The Legendre Memory Unit

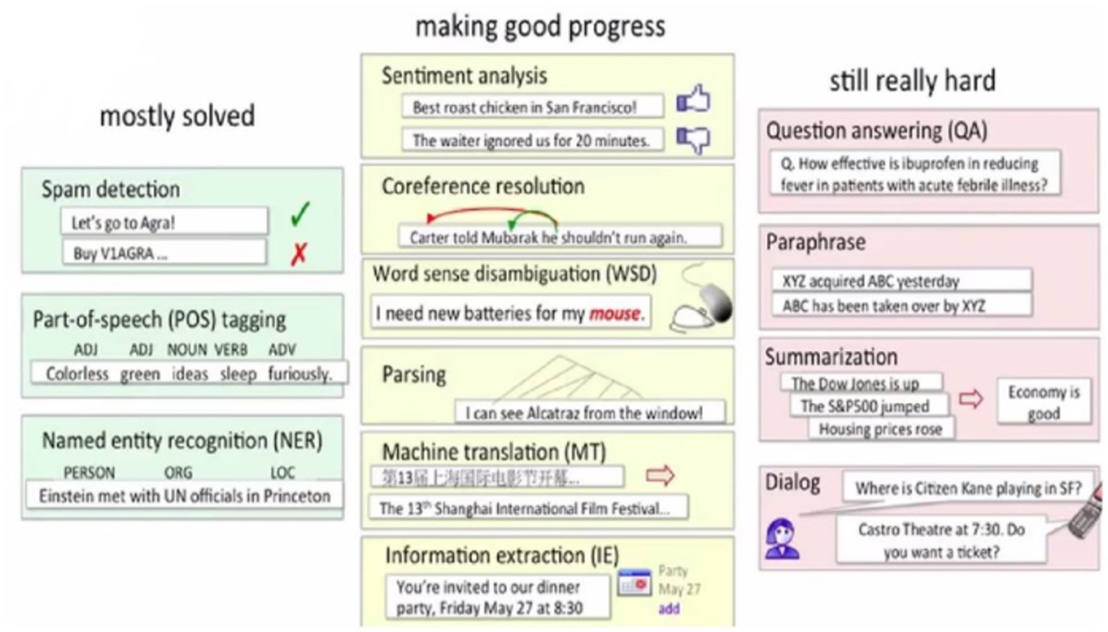
A neural network with optimal time series compression

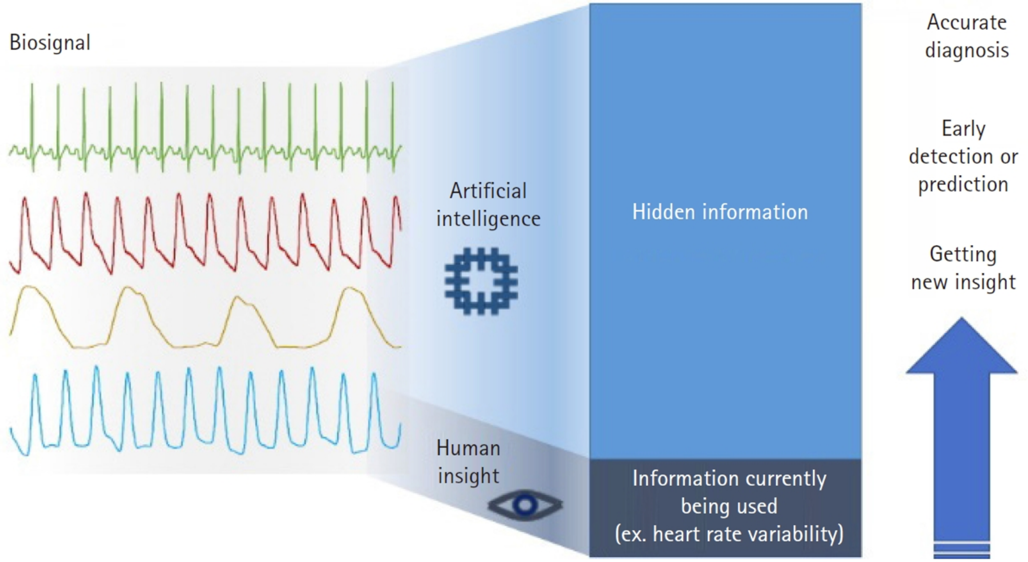
Chris Eliasmith
Centre for Theoretical Neuroscience
University of Waterloo
Applied Brain Research, Inc.

Time Series AI

Time Series Problems

- Speech recognition
- Language processing
- Network monitoring
- Stock market prediction
- Network monitoring
- Video processing (gesture recognition, tracking, etc.)
- Biosignals monitoring
- Signal processing

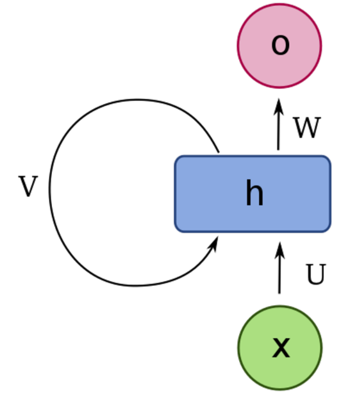




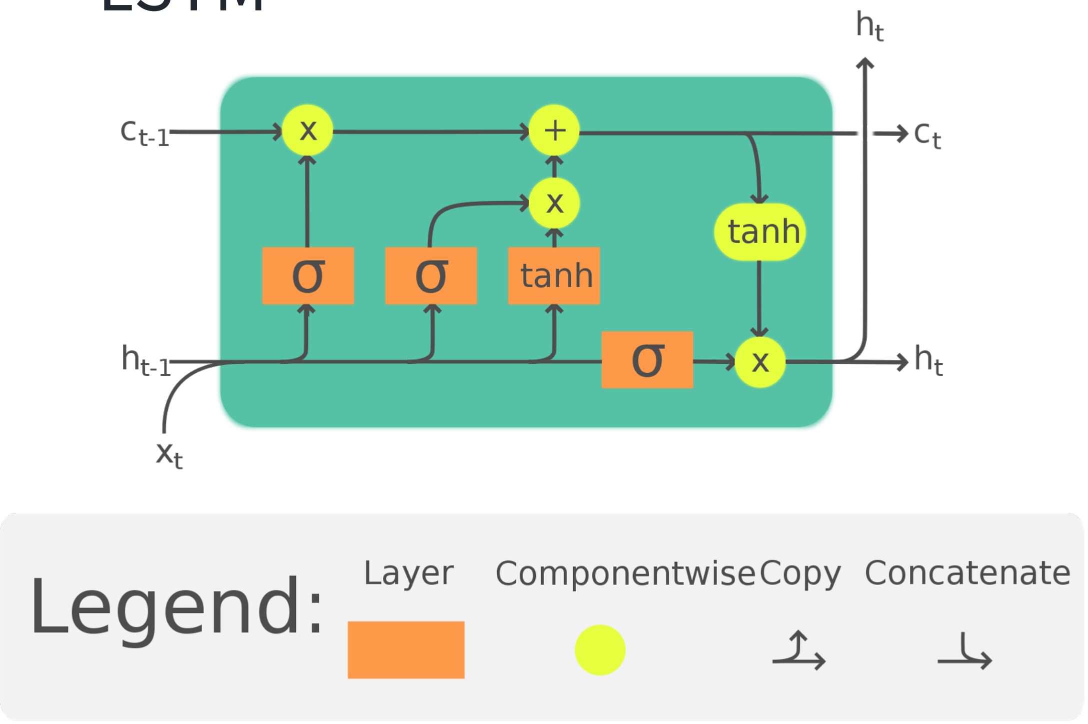
Long Short-Term Memory

- Proposed in 1995 as an improvement over vanilla **RNNs**
- "**LSTM** has become the **most cited** neural network of the 20th century"
- Generally very successful on time series data

