IT594: - DEEP NEURAL NLP

Dhirubhai Ambani Institute of Information and Communication Technology

DA-IICT

ENGINEERS WITH
SOCIAL RESPONSIBILITY

Term Paper Presentation

MEGA: Moving Average Equipped Gated Attention

Sarvesh Bagwe (202211006)
Vedant Dave (202211042)

Kashyap Halavadia(202003040)
Hiren Thakkar (202211074)

Course Instructor:
Prof. Sourish Dasgupt

Problems/ limitations with Transformers

1) **Weak inductive bias:**
   a) Almost no prior knowledge of dependency patterns. Trying learn directly from the data at every timestep.
   b) Position information only from absolute/relative positional embeddings.

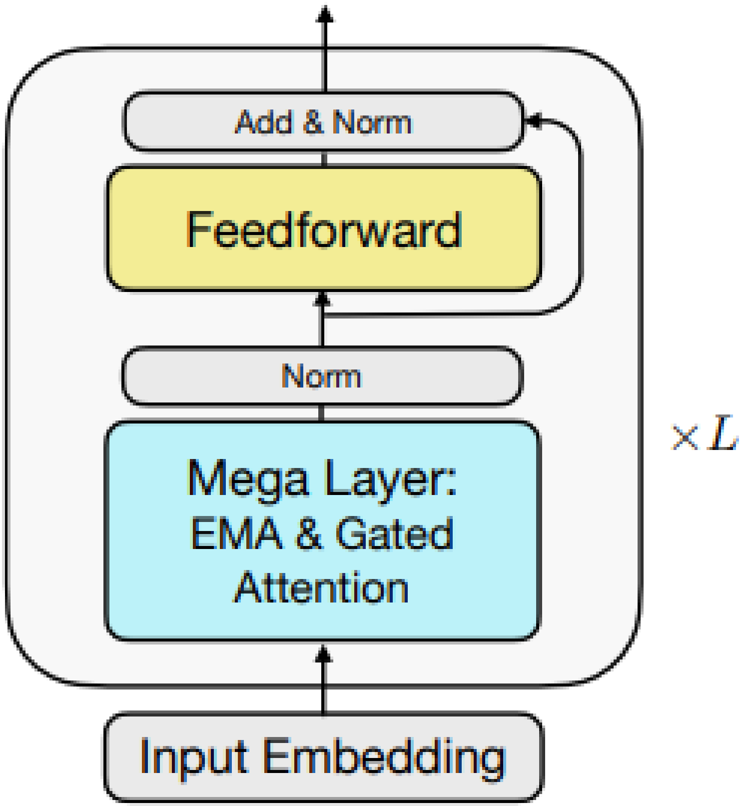
2) **Quadratic complexity:**
   a) Time complexity for computing attention matrix as well as space complexity for storing it is quadratic in terms of the length of the input sequence.
   b) O(h.n.n) where h is the no of attention heads and n is the length of the input sequence.

Inductive Bias

- **RNN :**
  - Strong and clear inductive bias.
  - Uses hidden states to capture local dependencies.
  - Recurrently model dependencies for effective learning.
- **Transformers' Attention Mechanism:**
  - Assumes no prior knowledge of dependencies.
  - Learns pairwise attention weights for input tokens.
  - Challenging for recognizing patterns, especially in long sequences.
- **Transformers' Limitations:**
  - It is inefficient for long sequence modeling.

MEGA: Executive Summary

- Effective and efficient drop-in replacement of attention for long sequence modelling.
- Exponential Moving Average (EMA)
- Mega-chunk: linear complexity of time and space.



