Transformers and Structured State Space Sequence (S4) CS480/680 Intro to Machine Learning
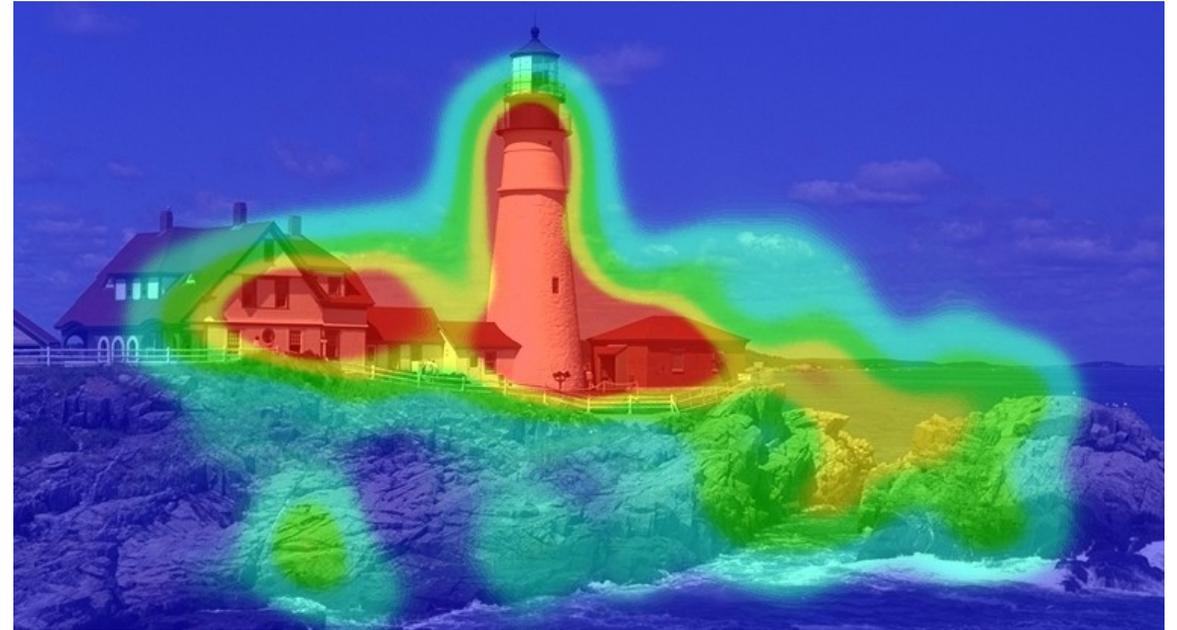
2023-3-9

Pascal Poupart
David R. Cheriton School of Computer Science

UNIVERSITY OF
WATERLOO

Attention

- Attention in Computer Vision

  - 2014: Attention used to highlight important parts of an image that contribute to a desired output

- Attention in NLP

  - 2015: Aligned machine translation

  - 2017: Language modeling with **Transformer networks**

UNIVERSITY OF
WATERLOO

Sequence Modeling

Challenges with RNNs	Transformer Networks
▪ Long range dependencies	▪ Facilitate long range dependencies
▪ Gradient vanishing and explosion	▪ No gradient vanishing and explosion
▪ Large # of training steps	▪ Fewer training steps
▪ Recurrence prevents parallel computation	▪ No recurrence that facilitate parallel computation

UNIVERSITY OF
WATERLOO

Attention Mechanism

- Mimics the retrieval of a **value** v_i for a **query** q based on a **key** k_i in database

