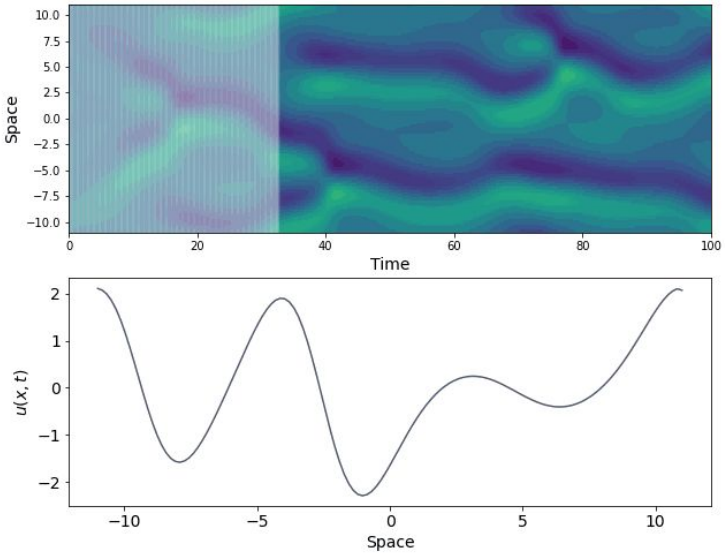
(Thanks to Uttam Paudel & Marta Luengo-Kovac)

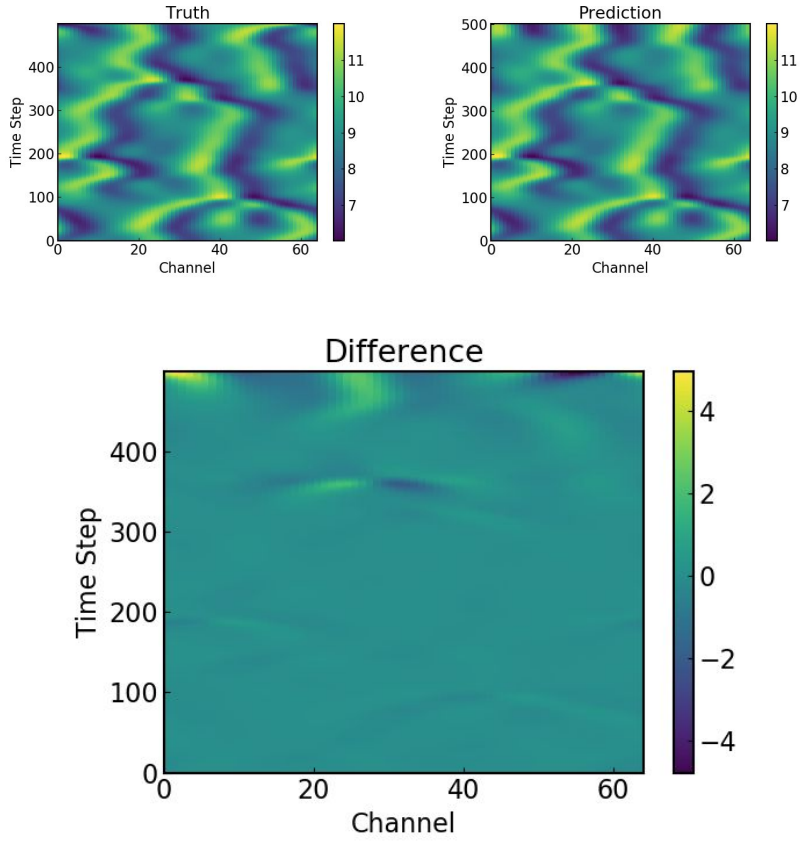
Summer 2020 Slides

Application #2: Kuramoto-Sivashinsky Equation

Simulated Fiber Ring Resonator + ESN by Matthew Ashner

$$u_t = -uu_x - u_{xx} - u_{xxxx}$$





Wrote the code to solve this.

Results from M. Ashner

Background

Processing Speeds and Power Consumption

Typical GPU
20 pJ/MAC

State of the Art ASIC
1 pJ/MAC

Figure 5: Typical benchmarks for the OPU, in time and energy, on a whole Transfer Learning training task. Starting from a pre-trained VGG16 Deep Neural Network, the task is to specify the classification layers to the STL10 database.

1 MAC:

a → a + (b x c)

LightOn OPU
$3x10^{15}$ MAC/s at 30 W → 0.01 pJ/MAC

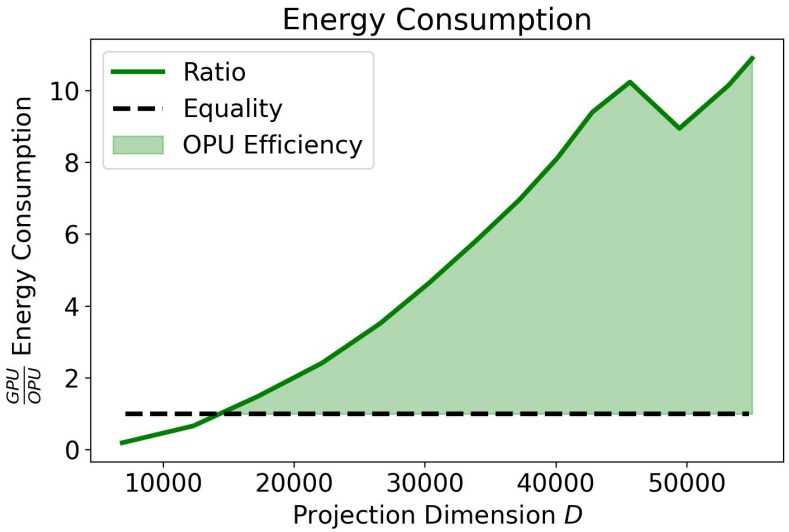
Our Setup
$2.1x10^{15}$ MAC/s at 30 W → 0.014 pJ/MAC

In theory, our setup should be approximately equivalent to LightOn's OPU setup.

OPU vs. GPU Claims

Trained on Fashion-MNIST Data

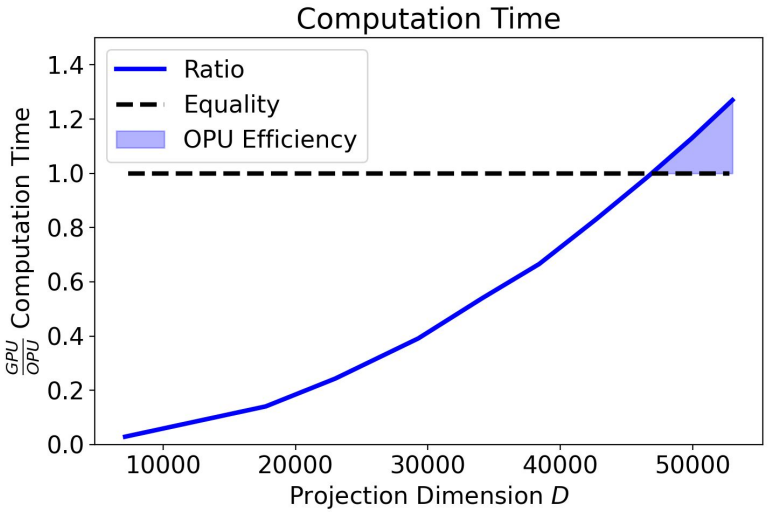
- Computation time using optical processing is faster for higher dimensional data
- Little to no loss of accuracy with appropriate corrections (non-negative kernels to mimic light intensity, for example)
- Energy consumption is smaller for higher dimensional data



Energy spent on computing a matrix multiplication (n,D) × (D,D). The batch size n is 3000 (solid line). The OPU is compared to an NVIDIA P100 GPU. Note that LightOn still does computations electronically, but convolution is done optically.

OPU vs. GPU Claims

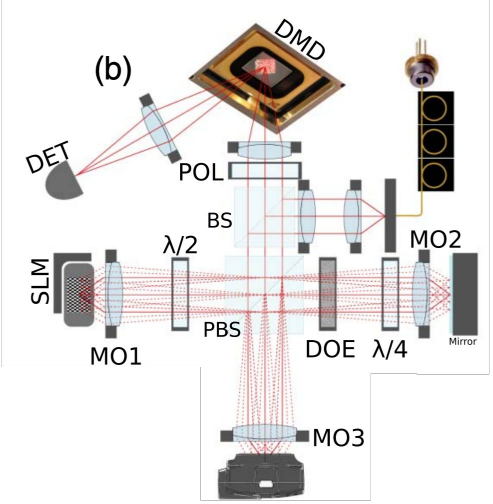
- LightOn claims 9x speedup using 1 OPU and 1 GPU over GPU-only for action recognition in videos
- For ImageNet dataset, OPU/GPU system is 8x faster and reduces energy cost by 20x
- Performance (*$3x10^{15}$ MAC/s*) improved by 2-3 orders of magnitude over pure silicon AI chips, e.g. Google's TPU



Time spent on computing a matrix multiplication (n,D) × (D,D). The batch size n is 3000 (solid line). The OPU is compared to an NVIDIA P100 GPU. Note that LightOn still does computations electronically, but convolution is done optically.