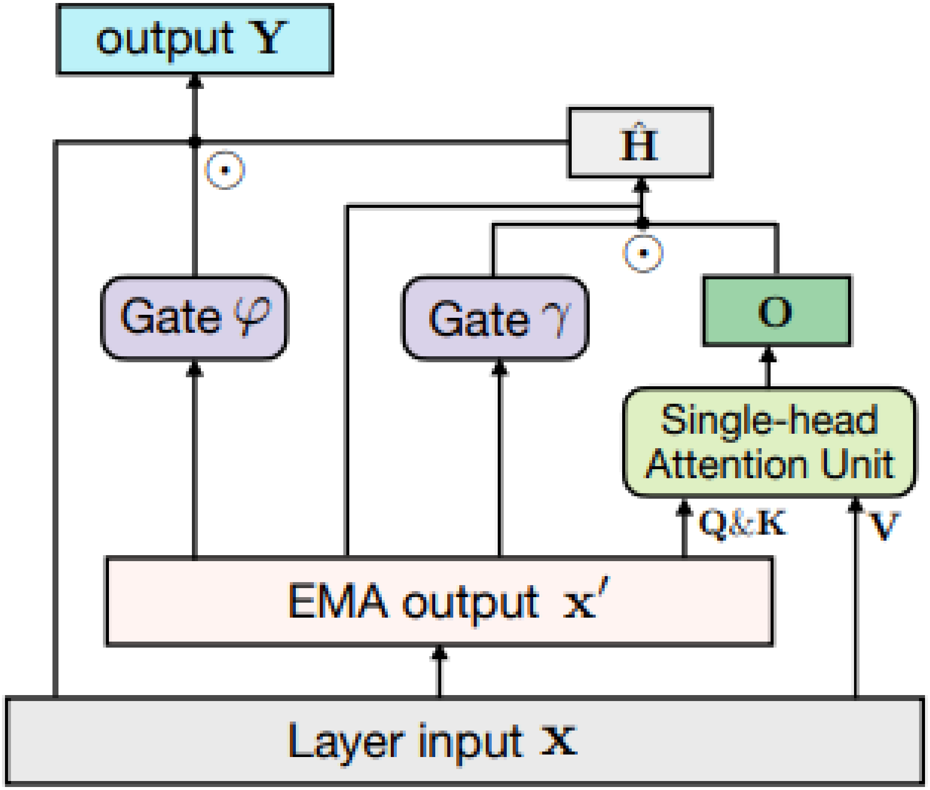
(a) Mega architecture.

MEGA: Architecture Outline

- **Exponential Moving Average (EMA)**
  - Local dependencies that decay exponentially over time .
  - Incorporates stronger inductive bias into the attention.

- **Single-headed Gated Attention**
  - Adding a reset gate to the attention output.
  - Theoretically proving that single-head gated attention is as expressive as multi-head one.

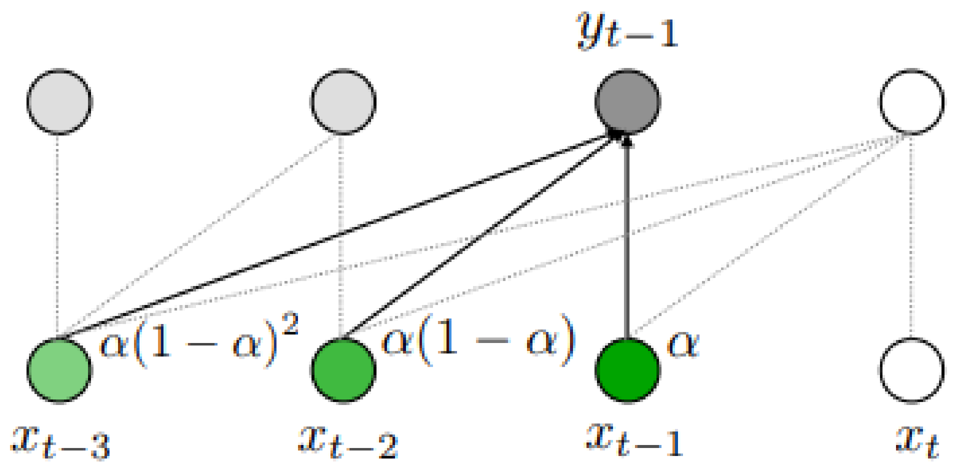
- **Mega-Chunk**
  - Applying attention to local chunks of fixed length.
  - Reducing quadratic complexity to linear.