RNN

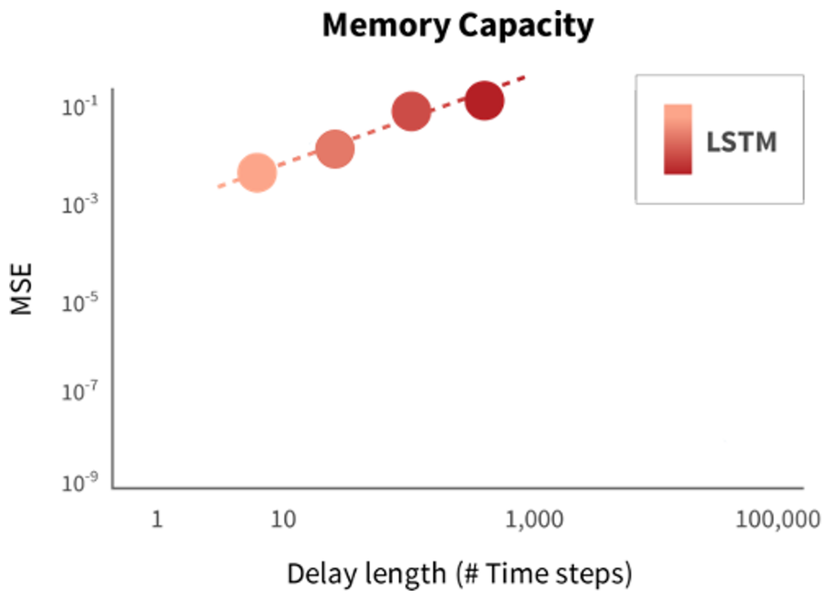


LSTM



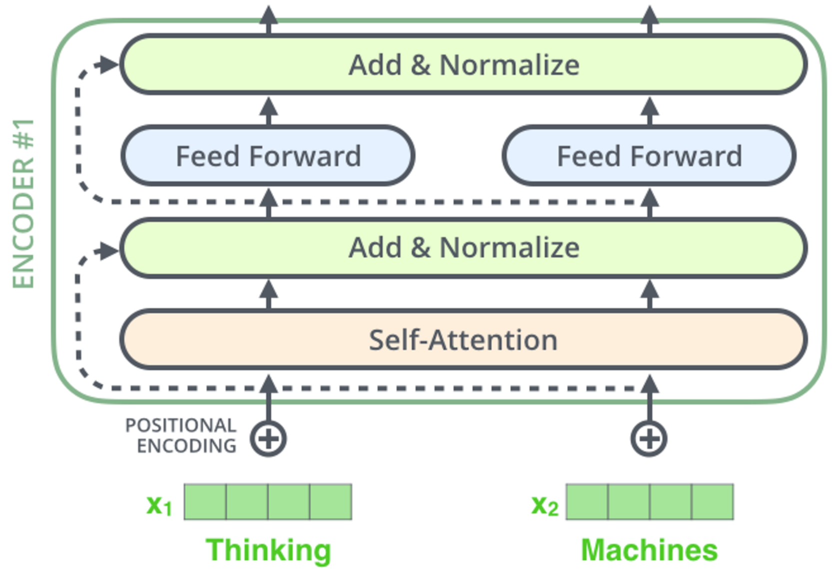
Limitations of the LSTM

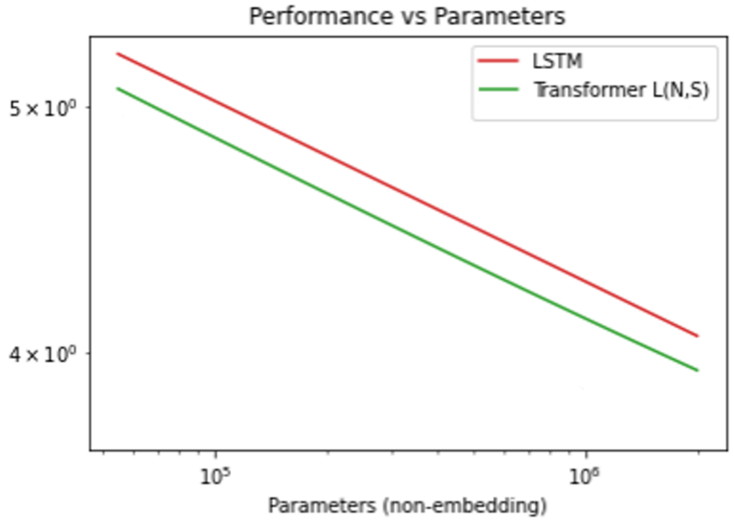
- Very difficult to train at large scales (e.g., for NLP)
  - Cannot effectively parallelize training
- 'Black box' processing gives little insight into behavior
- Sequences greater than 500-1000 samples cannot be effectively processed in practice



Transformer

- Borrowed ‘attention’ from LSTM work, makes time parallel
- Purely **feedforward** training, leverages huge GPU farms
- By far the **dominant** architecture for NLP (Google, Amazon, Apple, OpenAI, etc.)





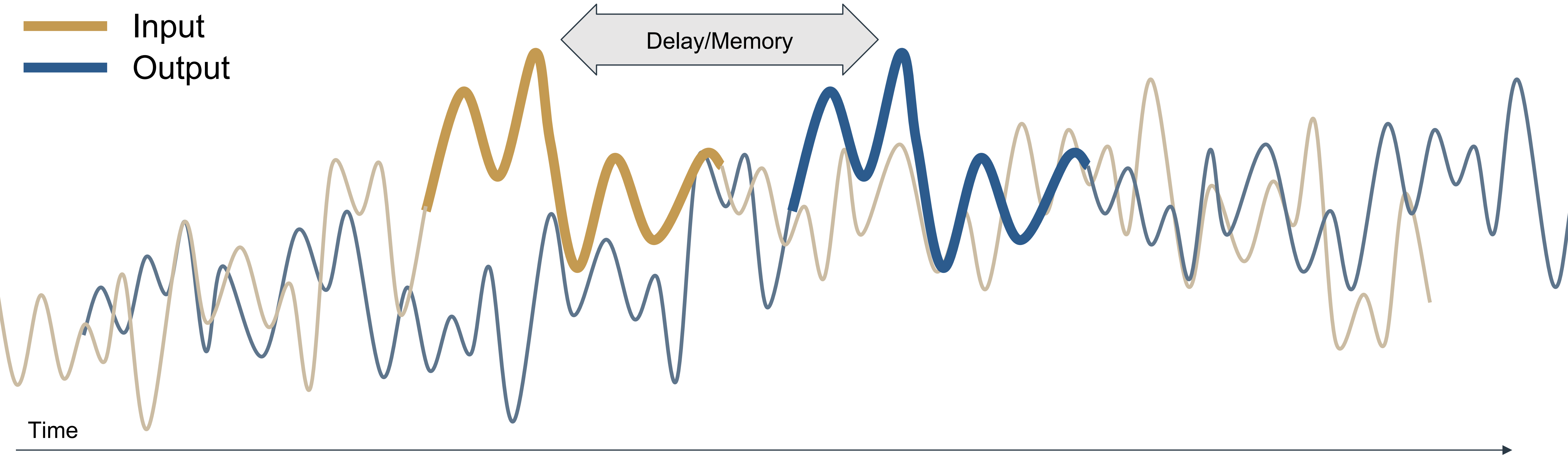
OpenAI (2021)

A **New** Neural Network

The problem

Perfect continuous delay is an **infinite dimensional** problem

- Infinite frequency content over any finite interval requires infinite information preservation



The problem

Ideal continuous delay: $y(t) = u(t - \theta)$

Or, in the Laplace domain: $F(s) = \frac{y(s)}{u(s)} = e^{-\theta s}$

Optimal finite version?