- Picture

$$attention(q, \boldsymbol{k}, \boldsymbol{v}) = \sum_i similarity(q, k_i) \times v_i$$
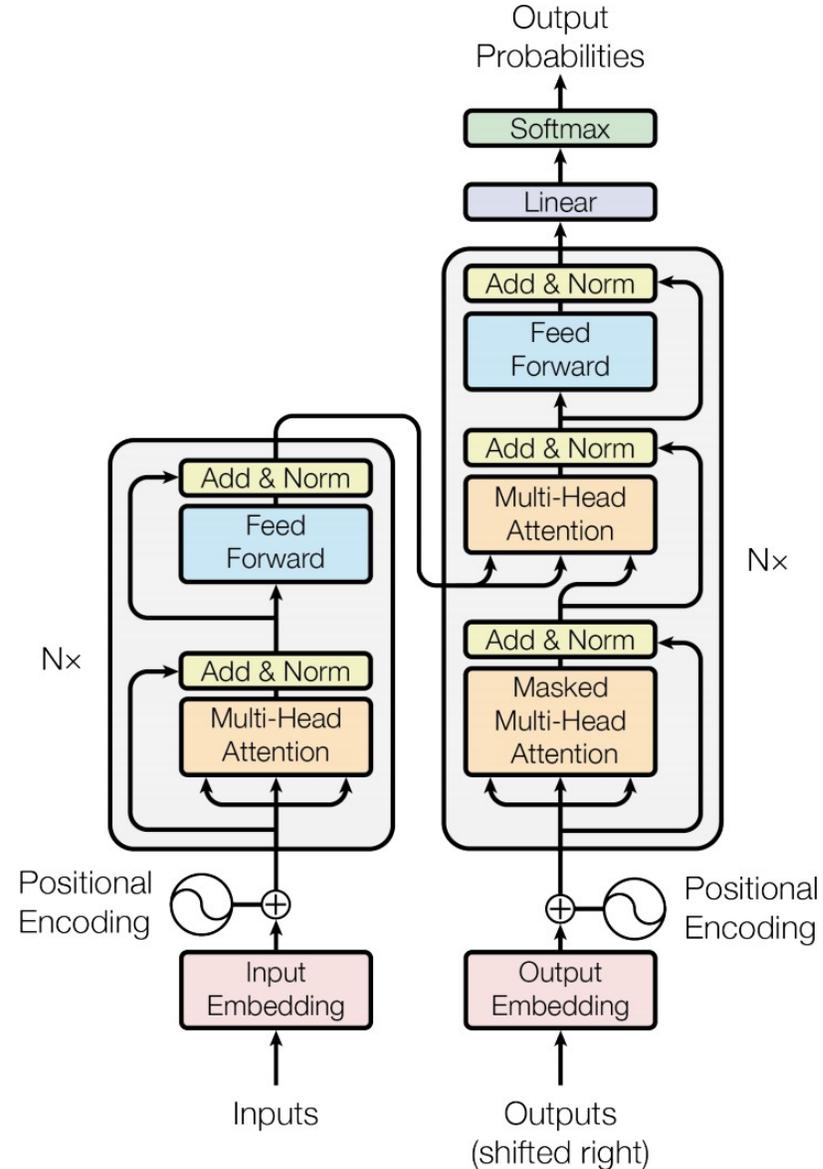
UNIVERSITY OF
WATERLOO

Attention Mechanism (Neural Architecture)

-

- Example: machine translation
  - Query: s_{i-1} (hidden vector for $i-1^{th}$ output word)
  - Key: h_j (hidden vector for j^{th} input word)
  - Value: h_j (hidden vector for j^{th} input word)

UNIVERSITY OF
WATERLOO

Transformer Networ

- Vaswani et al., (2017) Attention is all you need.

- Encoder-decoder based on attention (no recurrence)

Multihead attention

- Multihead attention: compute multiple attentions per query with different weights

$$multihead(Q, K, V) = W^O concat(head_1, head_2, ..., head_h)$$
$$head_i = attention(W_i^Q Q, W_i^K K, W_i^V V)$$
$$attention(Q, K, V) = softmax\left(\frac{Q^T K}{\sqrt{d_k}}\right) V$$

UNIVERSITY OF
WATERLOO

Masked Multi-head attention

- Masked multi-head attention: multi-head where some values are masked (i.e., probabilities of masked values are nullified to prevent them from being selected).