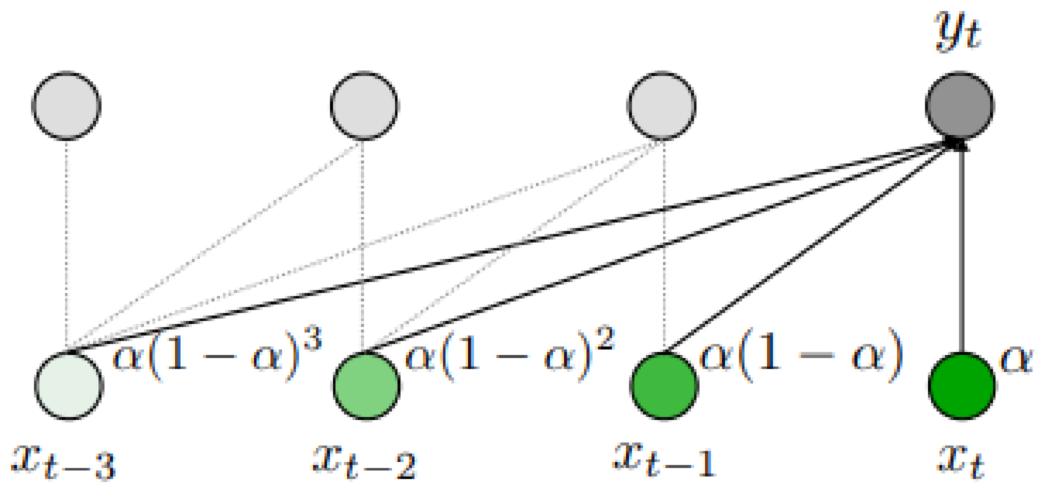


(b) Mega layer.

Exponential Moving Average (EMA)

- **Notations:** Assuming 1-dim input sequence $\boldsymbol{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathbb{R}^{n \times d}$
- **EMA:** $\mathbf{y}_t = \boldsymbol{\alpha} \odot \mathbf{x}_t + (1 - \boldsymbol{\alpha}) \odot \mathbf{y}_{t-1}, \quad \boldsymbol{\alpha} \in (0,1)^d$
- **Damped EMA:** $\mathbf{y}_t = \boldsymbol{\alpha} \odot \mathbf{x}_t + (1 - \boldsymbol{\alpha} \odot \boldsymbol{\delta}) \odot \mathbf{y}_{t-1},\ \boldsymbol{\delta} \in (0,1)^d$ is the damping factor.



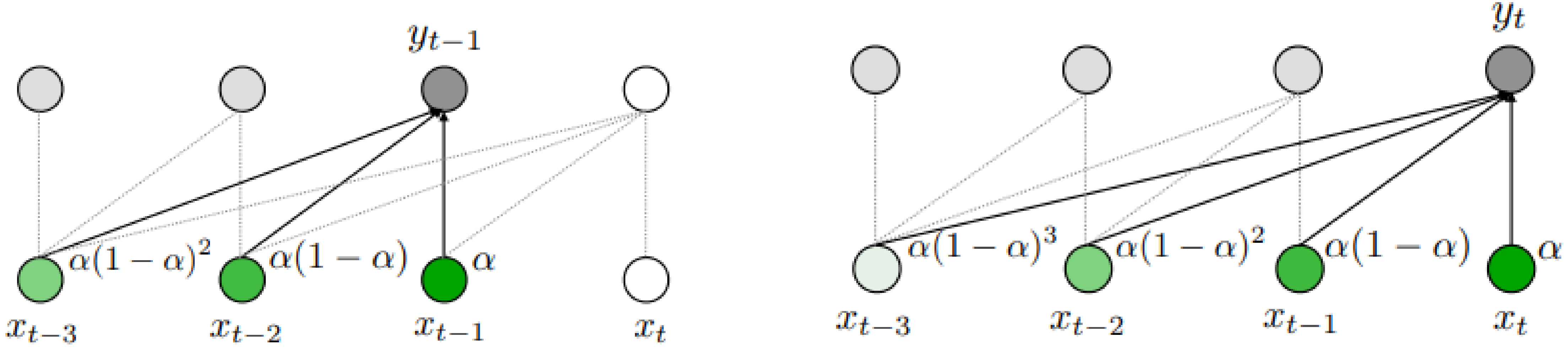


Efficient Algorithm for EMA

- Efficiently compute EMA outputs of all tokens in parallel.