Photonic Recurrent Neural Network

- Input state encoded using an SLM, weights of DMD adjusted during training
- Train on the first 200 points of the Mackey-Glass sequence to predict the following points
- Once DMD weights are trained, setup is passive and energy efficient - mirrors remain in fixed positions
- System speed limited by 5 Hz update rate of SLM



Bueno et al, "Reinforcement learning in a large-scale photonic recurrent neural network." (2018)

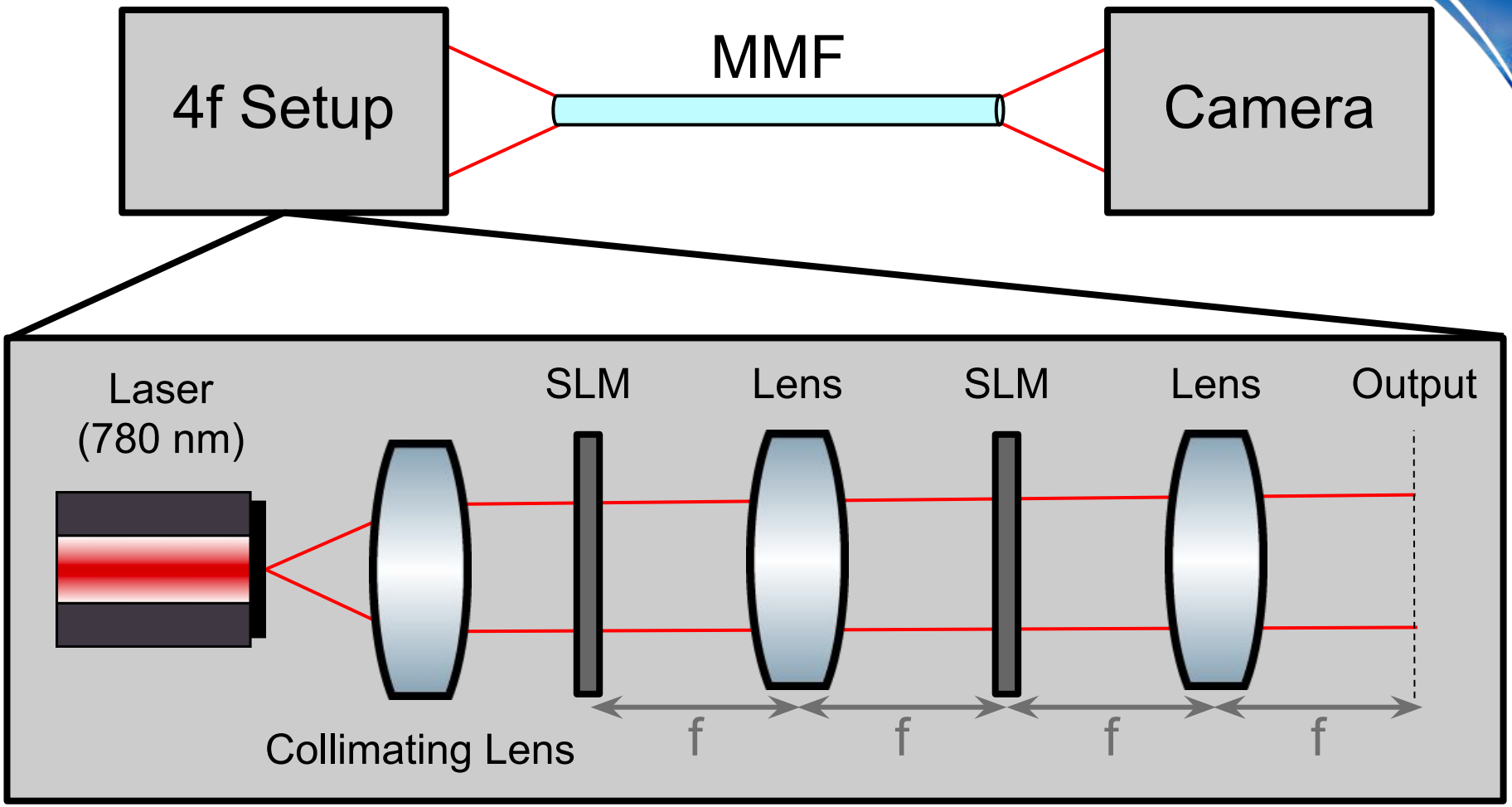
Hybrid Optoelectronic CNN

- Report on CNN with convolution implemented optically via 4f setup, fed into fully connected digital layers
- Image classification on grayscale CIFAR-10 dataset - 60,000 32x32 images in 10 classes
- Using an optical convolutional layer greatly reduces number of FLOPs

Method	***Accuracy***	***FLOPs***
Digital Only	*51.9 ± 1.3%*	*1,490,954*
Opt-Conv - Simulation	*51.0 ± 1.4%*	*3,779,136/180,234*
Opt-Conv - Experiment	*44.4%*	*180,234*

Chang et al, "Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification." (2018)

Our Setup



- Convolutional pre-processing and RC both implemented optically
- Each SLM has 1920 x 1152 pixels, 422.4 Hz frame rate

  → *(1920 x 1152)^2 x 422.4 Hz* = ***2.1 x 10^{15} MAC/s***

Optical Convolution

1. First lens performs a Fourier Transform on f(x,y) to yield F(p,q)

$$F(p,q) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\, e^{i(px+qy)}\, \mathrm{d}x\, \mathrm{d}y$$

2. Phase mask placed one focal length behind lens does a multiplication H(p,q)F(p,q)

3. Result passes through second lens, which performs a Fourier Transform on H(p,q)F(p,q)

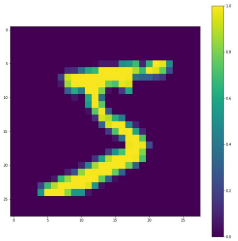
Overall result is the convolution of f(x,y) and h(x,y), by the convolution theorem:

$$\mathcal{F}\{f * h\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{h\}$$

Lugt, A. Vander, and E. N. Leith. "Techniques in optical data processing and coherent optics." (1969).

Convolution Schematic

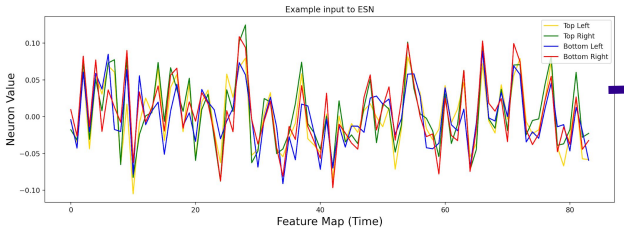
Block Diagram of Architecture

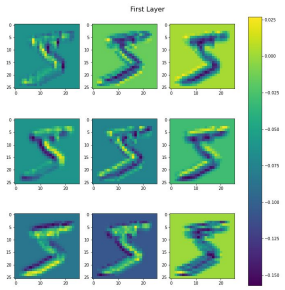


INPUT (SLM)