- When decoding, an output value should only depend on previous outputs (not future outputs). Hence we mask future outputs.

$$attention(Q, K, V) = softmax\left(\frac{Q^T K}{\sqrt{d_k}}\right) V$$

$$maskedAttention(Q, K, V) = softmax\left(\frac{Q^T K + M}{\sqrt{d_k}}\right) V$$

where M is a mask matrix of 0's and $-\infty$'s

UNIVERSITY OF
WATERLOO

Other layers

- Layer normalization:
  - Normalize values in each layer to have 0 mean and 1 variance
  - For each hidden unit h_i compute $h_i \leftarrow \frac{g}{\sigma}(h_i - \mu)$

    where g is a variable, $\mu = \frac{1}{H}\sum_{i=1}^{H} h_i$ and $\sigma = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(h_i - \mu)^2}$
  - This reduces "covariate shift" (i.e., gradient dependencies between each layer) and therefore fewer training iterations are needed

- Positional embedding (embedding to distinguish each position):
  $$PE_{position,2i} = \sin(position/10000^{2i/d})$$
  $$PE_{position,2i+1} = \cos(position/10000^{2i/d})$$

UNIVERSITY OF
WATERLOO

Comparison

- Attention reduces sequential operations and maximum path length, which facilitates long range dependencies

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

| Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

UNIVERSITY OF
WATERLOO

Results

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.0**	$2.3 \cdot 10^{19}$	

UNIVERSITY OF
WATERLOO

GPT and GPT-2

- Radford et al., (2018) Language models are unsupervised multitask learners
  - Decoder transformer that predicts next word based on previous words by computing $P(x_t|x_{1..t-1})$
  - SOTA in "zero-shot" setting for 7/8 language tasks (where zero-shot means no task training, only unsupervised language modeling)

UNIVERSITY OF
WATERLOO
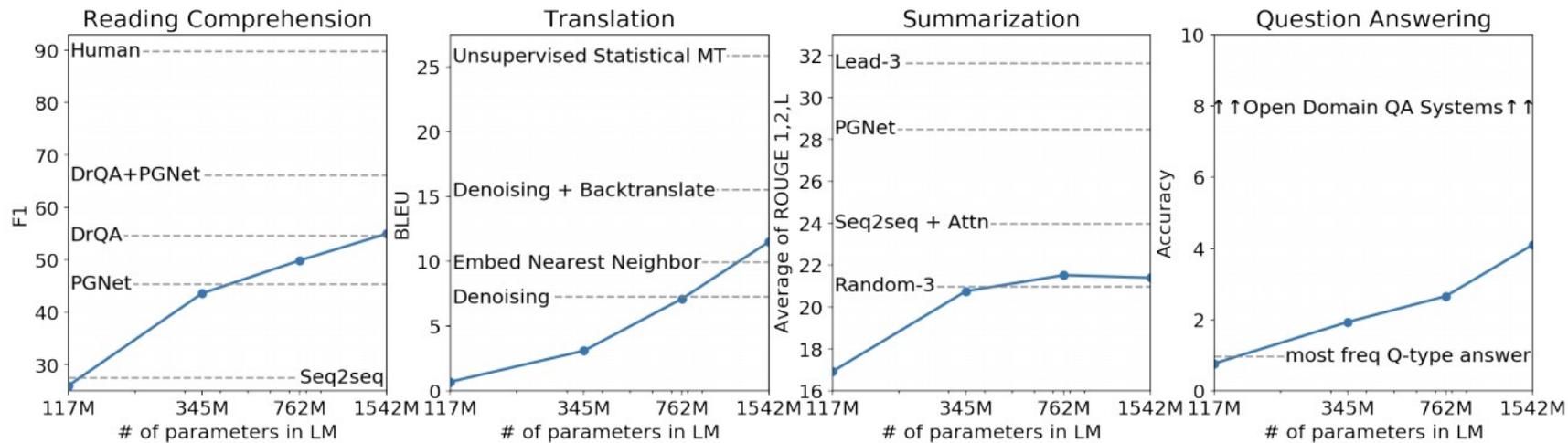
BERT (Bidirectional Encoder Representations from Transformers)