$$\mathbf{y}_t = \boxed{\boldsymbol{\alpha}} \odot \mathbf{x}_t + \boxed{(1 - \boldsymbol{\alpha} \odot \boldsymbol{\delta})} \odot \mathbf{y}_{t-1}$$

EMA weights are input independent



We can compute the weights for each input tokens in advance and compute EMAs with FFTs.

Single-headed Gated Attention in Mega

- **Adding a reset gate to the attention output**

Step1: $\boldsymbol{X}' = \mathrm{EMA}(\boldsymbol{X})$ **from the EMA layer:**

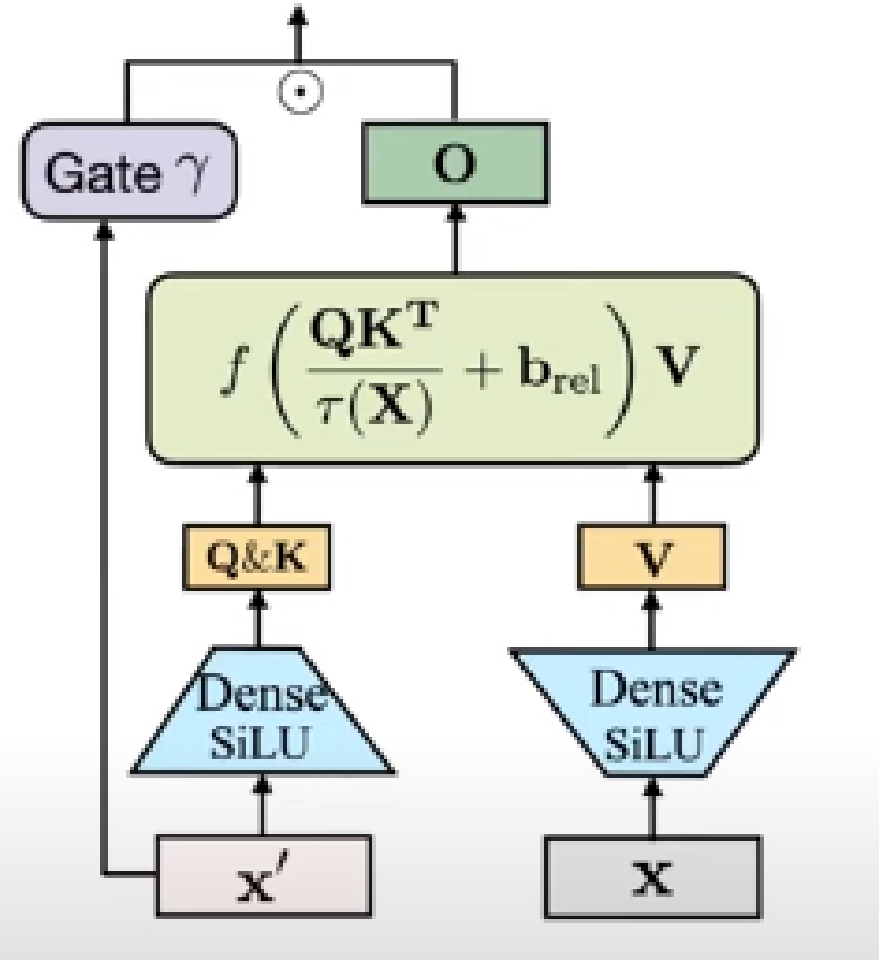
$$\boldsymbol{O} = f\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\tau(\boldsymbol{X})} + \boldsymbol{b}_{\mathrm{rel}}\right)\boldsymbol{V}$$