Equivalently: $F(s) = C(sI - A)^{-1}B + D$

The solution

The best approximation of any function with a rational function of order p/q is given by the Padé approximants

$$[p/q]\, e^{-\theta s} = \frac{\mathcal{B}_p(-\theta s)}{\mathcal{B}_q(\theta s)},$$

$$\mathcal{B}_m(s) := \sum_{i=0}^{m} \binom{m}{i} \frac{(p+q-i)!}{(p+q)!} s^i.$$

Choosing p=q-1, the state space becomes:

$$A = \begin{pmatrix} -v_0 & -v_0 & \cdots & -v_0 \\ v_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & v_{q-1} & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} v_0 & 0 & \cdots & 0 \end{pmatrix}^T$$

$$C = \begin{pmatrix} w_0 & w_1 & \cdots & w_{q-1} \end{pmatrix}$$

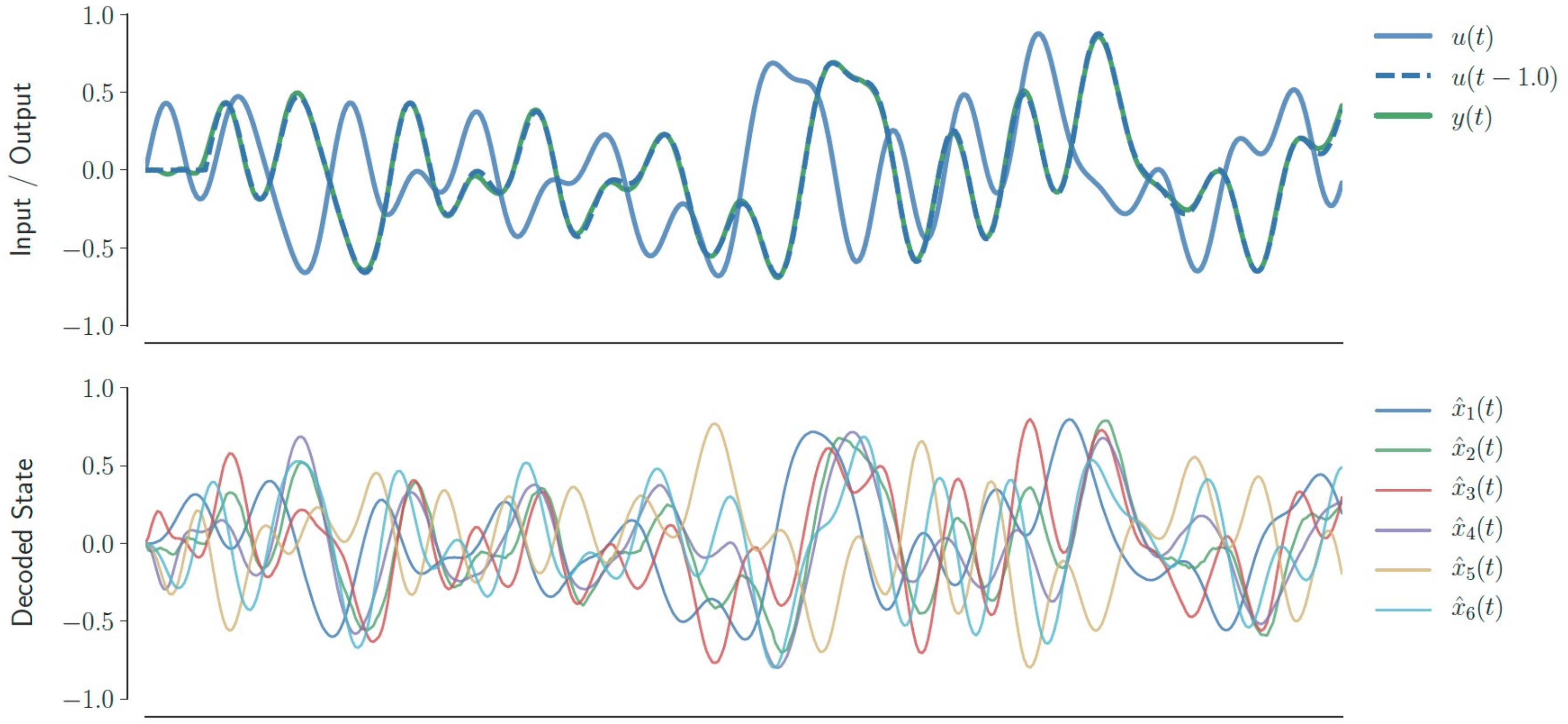
$$D = 0,$$

$$v_i := \frac{(q+i)(q-i)}{i+1} \theta^{-1} \qquad w_i := (-1)^{q-1-i} \left(\frac{i+1}{q}\right)$$

The solution

Our ‘Legendre Delay Network’ (LDN), is the **optimal finite solution** to this infinite dimensional system

$$\theta \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bu(t)$$



The solution

We can then convert that LTI to a normalized form:

$$A_{i,j} = \frac{(2i+1)}{\theta} \begin{cases} -1 & i < j \\ (-1)^{i-j+1} & i \geq j \end{cases}$$

$$B_i = \frac{(2i+1)(-1)^i}{\theta},$$

$$C_i = (-1)^i \sum_{l=0}^{i} \binom{i}{l} \binom{i+l}{j} (-1)^l,$$

$$D = 0, \quad i, j \in [0, d-1],$$

Projecting this state space onto the shifted Legendre polynomials

$$\tilde{\mathcal{P}}_i(r) = (-1)^i \sum_{j=0}^{i} \binom{i}{j} \binom{i+j}{j} (-r)^j = \mathcal{P}_i(2r-1), \quad r = \frac{\theta'}{\theta},$$

Gives

$$\boxed{u(t-\theta') \approx \sum_{i=0}^{q-1} \tilde{\mathcal{P}}_i\left(\frac{\theta'}{\theta}\right) x_i(t), \quad 0 \leq \theta' \leq \theta.}$$