- Devlin et al., (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

  — Decoder transformer that predicts a missing word based on surrounding words by computing $P(x_t | x_{1..t-1, t+1..T})$

  — Mask missing word with masked multi-head attention
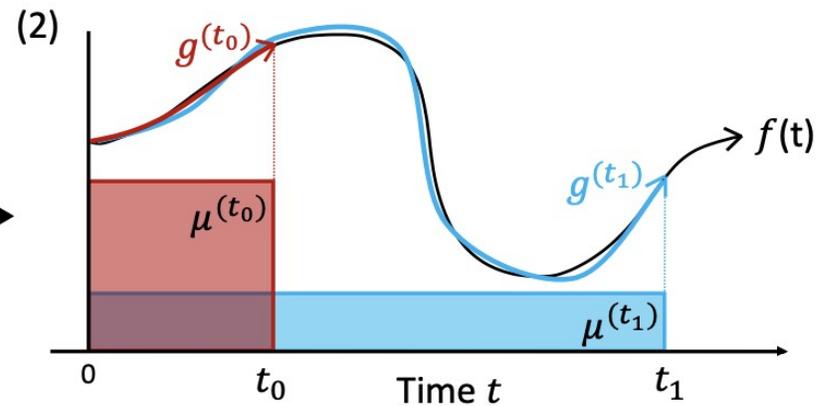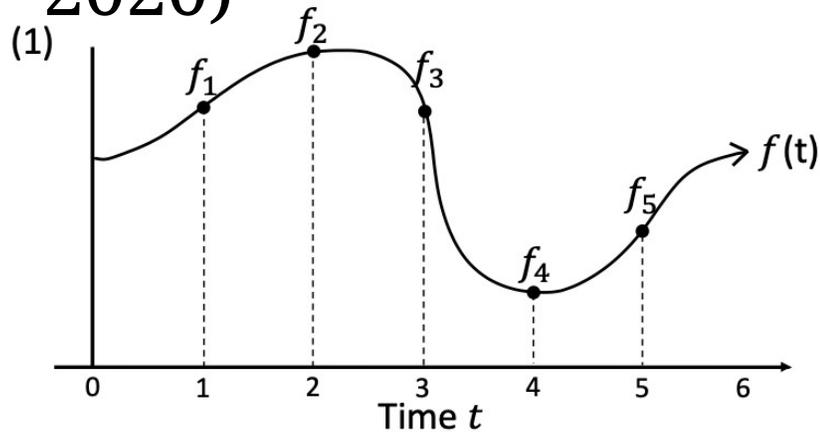
  — Improved state of the art on 11 tasks

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT$_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT$_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

UNIVERSITY OF
WATERLOO

Limitation

- Transformers scale quadratically with sequence length
  - In practice, sequence length often limited to 512 tokens

- How can we process long sequences?
  - Hierarchy of transformers (i.e., words→sentences→documents→corpus)
  - Approximate transformers (i.e., longformer, reformer, performer, etc.)
  - **Structured State Space Sequence (S4) model**

- S4: Very recent approach (Gu, Goel & Re, ICLR 2022)
  - Potential to displace transformers
  - S4 achieved state of the art on Long Range Arena benchmark
  - Scales linearly with sequence length

UNIVERSITY OF
WATERLOO

Structured State Space Sequence (S4) Model

- HiPPO: high-order polynomial projection operators (Gu et al., 2020)

(1)

f_1, f_2, f_3, f_4, f_5 — $f(t)$

Time t

proj$_t$

(2)