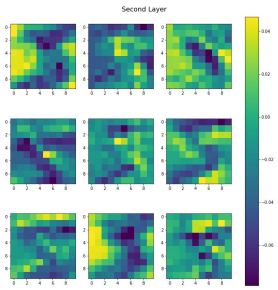
4f Convolution Setup

Time Series to ESN

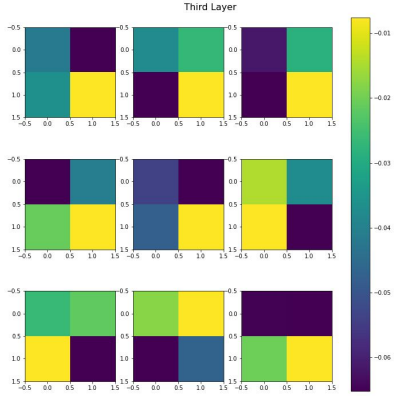




First Convolution Layer



Second Convolution Layer

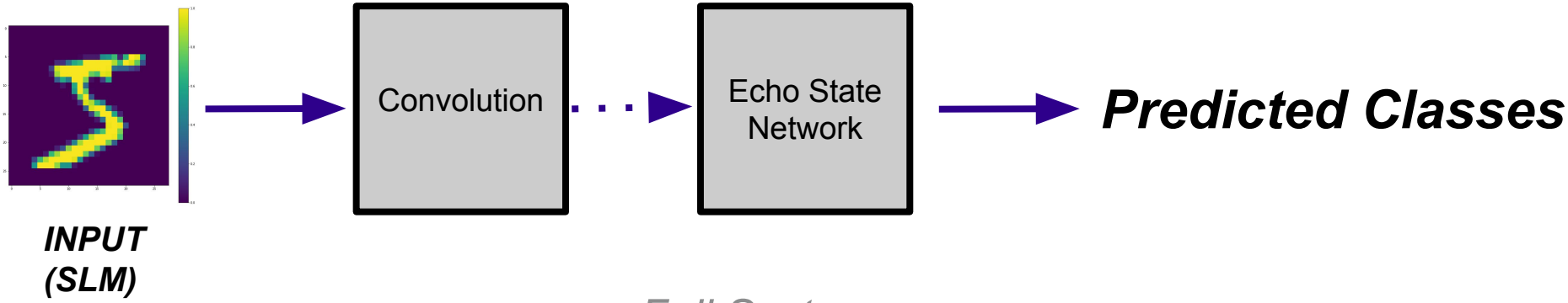


Third Convolution Layer

Explanation

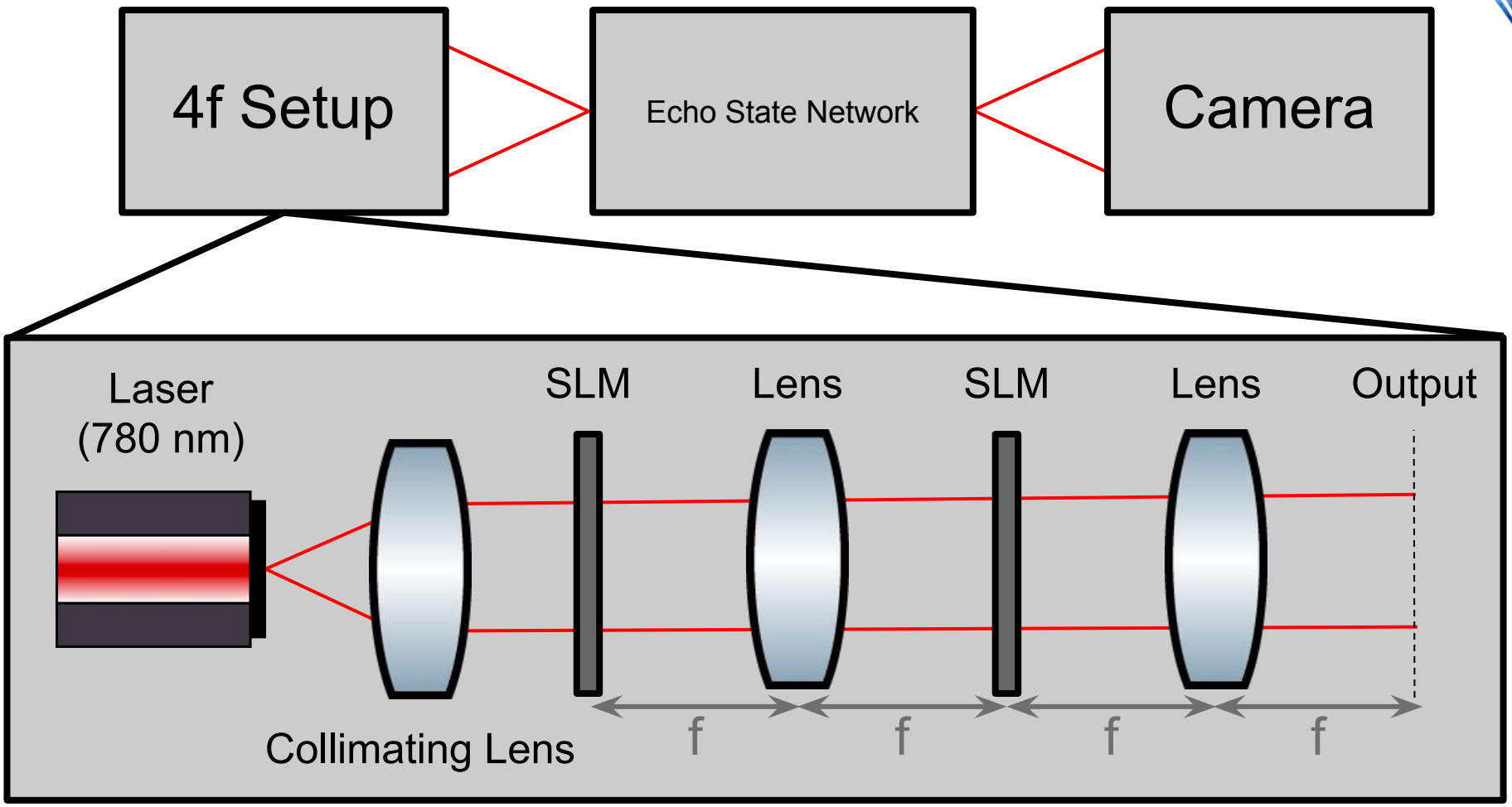
CNN → RC System

- Input signal encoded in SLM passes through optical convolution system
- Optical convolution system is three layers of convolution, outputting 84 randomized feature maps of the input data
- This last set of feature maps is 84 2x2 matrices corresponding to the input images; these are fed into an echo state network as a time series classification problem
- The echo state network classifies each batch as a function of time, and once trained, can be used to predict never-seen-before data



Full System

Supplemental Slides



KS Solution Algorithm (ETFRK

Let $L = 2\pi\tilde{L}$ be the system parameter relating to the grid length spatially. Let $u(x,t)$ represent a scalar field such that $u(x,t) = u(x+2\pi,t)$. We begin by expanding $u(x,t)$ in it's Fourier basis:

$$u(x,t) = \sum_{k=-\infty}^{k=\infty} a_k(t) e^{ikx/\tilde{L}}$$

We do this so that the entire calculation is in Fourier space, giving us scalar instead of matrix computation (for more information on this, see here.

We then implement the Fourth-Order Runge Kutta method. The complete set of update equations are given by:

$$a_n = u_n e^{ch/2} + \left(e^{ch/2} - 1\right) F\left(u_n, t_n\right)/c$$

$$b_n = u_n e^{ch/2} + \left(e^{ch/2} - 1\right) F\left(a_n, t_n + h/2\right)/c$$

$$c_n = a_n e^{ch/2} + \left(e^{ch/2} - 1\right) \left(2F\left(b_n, t_n + h/2\right) - F\left(u_n, t_n\right)\right)/c$$

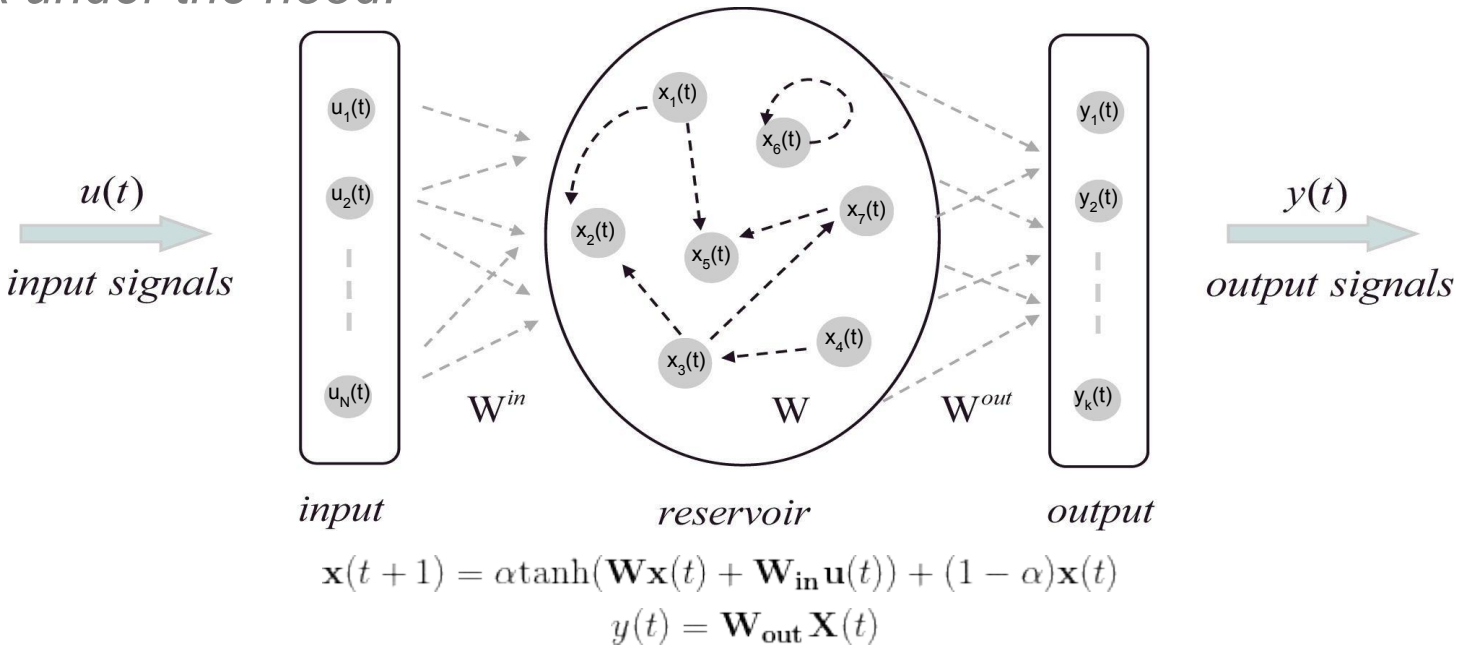
$$\begin{aligned} u_{n+1} =& u_n e^{ch} + \left\{F\left(u_n, t_n\right)\left[-4 - hc + e^{ch}\left(4 - 3hc + h^2c^2\right)\right]\right. \\ &+ 2\left(F\left(a_n, t_n + h/2\right) + F\left(b_n, t_n + h/2\right)\right)\left[2 + hc + e^{ch}(-2 + hc)\right] \\ &\left. + F\left(c_n, t_n + h\right)\left[-4 - 3hc - h^2c^2 + e^{ch}(4 - hc)\right]\right\}/h^2c^3 \end{aligned}$$

where F is our transformation to and from Fourier space. I also note that c is a constant (exponential time integration factor) and h is the system parameter $\tilde{L}$.

Summer 2019 Slides

An Echo State Network in Operation

A look under the hood.



$$\mathbf{x}(t+1) = \alpha \tanh(\mathbf{W}\mathbf{x}(t) + \mathbf{W}_{\text{in}}\mathbf{u}(t)) + (1-\alpha)\mathbf{x}(t)$$

$$y(t) = \mathbf{W}_{\text{out}}\mathbf{X}(t)$$

1. Generate a large, random reservoir ($\mathbf{W}^{\text{in}}$, $\mathbf{W}$).
2. Run every training input u(t) through the reservoir, collecting how the reservoir responds to each input x(t).
3. Compute $\mathbf{W}^{\text{out}}$ from these activation states x(t) using linear regression, minimizing the error between y(t) and $y^{\text{target}}(t)$.
4. Use the trained network on new input data $u^{\text{test}}(t)$, computing $y^{\text{predict}}(t)$ using the trained output weights $\mathbf{W}^{\text{out}}$.

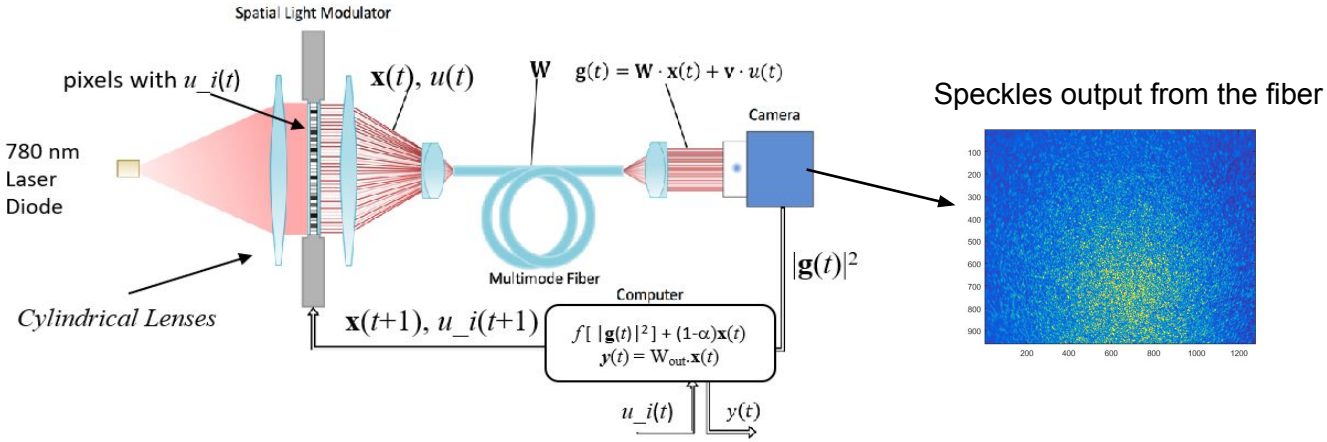
TAKEAWAY: Echo State Networks work by inputting a signal, seeing how the reservoir responds, and making predictions based on that response.