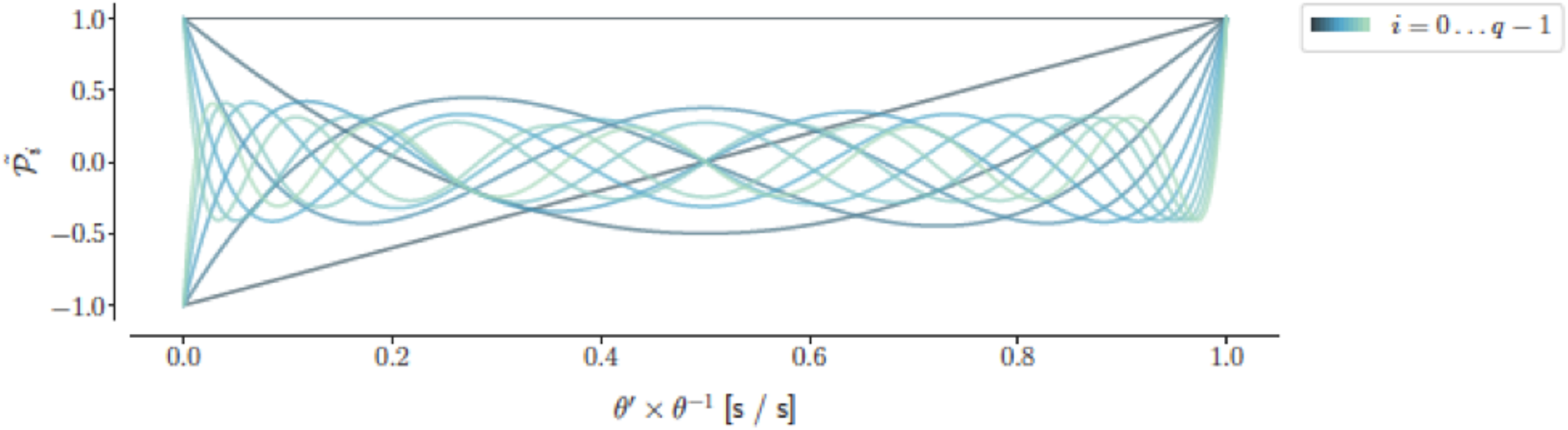
The solution: Examples

For q=6 this results in the following matrices

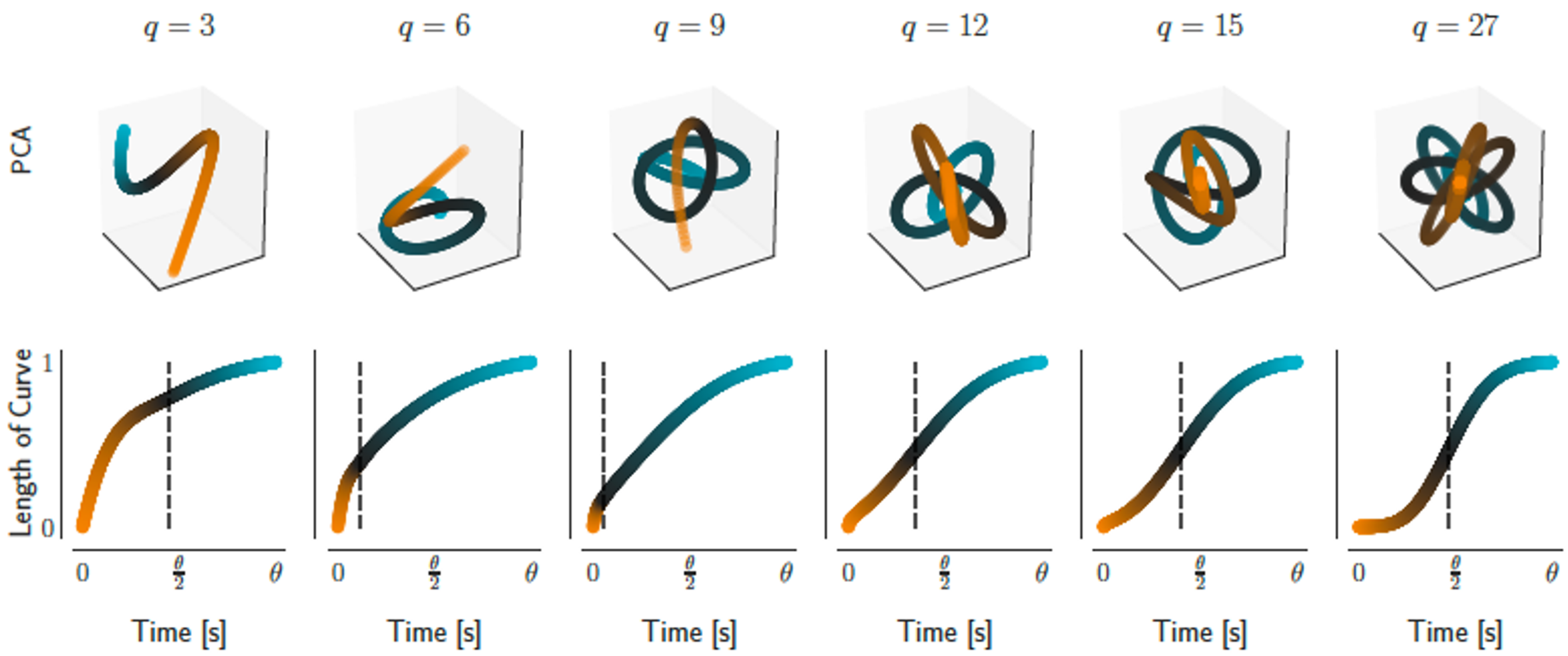
$$A = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ 3 & -3 & -3 & -3 & -3 & -3 \\ -5 & 5 & -5 & -5 & -5 & -5 \\ 7 & -7 & 7 & -7 & -7 & -7 \\ -9 & 9 & -9 & 9 & -9 & -9 \\ 11 & -11 & 11 & -11 & 11 & -11 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ -3 \\ 5 \\ -7 \\ 9 \\ -11 \end{pmatrix}$$