$g^{(t_0)}$ $\quad$ $g^{(t_1)}$ $\quad$ $f(t)$

$\mu^{(t_0)}$ $\quad$ $\mu^{(t_1)}$

t_0 $\quad$ Time t $\quad$ t_1

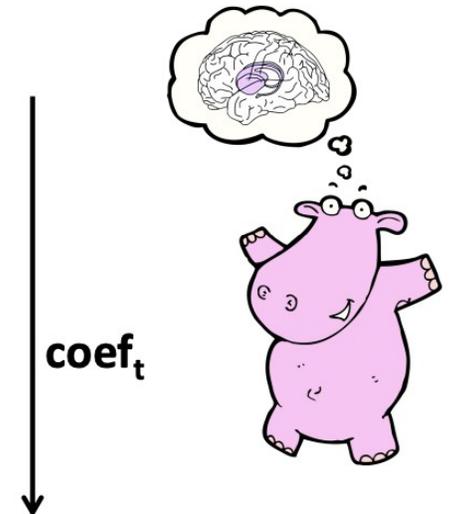**coef$_t$**

(4)

Discrete-time HiPPO Recurrence

$$c_{k+1} = A_k c_k + B_k f_k$$

discretize

(3)

$$c(t_0) = \begin{bmatrix} 0.1 \\ -1.1 \\ 3.7 \\ 2.5 \end{bmatrix} \qquad c(t_1) = \begin{bmatrix} 1.5 \\ 2.9 \\ -0.3 \\ 2.0 \end{bmatrix}$$