Figure: Ma et al., 2016

Optical Echo State Networks

Implementing neural networks in the lab

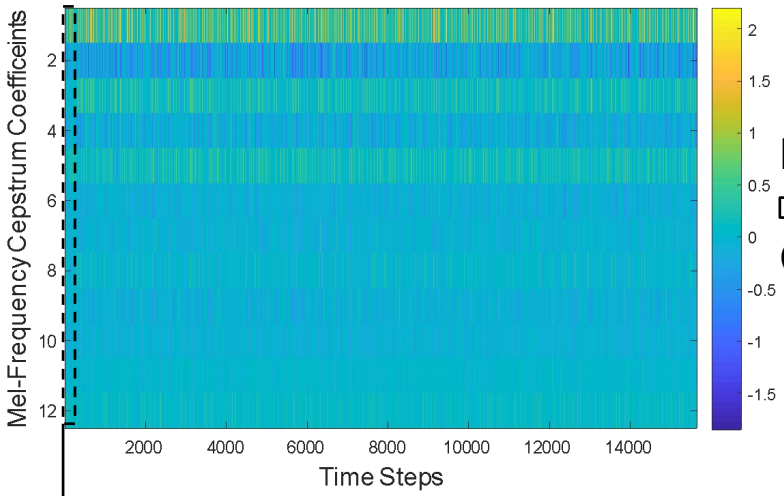
MOTIVATIONS + TAKEAWAYS: Using photonics, we can automatically perform the matrix multiplications "at the speed of light." This system implements an echo state network by using a speckle pattern as a substitute for electronic computation.



1. Encode the input as voltages into the pixels of the spatial light modulator (SLM).
2. Pass output of SLM into a multimode fiber to generate speckle pattern.
3. Apply non-linearity and feedback fraction of previous neurons to obtain new neuron values from speckle image.
4. Inject these neurons back into the SLM. Store this at each time step to build **X**(t).
5. Calculate $\mathbf{W}^{out}$ from **X**(t) using $\mathbf{Y}(t) = \mathbf{W}^{out}\mathbf{X}(t)$.
6. Run the system on new data to obtain an $\mathbf{X}^{test}(t)$. Recover input and classify by $\mathbf{Y}^{test}(t) = \mathbf{W}^{out}\mathbf{X}^{test}(t)$.

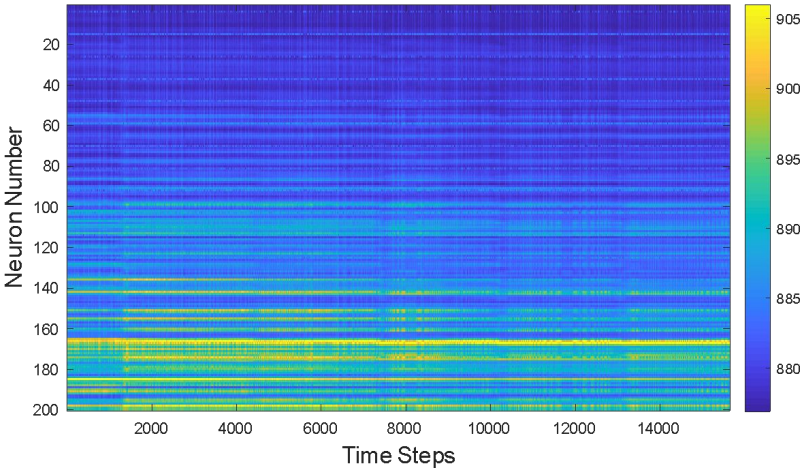
Experimental test of the optical reservoir computer

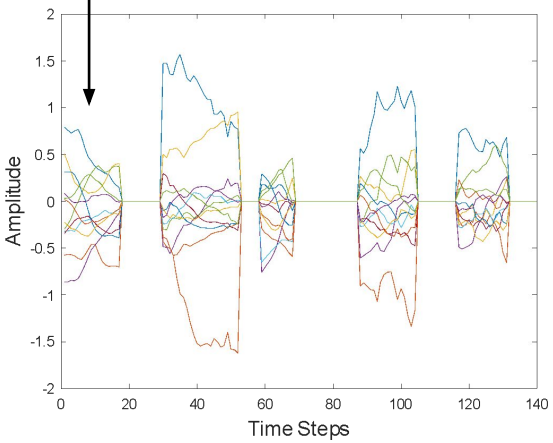
Training and Test Input



Reservoir Computer

Neurons Output





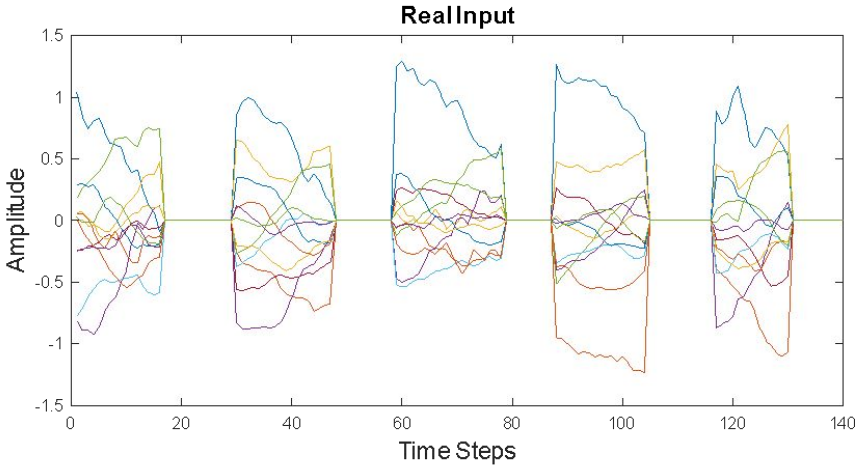
- Send input $u_i(t)$ to 12 pixels of SLM
- Take an image of the speckle pattern on the computer
  - Convert pixel values to neuron values
  - $\mathbf{x}(t+1) = \alpha f[\mathbf{W}\,\mathbf{x}(t) + \nu\, u_i(t)] + (1-\alpha)\,\mathbf{x}(t)$
- Inject feedback "neurons" $\mathbf{x}(t+1)$ to SLM
- Store vectors into $\mathbf{X}$ matrix of neurons for duration of training sequence
- Calculate output weight W_{out} matrix from training data and neuron matrix
  - $\mathbf{Y}(t) = W_{out}\mathbf{X}(t)$
- Continue running system to obtain test neuron matrix X
- Multiply test neuron X matrix by W_{out} to recover waveform and classify

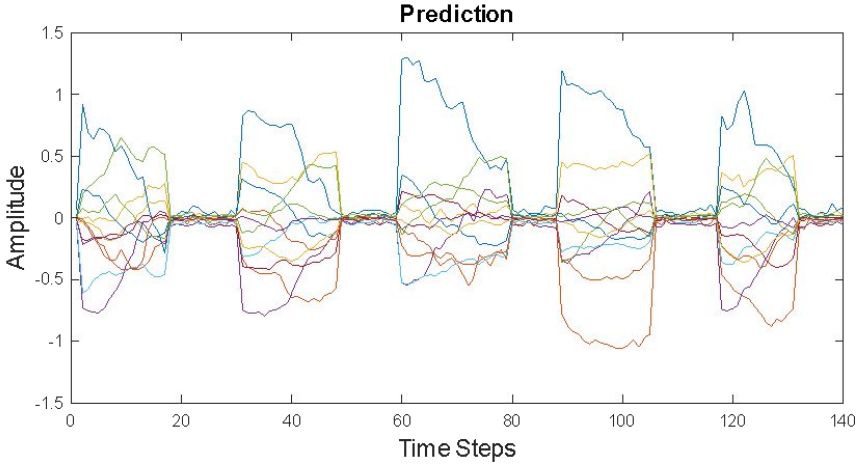
9 different speakers' MFCC coefficients ($Y_{predict} = W_{out}X_{test}$)

The features of the 12 inputs get mapped to 200 neurons. This enables important data features to be extracted automatically through the training process.