For q=1 this system is a first-order low pass

Shifted Legendre Basis

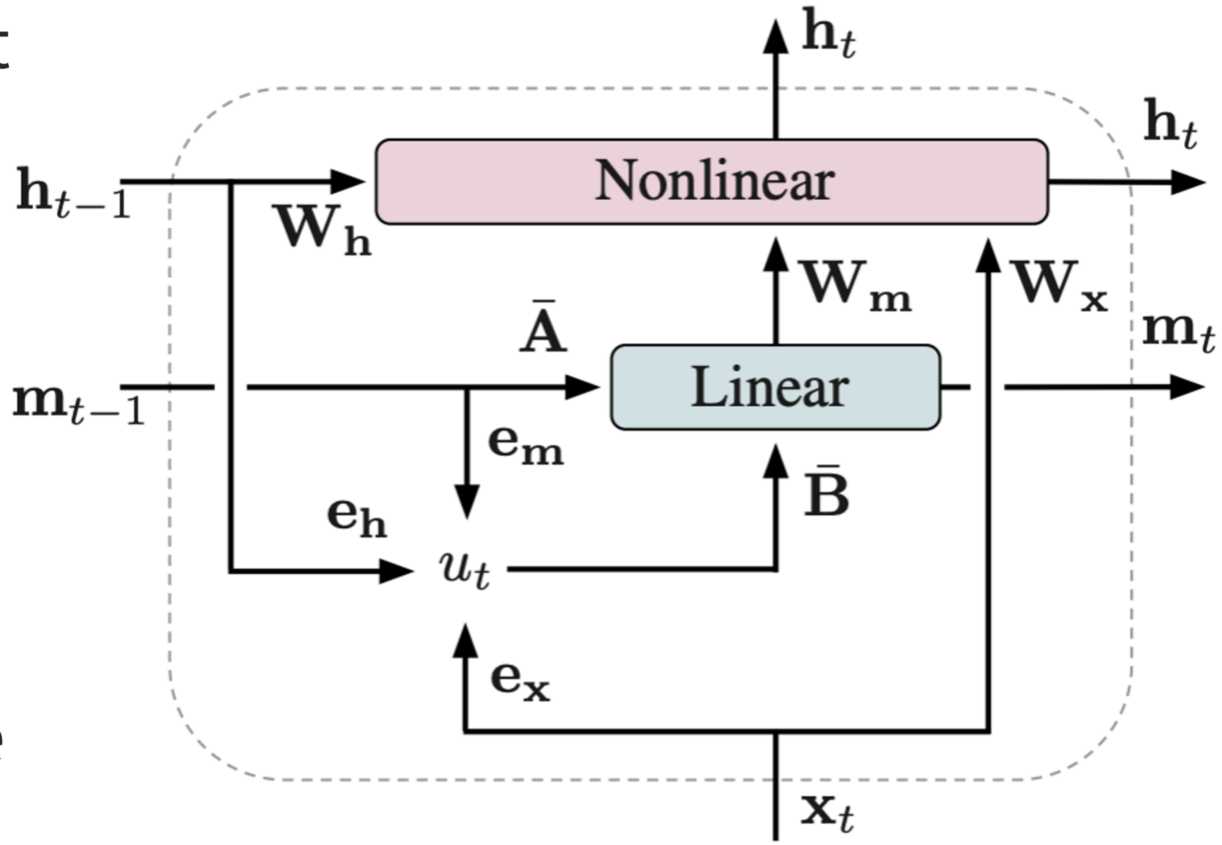


The solution: Impulse response

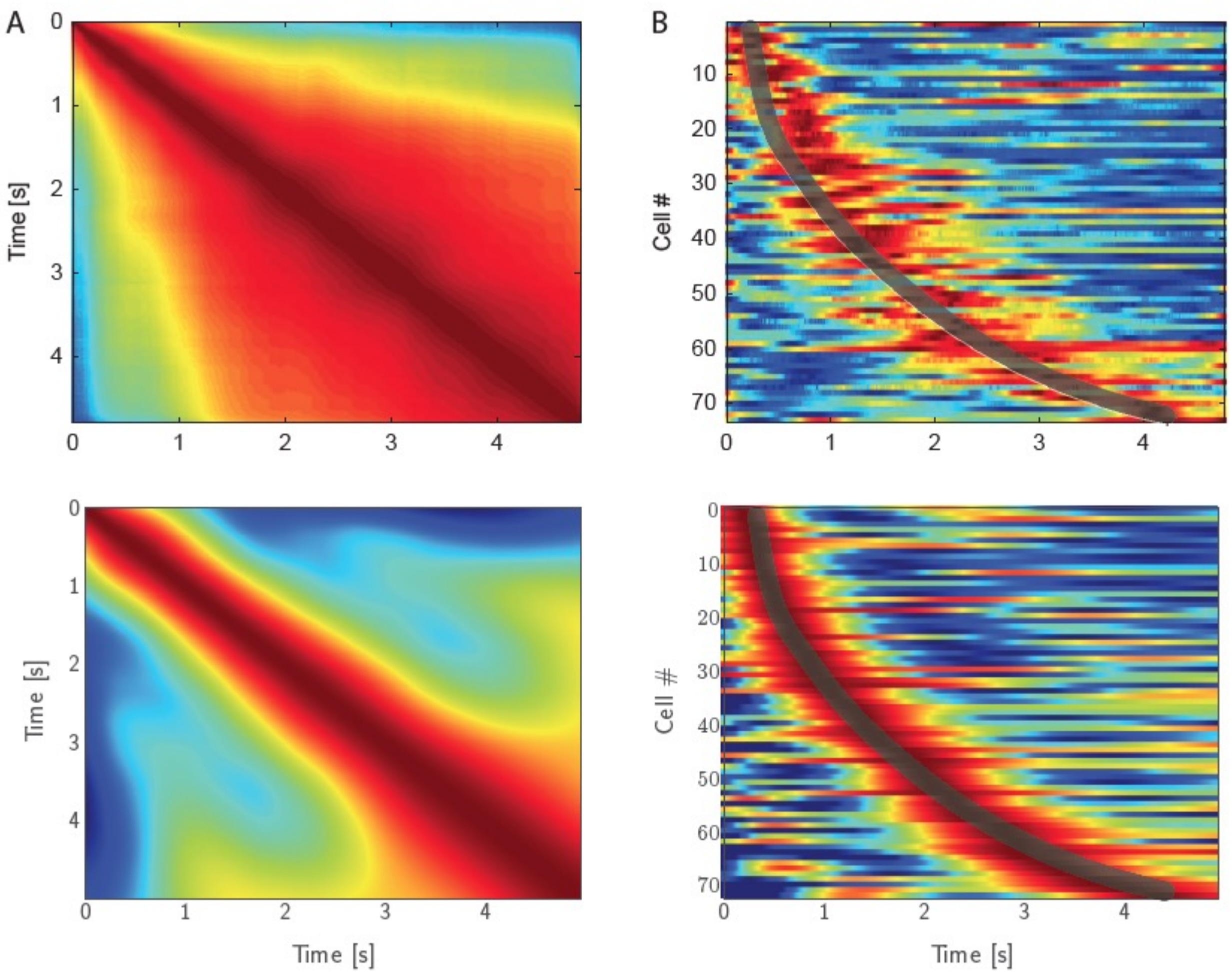


The Legendre Memory Unit (**LMU**)

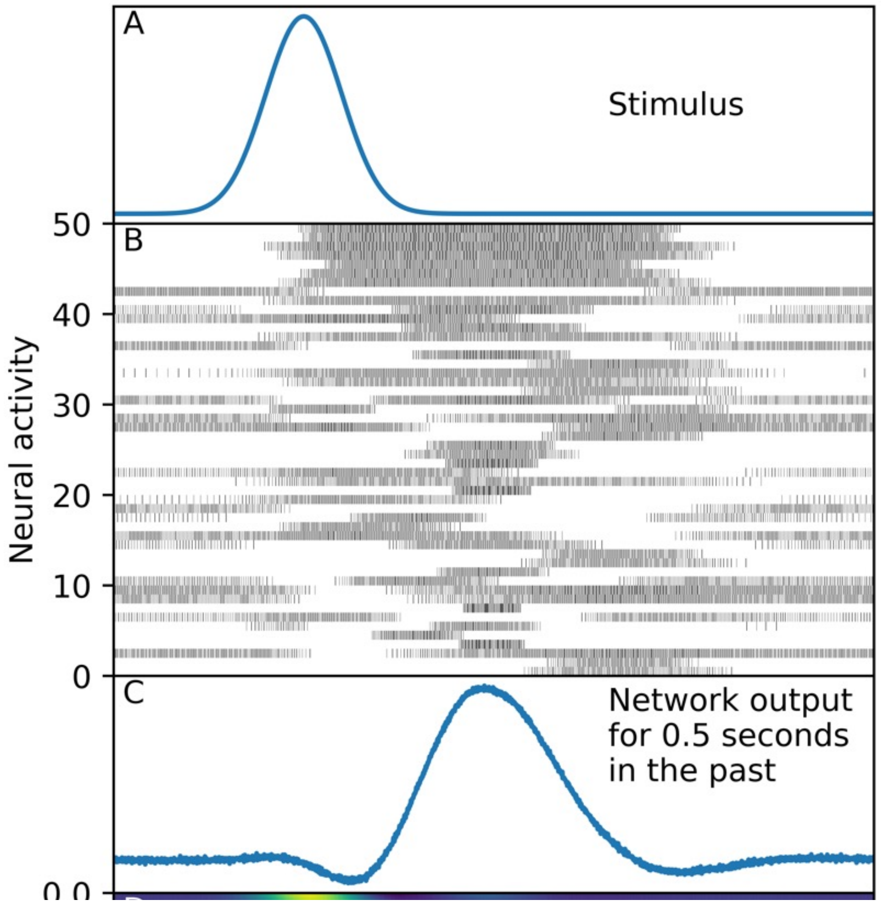
- The **LMU** has this linear system at the core of its architecture.
- This provides state-of-the-art performance and is parameter-efficient
- Neural Connection: Explains time cells, dynamic, spiking (or not)



Time Cells

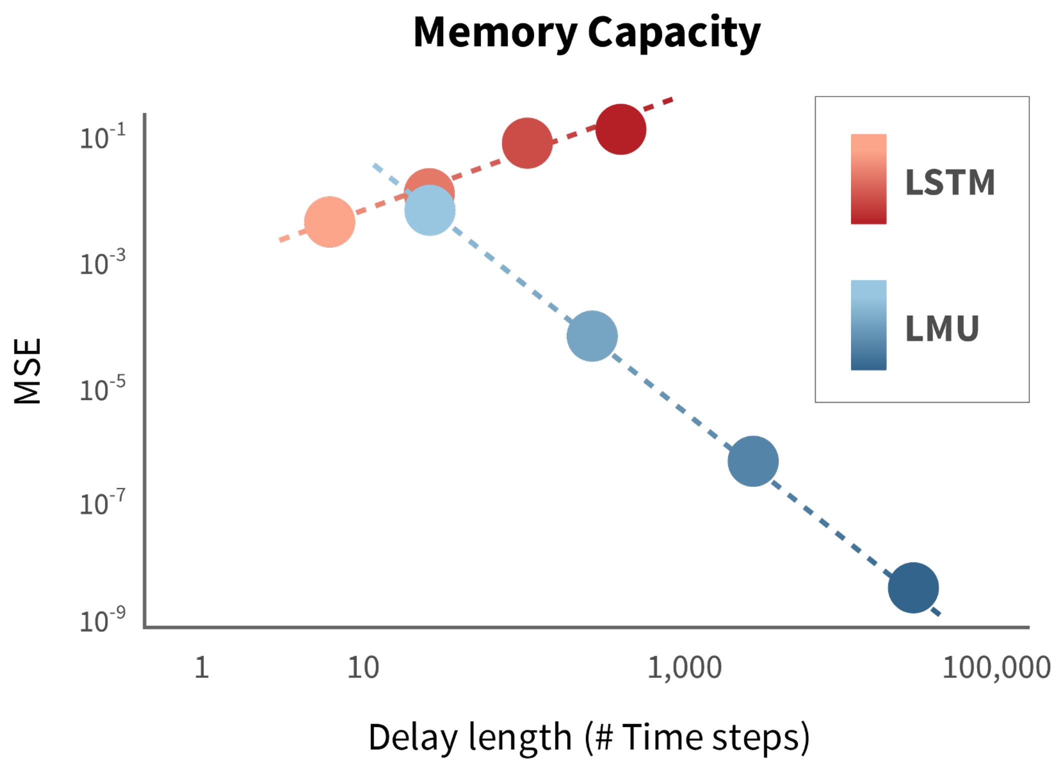


A
B
Time [s]
Cell #
Time [s]
Time [s]



A
Stimulus
B
Neural activity
C
Network output
for 0.5 seconds
in the past

Efficient and Accurate



Better accuracy with **fewer parameters** means using **less power.**

The **LMU** with **500** parameters **outperforms LSTMs** with **41,000** parameters.

LMUs are 1,000,000x **more accurate** while remembering 10,000x **more data** than **LSTMs**.