Continuous-time HiPPO ODE

$$\frac{d}{dt} c(t) = A(t) c(t) + B(t) f(t)$$

UNIVERSITY OF WATERLOO
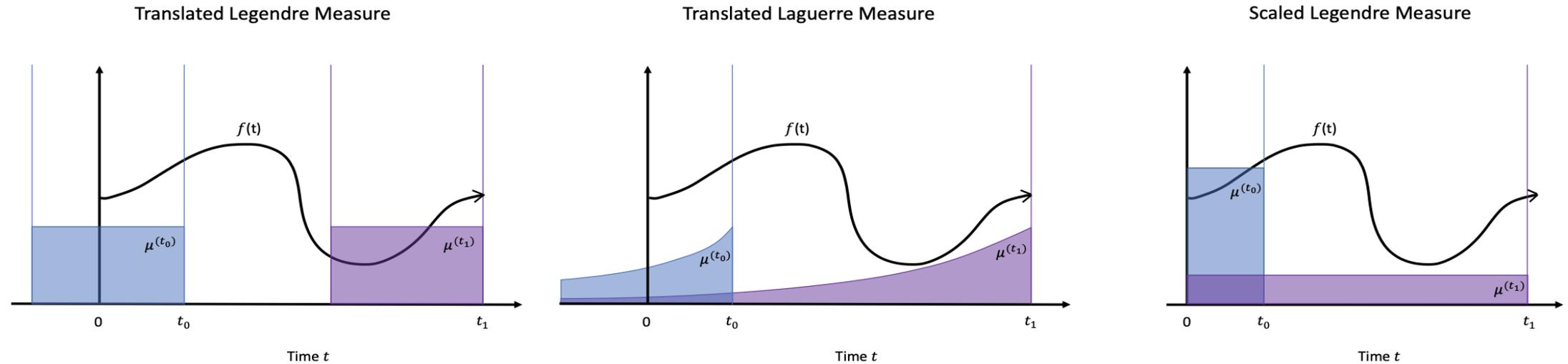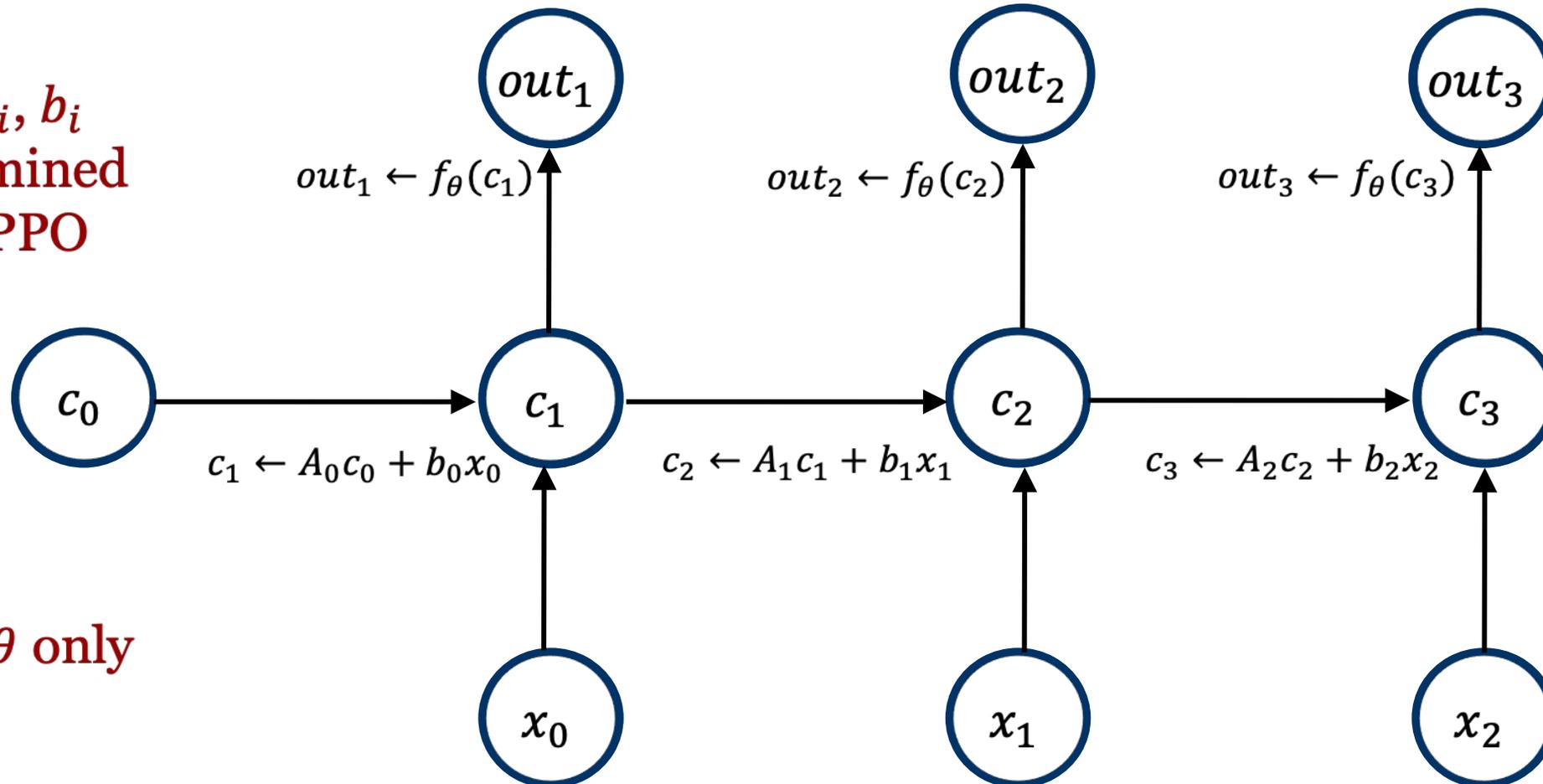
Measures (importance given to past history)

Figure 5: **Illustration of HiPPO measures.** At time t_0, the history of a function $f(x)_{x \leq t_0}$ is summarized by polynomial approximation with respect to the measure $\mu^{(t_0)}$ (blue), and similarly for time t_1 (purple). (Left) The Translated Legendre measure (**LegT**) assigns weight in the window $[t - \theta, t]$. For small t, $\mu^{(t)}$ is supported on a region $x < 0$ where f is not defined. When t is large, the measure is not supported near 0, causing the projection of f to forget the beginning of the function. (Middle) The Translated Laguerre (**LagT**) measure decays the past exponentially. It does not forget, but also assigns weight on $x < 0$. (Right) The Scaled Legendre measure (**LegS**) weights the entire history $[0, t]$ uniformly.