Input vs prediction of the Mel coefficients waveform

First 140 time steps of the prediction

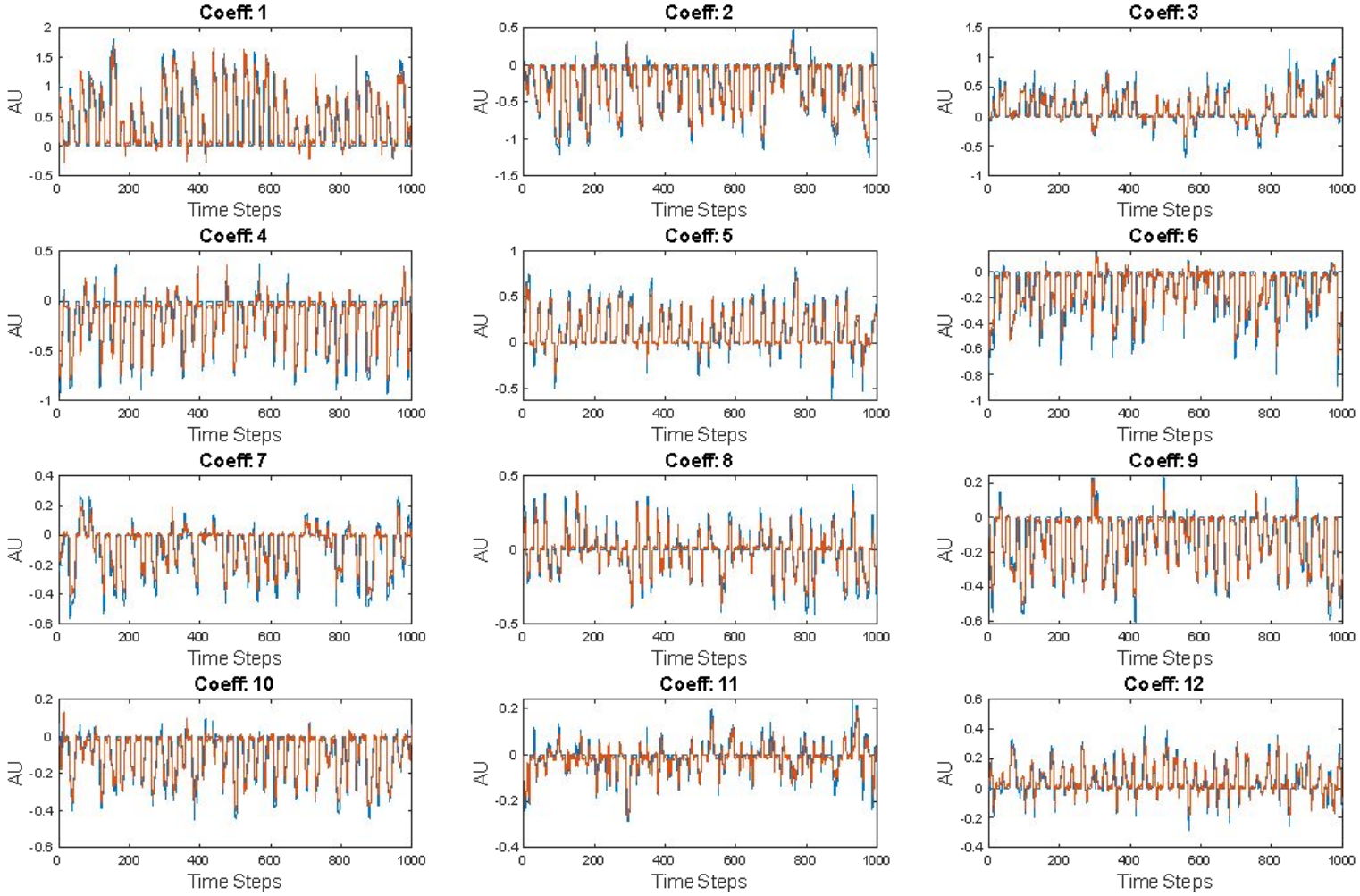




TAKEAWAY: *All features of the Mel coefficients are well captured by the RC.*

Input vs prediction of the Mel coefficients waveform

First 1000 time steps of the prediction



Blue: Input waveform

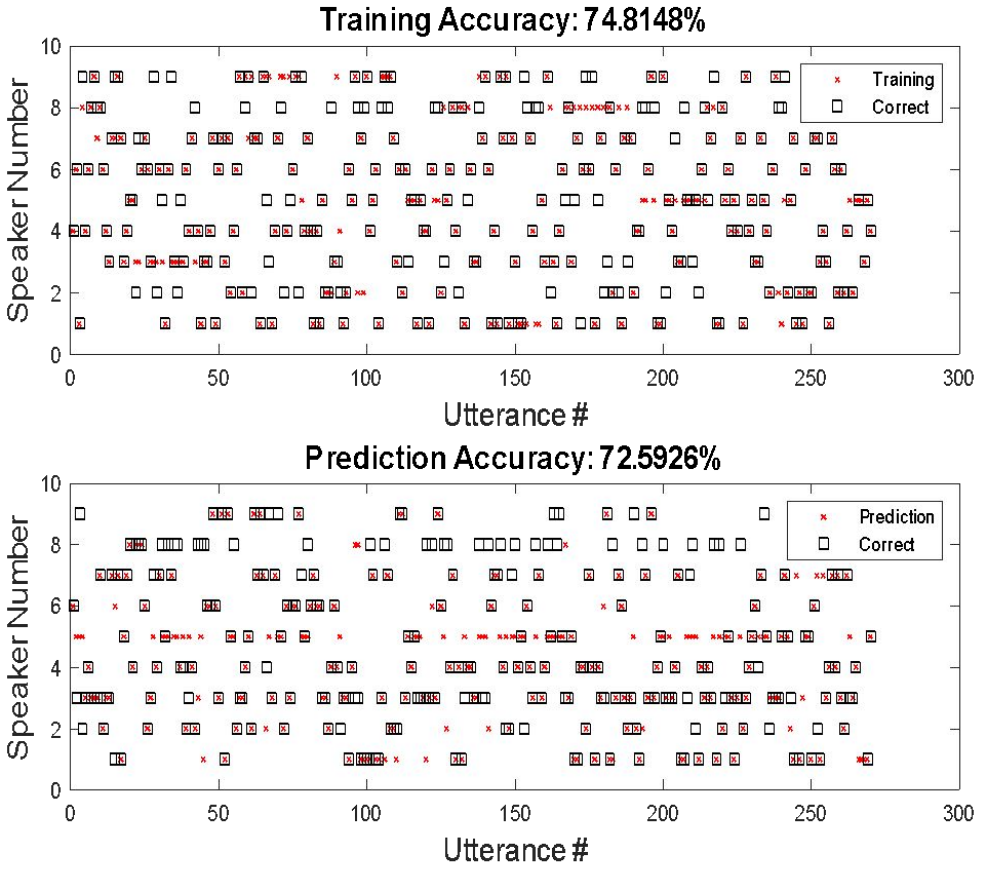
Orange: Reconstructed waveform

Waveform reconstruction is good but not perfect!

Classification accuracy

Can successfully predict the speaker with never seen training data

- First 270 audio inputs are used as training.
- Remaining 270 are used for test/prediction.
- Overlap of red cross and black square correspond to correct predictions.
- Obtain ~73% accuracy with regularized regression
- Random Guess Accuracy: ~11%



TAKEAWAY: *Successfully classified audio files using optical RC.*

Toward the future…

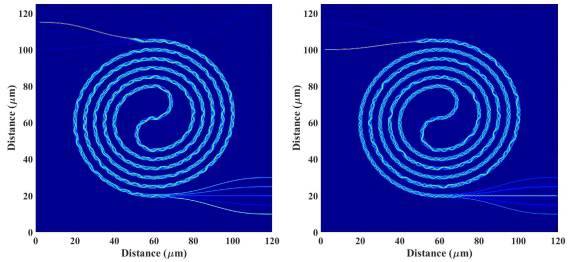
What are the next steps?

- Short Term
  - Improve the speed of data acquisition + processing
  - Stability of the optics (thermal stability, vibrational stability, etc.)
  - Optimize the kinds of problems and parameter space in which we operate echo state network
  - Understand the effect of noise and drift on the experimental setup
  - Increase number of nodes + compare simulation optimized parameters with experimental optimized parameters

- Long Term
  - Robust signal classification
  - Raw data processing (instead of intermediate MFCC-like steps)
  - Photonic integrated circuit for echo state networks

Example of planar waveguide on SOI platform

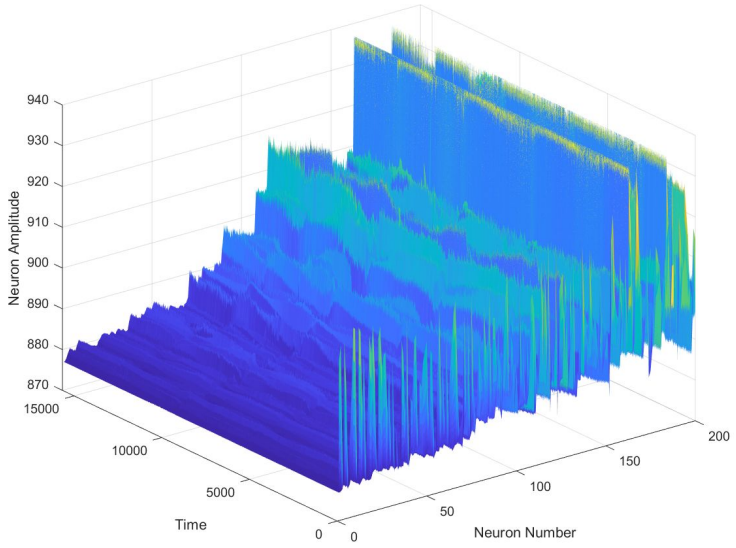
Simulated propagation for different input ports



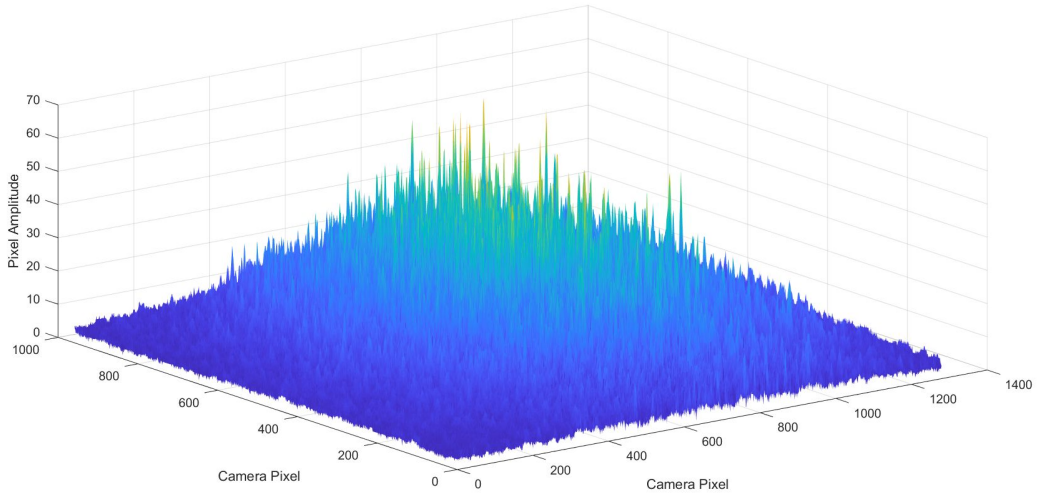
CONCLUSION: *Much progress has been made, but as always, there is more to do.*