A **State-of-the-art** Neural Network

Benchmarks: SotA performance on psMNIST

Permuted, sequential MNIST is the **standard benchmark** for all new RNNs

Simplified LMU holds **SotA record** with 98.49% accuracy

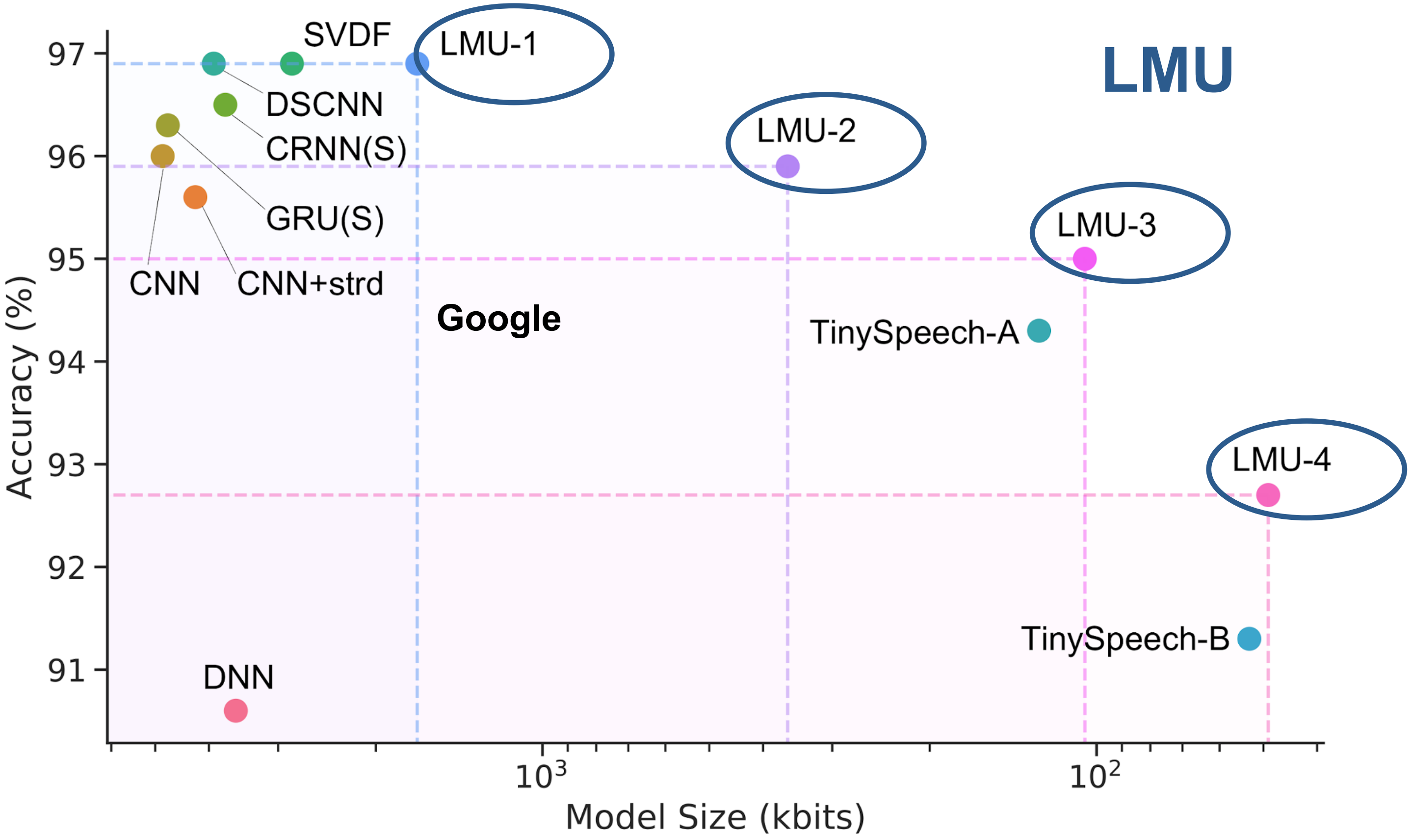
LMU uses **60% fewer params** than all other models

Model	Accuracy
RNN-orth	89.26
RNN-id	86.13
LSTM	89.86
LSTM-chrono	88.43
GRU	92.39
JANET	91.94
SRU	92.49
GORU	87.00
NRU	95.38
Phased LSTM	89.61
LMU	97.15
HiPPO-LegS	98.3
FF-baseline	98.20
Our Model	**98.49**

Simplified **LMU**

Practical: SotA on Keyword Spotting

At the edge

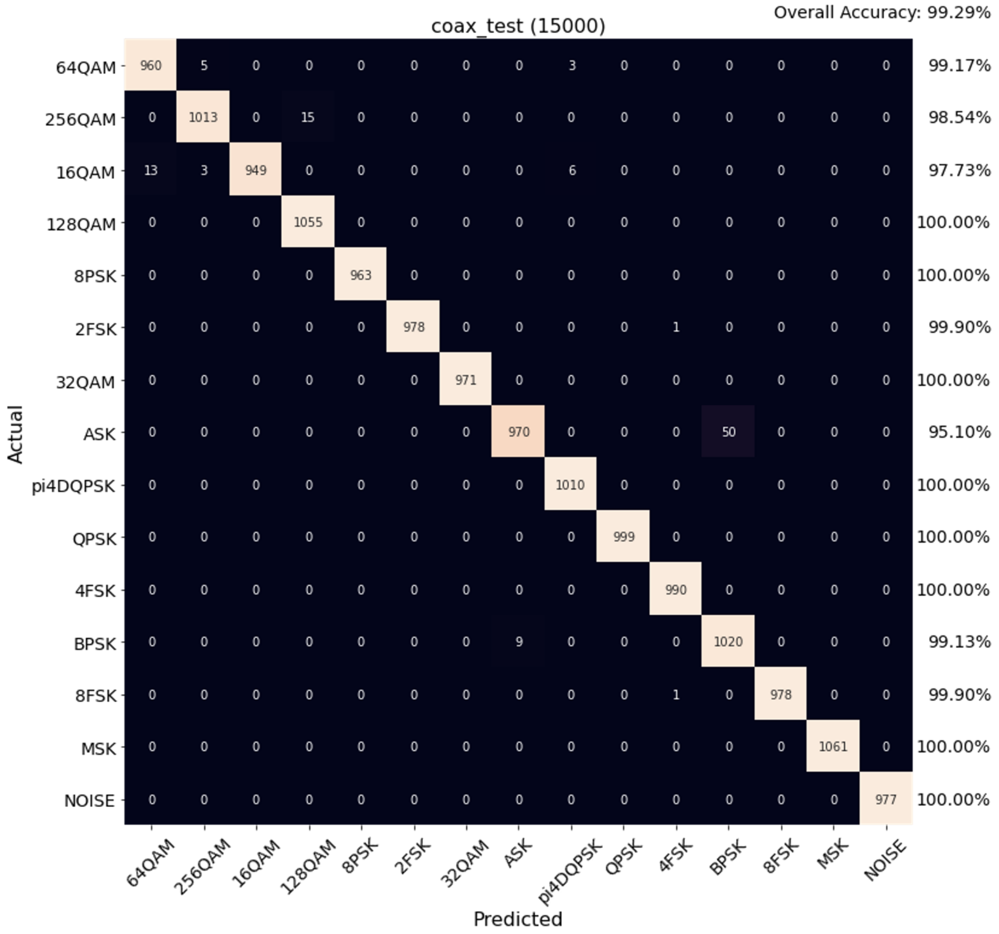


Smaller

LMU gives the **most power efficient streaming** KWS models.