Step2: Attention:

$$\gamma = \mathcal{G}(X)$$

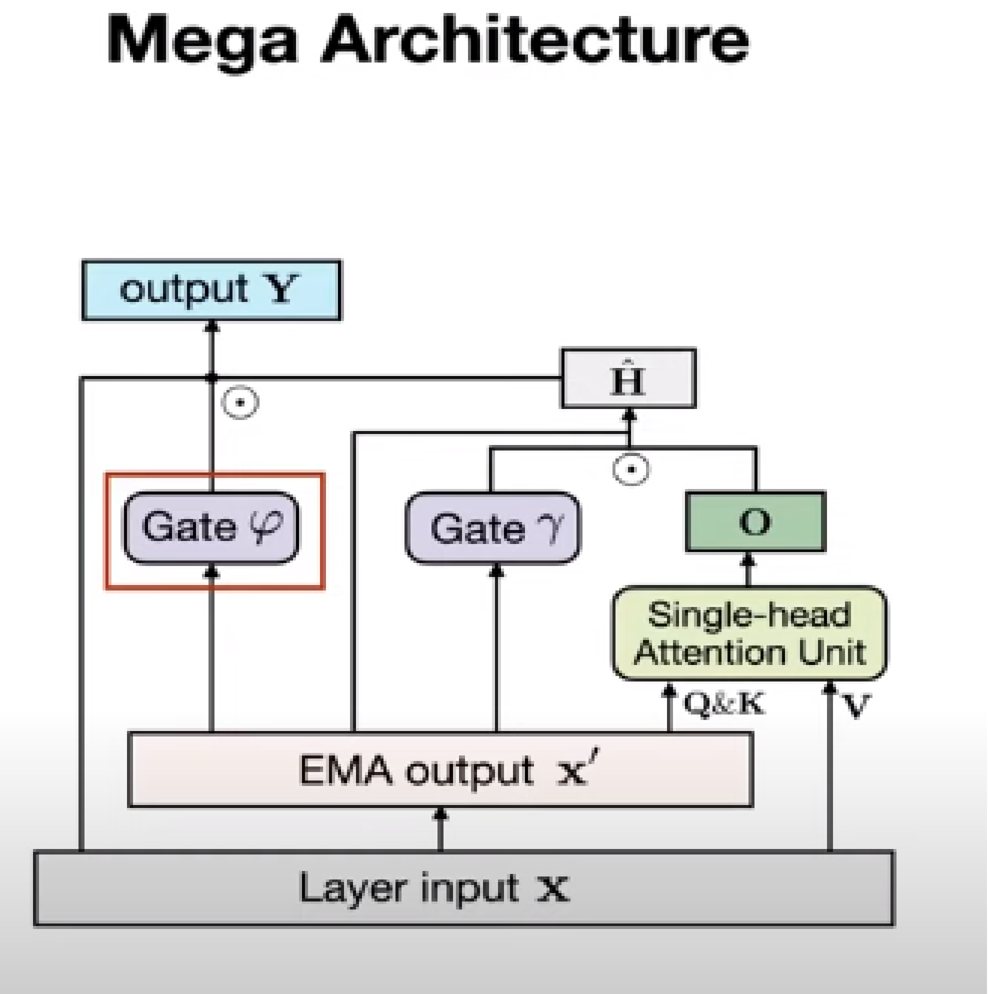
Step3: Gated Attention:

$$O_{\mathrm{SHGA}} = O \odot \gamma$$



Single-headed gated attention is as expressive as multi-head one.