Visualizations

Neuron Value vs. Time

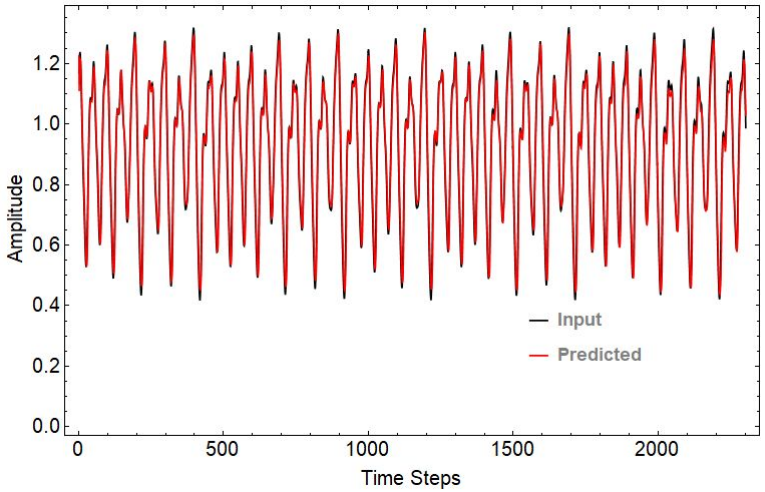


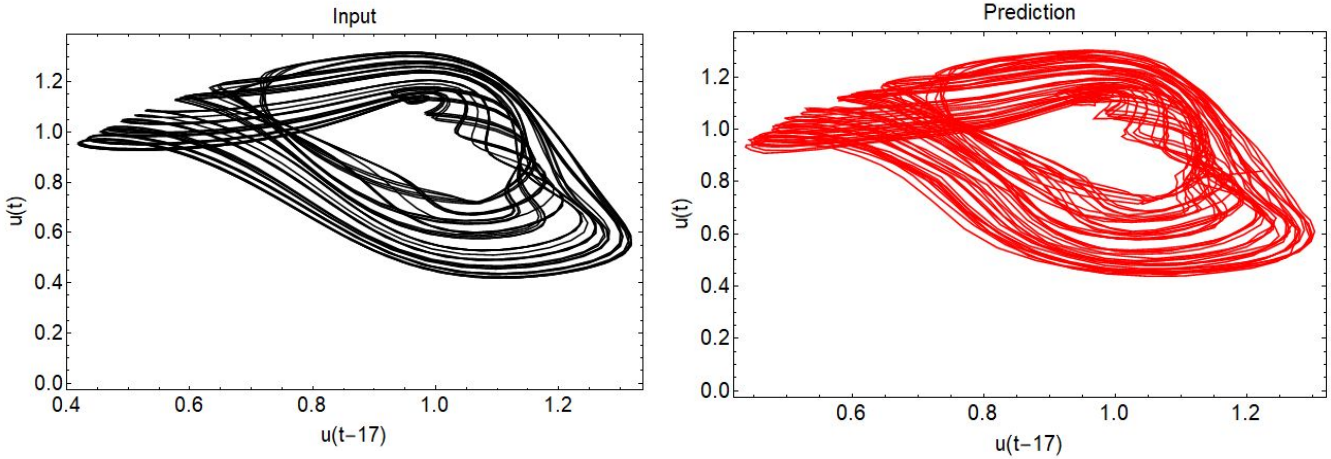
Background



Mackey-Glass non-linear time-series prediction

- Mackey-Glass function gives a complex, near-chaotic dynamics and is extremely difficult to predict.
- A Mackey-Glass time-series is mapped using the optical hardware. The system can map-out the series with high accuracy.



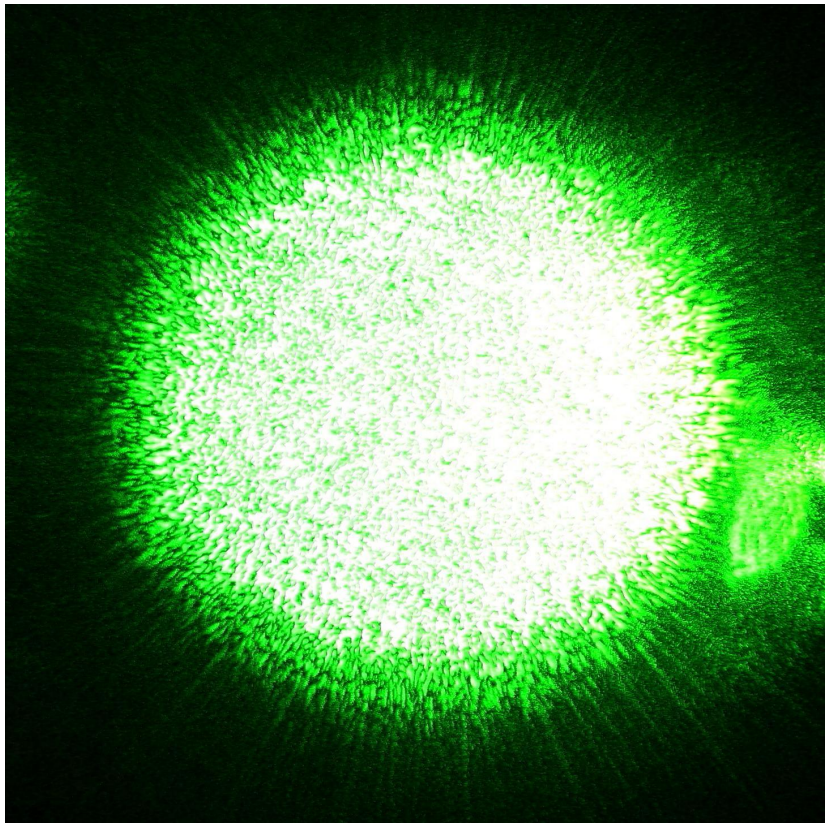


Chaotic attractor for the input and prediction generated from the time delayed MG input. The plots show that the system operates in a high-dimensional regime and our optical reservoir computer is fully able to map the dynamics.

Waveguide speckle-based optical reservoir computer can map near-chaotic dynamics

What is speckle?

- Pattern of light on a surface produced by mutual interference of wave fronts
- Typically occur in diffuse reflection of monochromatic light (like laser light)
- Results because different phases and amplitudes add to give a resultant, random pattern of light



Mathematical Formulation of Echo State Networks

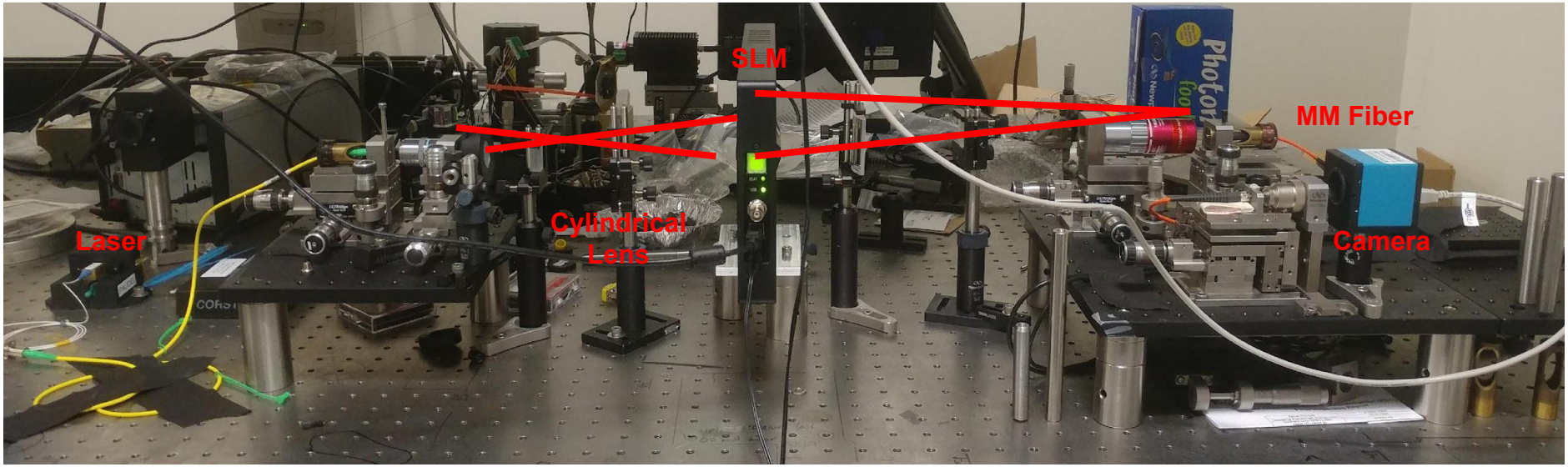
$$\mathbf{x}(t+1) = \alpha f[\mathbf{W}\,\mathbf{x}(t) + \boldsymbol{v}\,u_i(t)] + (1-\alpha)\,\mathbf{x}(t)$$

$$y(t) = \mathbf{W}^{\text{out}}\mathbf{x}(t)$$

- $\mathbf{x}(t)$ are the values of the neurons at time t
- α is the "leaking rate"
- f is the nonlinear sigmoidal function (e.g. tanh(x), sigmoid)
- $\mathbf{W}$ and $\boldsymbol{v}$ are the random weight matrices from speckle
- $u_i(t)$ are the input signals
- $y(t)$ is the output signal
- $\mathbf{W}_{\text{out}}$ is the vector of output weights
- $\mathrm{X}_{\text{training}} = \{\mathbf{x}(0), \mathbf{x}(1),\ldots, \mathbf{x}(n_{\text{training}})\}$
- $\mathbf{W}_{\text{out}} = (\mathrm{X}_{\text{training}})^{\text{PseudoInverse}}\,\mathrm{Y}_{\text{training}}$
- $\mathrm{X}_{\text{test}} = \{\mathbf{x}(n_{\text{training}}+1),\ldots,\mathrm{x}(n_{\text{training}}+ n_{\text{test}})\}$
- $\mathrm{Y}_{\text{test}} = \mathbf{W}_{\text{out}}\,\mathrm{X}_{\text{test}}$

n_{training} = length of training data
n_{test} = length of test data

Experimental Layout



Japanese Language Classification

Slide from Marta Luengo-Kovac + Uttam Paudel

***GOAL:** Classify spoken Japanese vowels into their corrects bins using an optical reservoir computer.*

- Nine male speakers uttered two Japanese vowels /ae/ successively
- The audio files are pre-processed to generate Mel-frequency cepstrum coefficients (MFCC)
- Each sequence consists of a time series of 12 MFCC coefficients
- Training set: 270 sequences (30 utterances by 9 speakers)
- Testing set: 270 sequences (24-88 utterances by the same 9 speakers)
- Length of time series: 7-29 depending on the utterance
- The task is to classify speakers using the optical RC