Practical: SotA on RF classification

- LMU has achieved **99.65% on coax** and **95.15% on OTA**
- Coax is **3x** reduction in error
- OTA is **2x** reduction in error
- ~300K params (relatively small), 1 layer **LMU**
- Designed to run **online** (no buffering as with convnet)
- Designed to run at the **edge**



Scalable: SotA for Size and Accuracy

Natural language processing

- IMDB sentiment analysis (160x fewer params)
- QQP semantic similarity (650x fewer params)
- SNLI inferential relations (60x fewer params)

Pre-training for IMDB

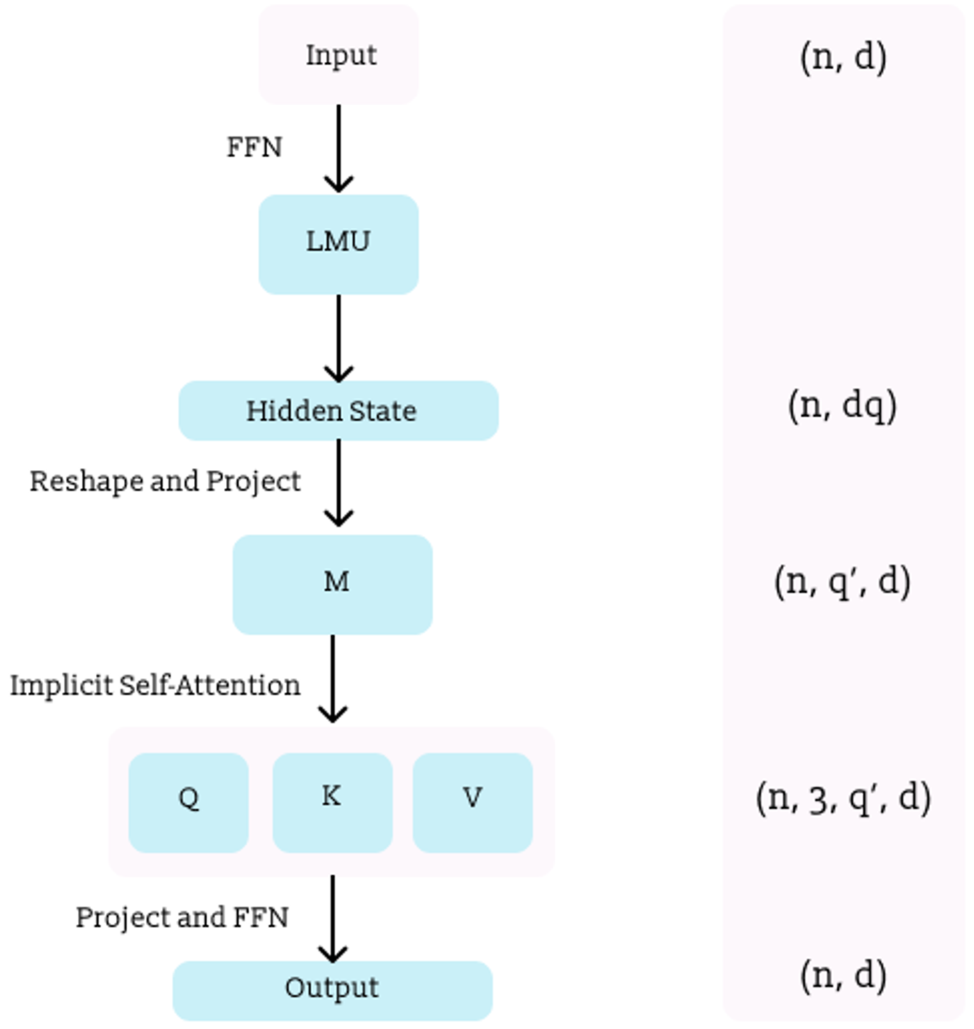
- LSTM Radford et al. (2017) and Transformer Sanh et al. (2019)

Model	IMDB	QQP	SNLI
LSTM	87.29	82.58/81.4	77.6
Our Model	**89.10**	**86.95/85.36**	**78.85**

Model	# parameters (Millions)	Accuracy
LSTM	75	92.88
DistilBERT	66	92.82
Our Model	34	**93.20**

A New **LMU** Architecture for NLP

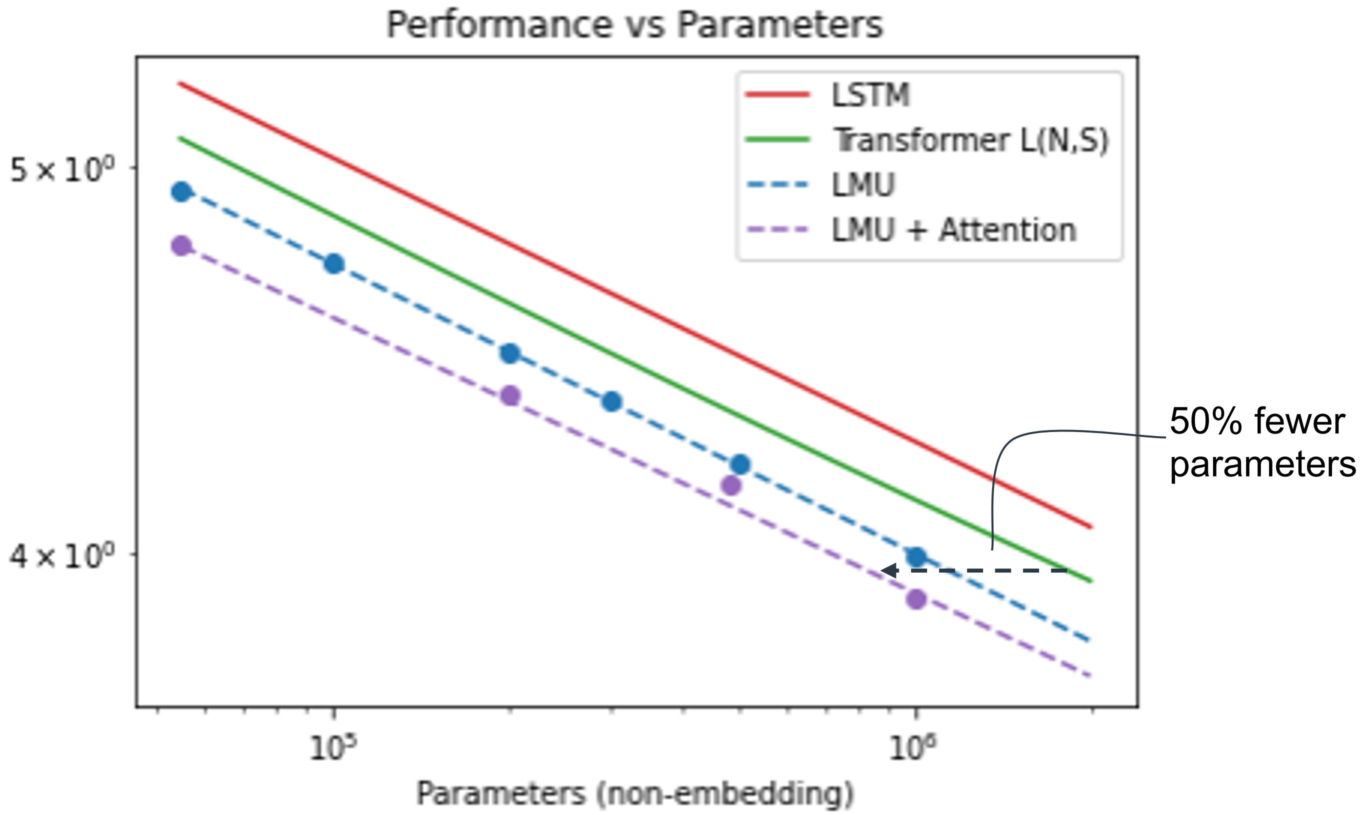
- **Legendre Memory Unit**
  - **Parallel training** with Recurrent Inference
  - **Implicit** Self-Attention
    - Use 'implicit self-attention' on the hidden state at each time-step
    - Choose q' << q exploiting LMU compression



Scalable: SotA for Size and Accuracy

Fundamentally better scaling than transformers (OpenAI, 2021)