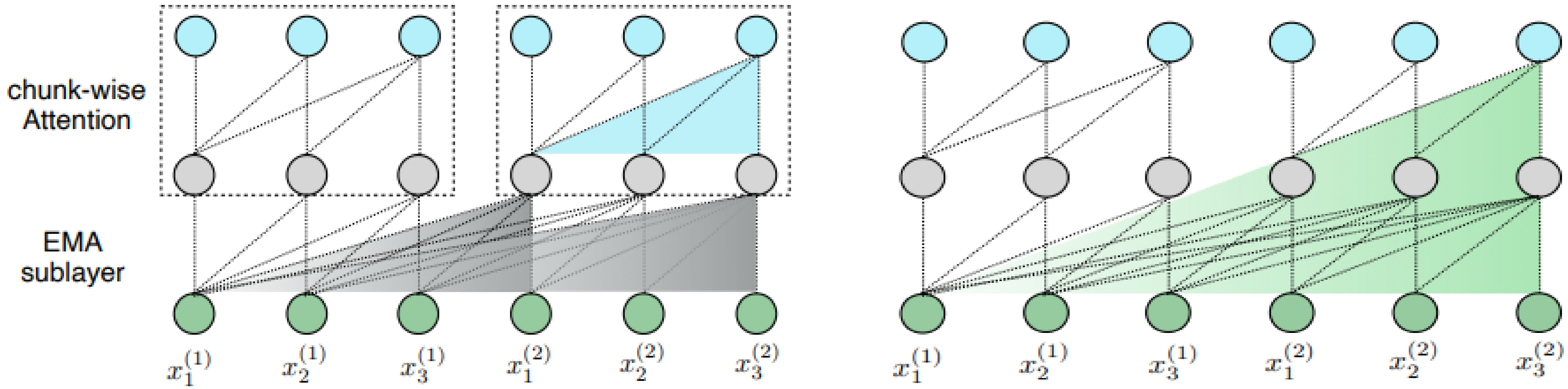
Mega Architecture: Reset and Update gate



$$Y = \phi \odot \hat{H} + (1 - \phi) \odot X$$

Mega-Chunk: Efficient Mega

- Split input sequences into multiple chunks with fixed length.
- Applying attention to each chunk
    - Linear complexity and easy implementation
    - But will we lose contextual information between chunks?
    - **Fortunately, EMA preserves the information from previous chunks.**



Model Evaluation: Experiments

- **Long Range Arena (LRA):**
  - 3 tasks on byte-level text classification
  - 3 tasks on pixel-level image classification
- **Language Modeling:**
  - Enwiki8 (character-level)
  - WikiText-103 (word-level)
- **Machine translation:**
  - VMT’14 English-German
- **Image Classification:**
  - ImageNet-1K
- **Raw Speech Classification:**
  - Speech commands

Experimental Results

	LRA	WMT'14	WikiText-103	ImageNet	Raw-SC
XFM	59.24	27.68	18.66	81.80	31.24
S4	85.86	–	20.95	–	**97.50**
Mega	**88.21**	**29.01**	**18.07**	**82.31**	97.30

Analysis on LRA: Accuracy and Efficiency

- A benchmark of 6 sequence tasks for long range sequence modelling. Intentionally designed to be challenging.

Models	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg.	Speed	Mem.
XFM	36.37	64.27	57.46	42.44	71.40	✗	54.39	–	–
XFM‡	37.11	65.21	79.14	42.94	71.83	✗	59.24	1×	1×
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.67	0.8×	0.24×
Linformer	35.70	53.94	52.27	38.56	76.34	✗	51.36	5.5×	0.10×
BigBird	36.05	64.02	59.29	40.83	74.87	✗	55.01	1.1×	0.30×
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.41	**5.7**×	**0.11**×
Luna-256	37.98	65.78	79.56	47.86	78.55	✗	61.95	4.9×	0.16×
S4-v1	58.35	76.02	87.09	87.26	86.05	88.10	80.48	–	–
S4-v2	59.60	86.82	90.90	88.65	94.20	96.35	86.09	–	–
S4-v2‡	59.10	86.53	90.94	88.48	94.01	96.07	85.86	4.8×	0.14×
MEGA	**63.14**	**90.43**	**91.25**	**90.44**	**96.01**	**97.98**	**88.21**	2.9×	0.31×
MEGA-chunk	58.76	90.19	90.97	85.80	94.41	93.81	85.66	5.5×	0.13×

Thank you for your time!