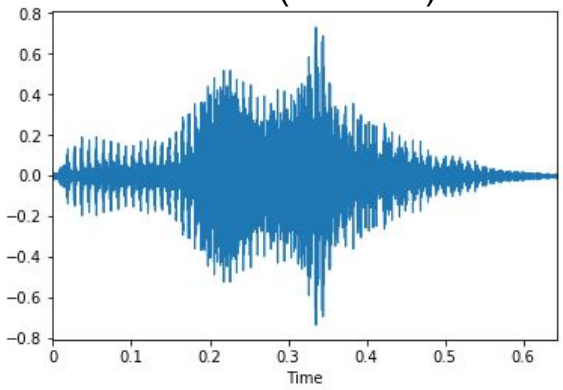
Mel-frequency cepstrum coefficients

Widely-used method for speech classification

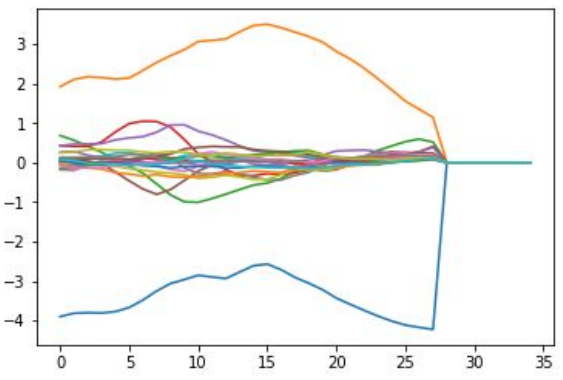
Slides from Marta Luengo-Kovac + Uttam Paudel

- The mel-frequency cepstrum is essentially the spectrum of the power spectrum of a sound
  - *Compute the power spectrum of a signal*
  - *Bin the power spectrum based on the mel scale (this scale more closely matches the human auditory system – e.g. less sensitive at higher frequencies)*
  - *Take the log of the power at each mel frequency (because human hearing follows a log scale)*
  - *Take the discrete cosine transform of the mel log powers to get the spectrum of the log of the power spectrum*
  - *The MFCCs are the amplitudes of the resulting spectrum*

"One" (.wav file)

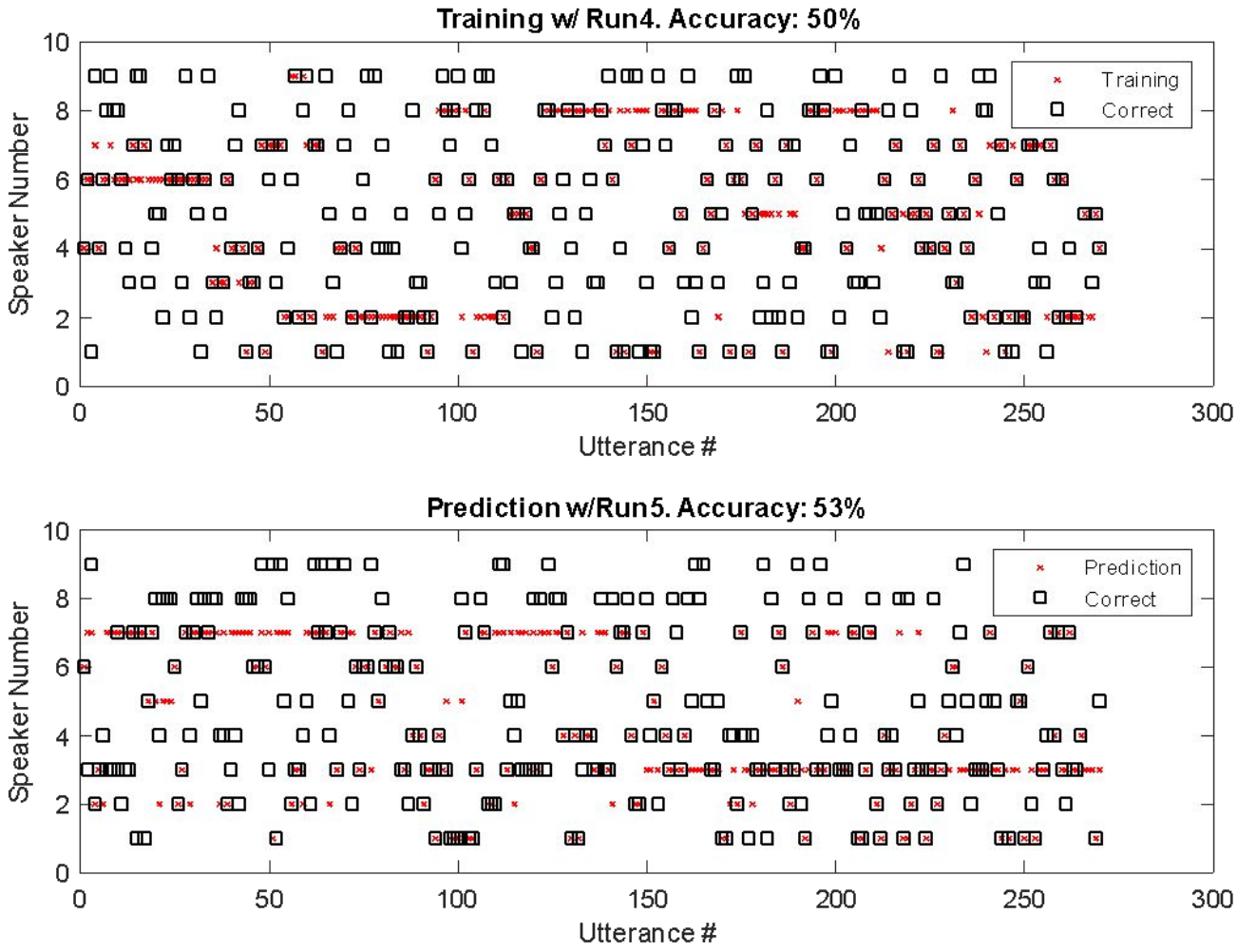


MFCCs



  - *Generally only coefficients 2-13 are used (the first coefficient is usually a large offset and the higher coefficients are mostly high frequency noise)*

System stability (Experiment)



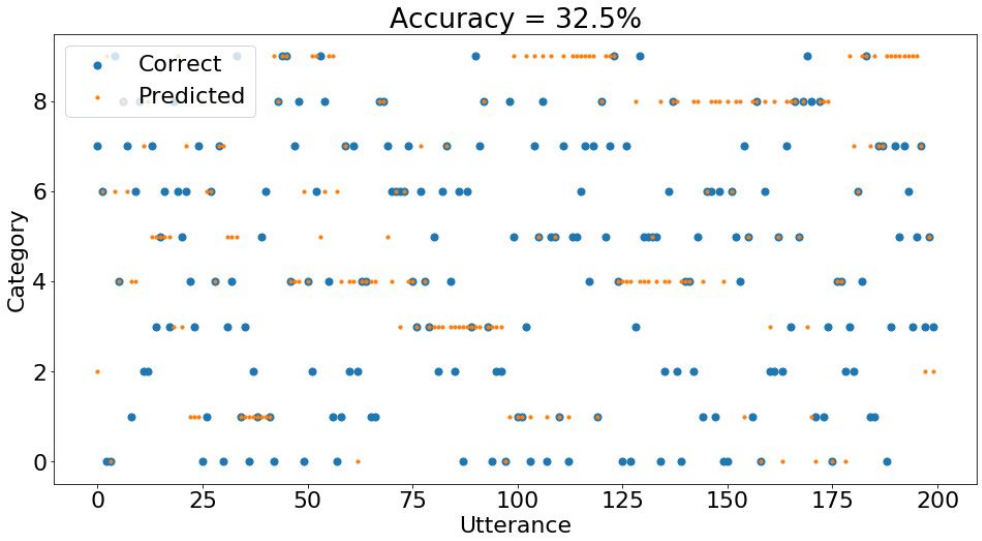
Training and test were taken over 6 hours time window.

Integrated photonics should allow >GHZ processing, training would take microsecond time.

English spoken digits classification

Four speakers with different accents

- The 10 English digits were spoken 10 times each by 4 different speakers (2 with American accents, one with a French accent, one with a German accent).
- Of those 10 times, half data were used for training and half were used for testing, resulting in 200 training utterances and 200 testing utterances.
- The raw data was the waveform of each utterance. This was converted into its mel-frequency cepstrum coefficients before being fed into the RC.
- The prediction dataset are not seen during the training period.