- During learning and inference

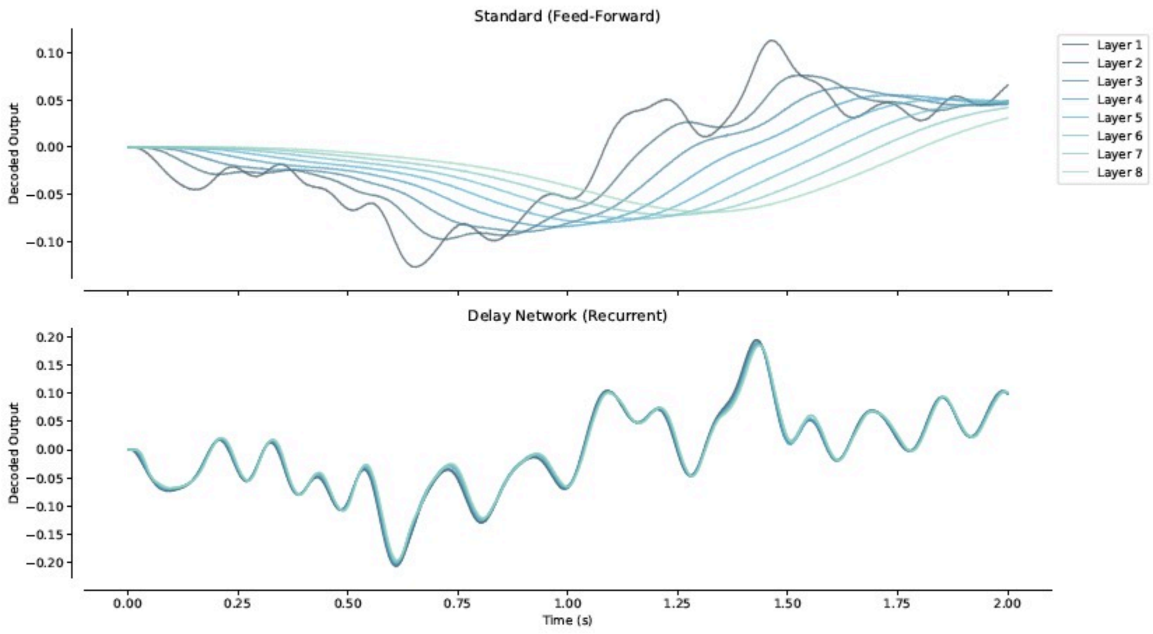


Model	Compute	Memory
Transformer	$O(N^2)$	$O(N^2)$
LMU	$O(N)$	$O(1)$

N = sequence length

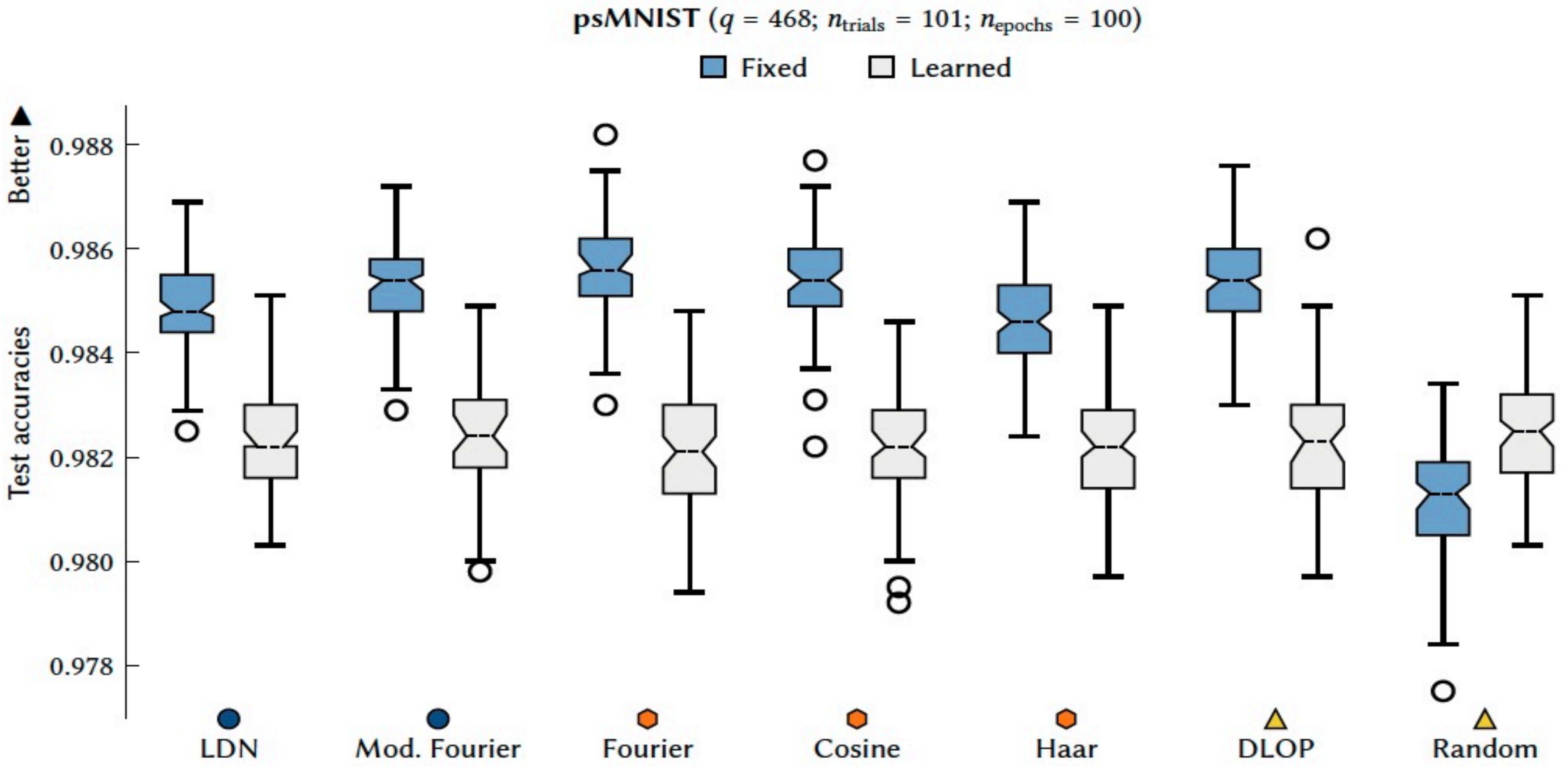
Other applications

- Biosignals analysis (e.g. R-peak detection, arrythmia)
- Network monitoring (SoTA on KDD99)
- Anomaly detection (SoTA on audio DCASE 2020 challenge)
- Nonlinear dynamical system prediction (SoTA on MacKey Glass)
- Signal processing, e.g., instantaneous signal propagation



Other Bases

- You can use other bases, some have advantages (e.g. Mod Four $O(N^{2.3})$)
- Analytic always better than learned (except random)



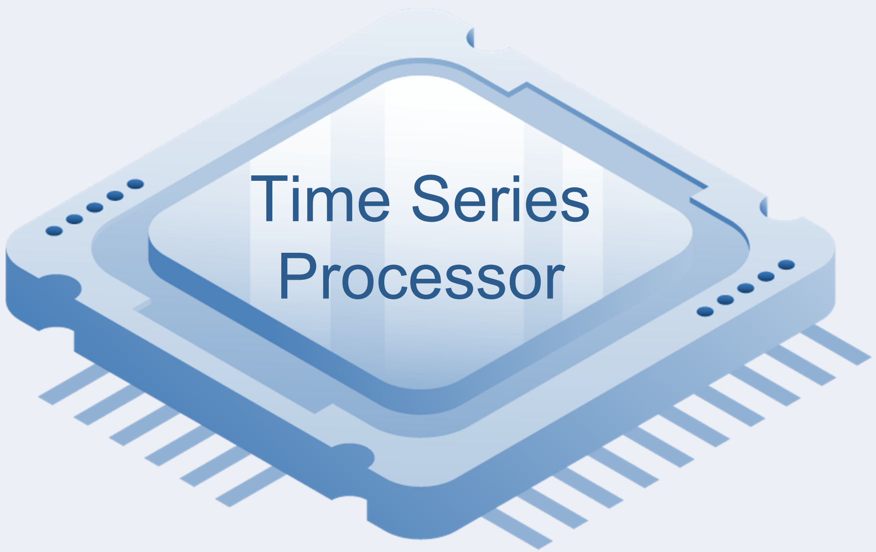
Hardware: AI Time Series Processor

Uses
All time series data processing at the edge, including:
- Edge Full ASR and NLP
- Edge Sensor Data Processing

Reduce edge and cloud costs
Edge ASR and NLP cost reduces from >\$50 down to <\$4 per chip

Saves energy
Full ASR at <25 mW (vs ~5W), battery power IoT

Reduce data center comms and energy costs, increase privacy and reduce latency
>10^6 reduction of data sent from edge, <60ms latency



Replaces CPUs and GPUs at the edge

Further Information

Research, Papers
http://compneuro.uwaterloo.ca

Nengo software, Tutorials, Demo videos
http://nengo.ai

Applied Brain Research
http://appliedbrainresearch.com