UNIVERSITY OF
WATERLOO

RNN with HiPPO

NB: A_i, b_i determined by HiPPO

$out_1 \leftarrow f_\theta(c_1)$

$out_2 \leftarrow f_\theta(c_2)$

$out_3 \leftarrow f_\theta(c_3)$

$c_1 \leftarrow A_0 c_0 + b_0 x_0$

$c_2 \leftarrow A_1 c_1 + b_1 x_1$

$c_3 \leftarrow A_2 c_2 + b_2 x_2$

Train θ only

UNIVERSITY OF WATERLOO

Computational Complexity

- S4 scales better than CNNs, RNNs and Transformers

Table 1: Complexity of various sequence models in terms of sequence length (L), batch size (B), and hidden dimension (H); tildes denote log factors. Metrics are parameter count, training computation, training space requirement, training parallelizability, and inference computation (for 1 sample and time-step). For simplicity, the state size N of S4 is tied to H. Bold denotes model is theoretically best for that metric. Convolutions are efficient for training while recurrence is efficient for inference, while SSMs combine the strengths of both.

	Convolution[3]	Recurrence	Attention	S4
Parameters	LH	$\boldsymbol{H^2}$	$\boldsymbol{H^2}$	$\boldsymbol{H^2}$
Training	$\boldsymbol{\tilde{L}H(B+H)}$	BLH^2	$B(L^2H + LH^2)$	$\boldsymbol{BH(\tilde{H}+\tilde{L})+B\tilde{L}H}$
Space	$\boldsymbol{BLH}$	$\boldsymbol{BLH}$	$B(L^2+HL)$	$\boldsymbol{BLH}$
Parallel	**Yes**	No	**Yes**	**Yes**
Inference	LH^2	$\boldsymbol{H^2}$	$L^2H + H^2L$	$\boldsymbol{H^2}$

From Gu, Goel & Re (2022)

UNIVERSITY OF
WATERLOO

Results: Long Range Arena

Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.56
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Linear Trans.	16.13	65.90	53.09	42.34	75.30	✗	50.46
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
FNet	35.33	65.11	59.61	38.67	77.80	✗	54.42
Nyströmformer	37.15	65.52	79.56	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	47.38	77.72	✗	59.37
S4	**59.60**	**86.82**	**90.90**	**88.65**	**94.20**	**96.35**	**86.09**

From Gu, Goel & Re (2022)

UNIVERSITY OF
WATERLOO

Results: Speech and Images

Table 5: (**SC10 classification**) Transformer, CTM, RNN, CNN, and SSM models. (*MFCC*) Standard pre-processed MFCC features (length 161). (*Raw*) Unprocessed signals (length 16000). (*0.5×*) Frequency change at test time. ✗ denotes not applicable or computationally infeasible on single GPU. *Please read Appendix D.5 before citing this table.*

	MFCC	RAW	0.5×
Transformer	90.75	✗	✗
Performer	80.85	30.77	30.68
ODE-RNN	65.9	✗	✗
NRDE	89.8	16.49	15.12
ExpRNN	82.13	11.6	10.8
LipschitzRNN	88.38	✗	✗
CKConv	**95.3**	71.66	<u>65.96</u>
WaveGAN-D	✗	<u>96.25</u>	✗
LSSL	93.58	✗	✗
S4	<u>93.96</u>	**98.32**	**96.30**

From Gu, Goel & Re (2022)

Table 6: (**Pixel-level 1-D image classification**) Comparison against reported test accuracies from prior works (Transformer, RNN, CNN, and SSM models). Extended results and citations in Appendix D.

	sMNIST	pMNIST	sCIFAR
Transformer	98.9	97.9	62.2
LSTM	98.9	95.11	63.01
r-LSTM	98.4	95.2	72.2
UR-LSTM	99.28	96.96	71.00
UR-GRU	99.27	96.51	74.4
HiPPO-RNN	98.9	98.3	61.1
LMU-FFT	-	98.49	-
LipschitzRNN	99.4	96.3	64.2
TCN	99.0	97.2	-
TrellisNet	99.20	98.13	73.42
CKConv	99.32	98.54	63.74
LSSL	<u>99.53</u>	**98.76**	<u>84.65</u>
S4	**99.63**	<u>98.70</u>	**91.13**

UNIVERSITY OF
WATERLOO