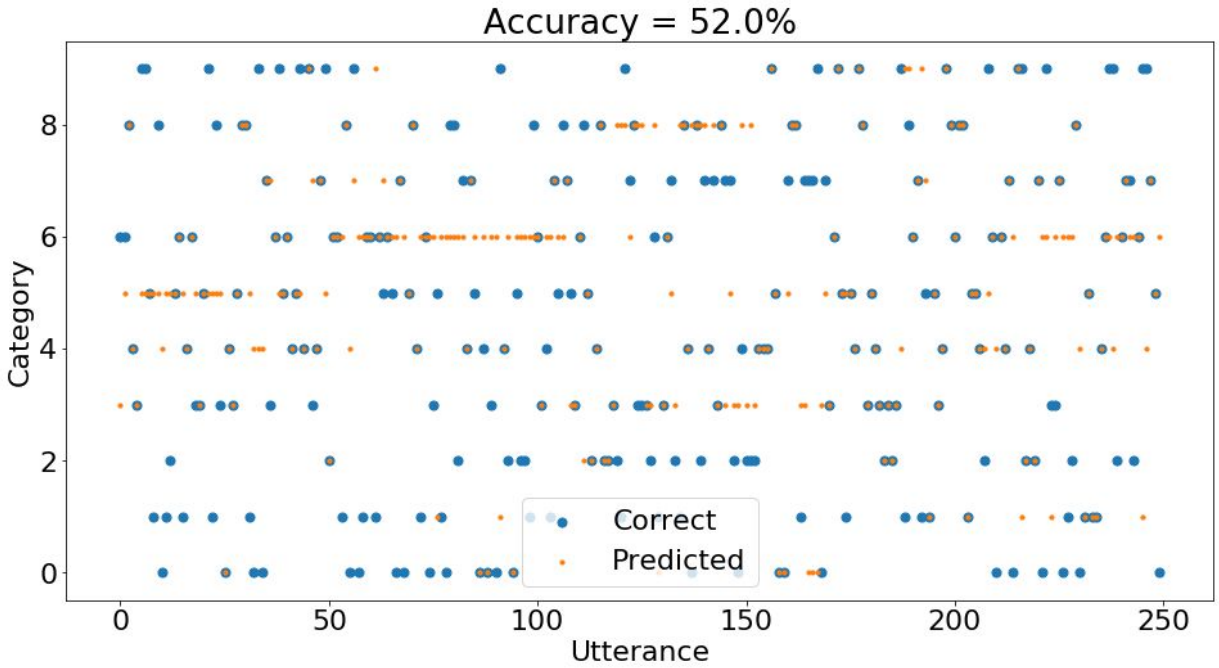
- Obtain 32.5% accuracy which is above random chance (10%).
- The lower accuracy might be due to multiple speakers with different accents.
- Longer training dataset should increase the accuracy.

Above random guess classification for English spoken digits problem.

English spoken digits classification

Single speaker

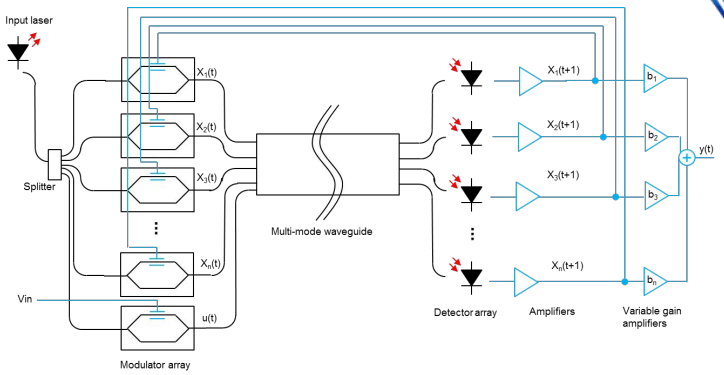
- The 10 English digits were spoken by a single speaker.
- Half data were used for training and half were used for testing, resulting in 200 training utterances and 200 testing utterances.
- The raw data was the waveform of each utterance. This was converted into its mel-frequency cepstrum coefficients before being fed into the RC.
- The prediction dataset are not seen during the training period.



52% classification accuracy for English spoken digits by a single speaker.

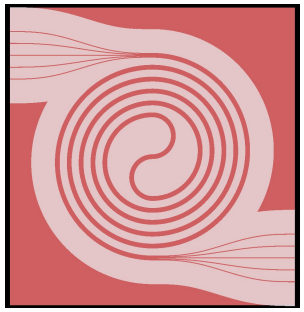
Chip-scale integrated photonics reservoir computer

- Lasers can be fabricated (InP) or chip-mounted (Si)
- Modulators array could be build using electro-absorption devices (InP) or compact ring modulators (Si).
- Planar multimode waveguide in silicon is verified to generate the speckles as predicted.
- Germanium detectors can be fabricated on silicon platform.
- All components can be readily fabricated on an integrated platform using a commercial foundry.

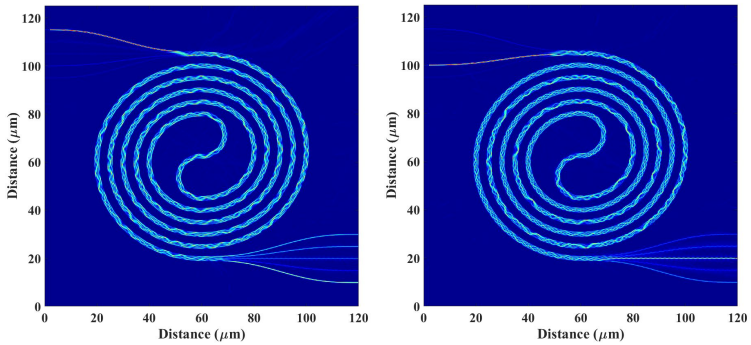


A schematic for an optical reservoir computer implemented on photonic integrated circuit.

Top-view



Example of planar waveguide on SOI platform



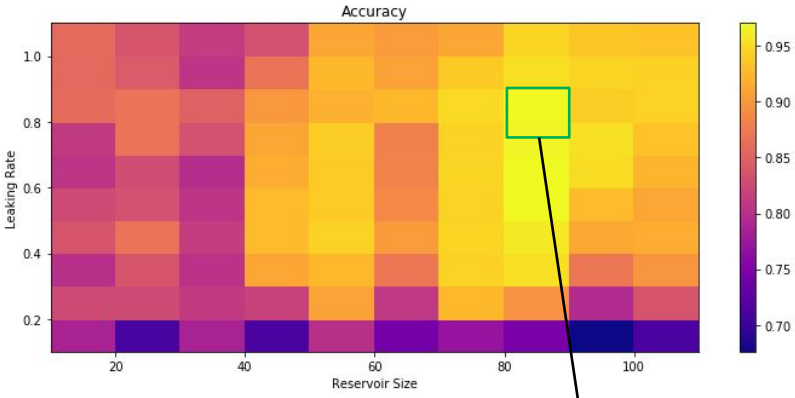
Simulated propagation for different input ports

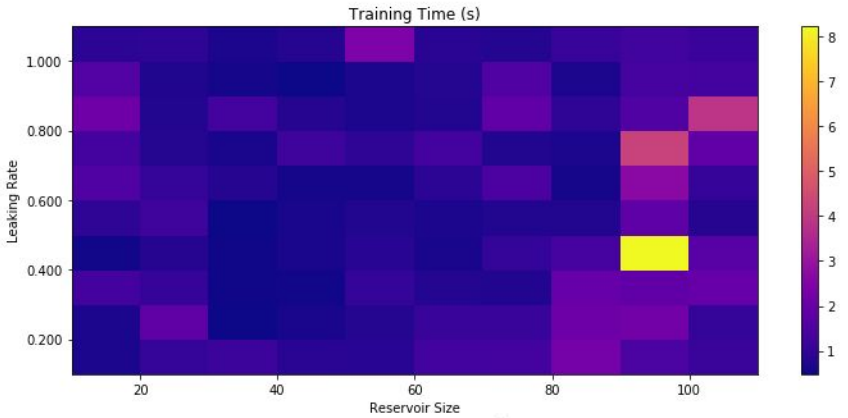
Clear path-forward for building integrated photonics circuit. Can achieve>GHZ processing with a PIC, training would take ~microsecond time.

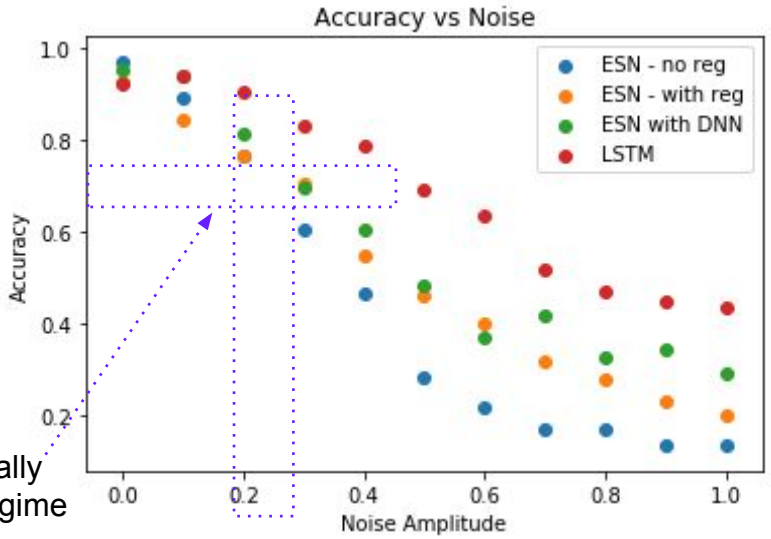
Reservoir computer on a personal laptop

Alpha, neuron size, and noise analysis

- The training sequences were sent in order (i.e. all of Speaker 1's sequences, then all of Speaker 2's sequences, etc.)
- The test sequences were sent in a random order







- Incorrect predictions indicated by blue dots.
- **Reservoir size = 80**, Leak rate = 0.8
- **Accuracy = 97.3%, Training time = 0.6 s**

Experimentally achieved regime

80 high-quality neurons are sufficient for classification, doable number for high-speed PIC

- really quick overview:
  - NN, why they're important → ESN is lower SWAP and we can do the calculations optically. reference last summer's work
- Introduce feed forward neural networks
  - 3b1b animation, trained w back propagation and glorified linear algebra problem
  - Training takes a long time, but great performance. ESN is like this, but the nodes are recurrent and you only train the output weights
  - ESN works by dimensionality expansions to tease at details
  - **Takeaway:** We know ESNs work, but what sorts of problems are they good at, are they truly more efficient than electronic computing, and what makes a prediction good?
- Applications
  - CNN + RC Architecture for Image Recognition + Video Recognition (see hannah's talk!)
  - Kuramoto Sivashinsky Equation solution -- didn't do much with ESN but wrote the code to solve the equation
  - Phased Array
- Performance
  - Performance of phased array made me question singular value decomposition + alternate modes of shuffling
- Future Work:
  - Understand exactly how and when ESN performs well