# Challenges in Long Sequence Encoding and State-space Models

Yao Fu

University of Edinburgh

25th August 2022

# Guidelines

Long Sequence Encoding

The NLP Approach

State-space Models

Mathematical Principles of State-space Models

Challenges

# Before we start

The major content of this talk is not my own work, but many techniques are related to my own work.

Encoding long documents is important, open, and unsolved

The goal of this talk is to examine the problem, discuss existing techniques, identify challenges and opportunities

# Guidelines

Long Sequence Encoding

The NLP Approach

State-space Models

Mathematical Principles of State-space Models

Challenges

# Long Sequence Encoding

Given a sequence $x_1, x_2, \ldots, x_T$, T being very long, e.g., T =16000, find an encoder architecture

$$r_1, r_2, \ldots, r_T = \text{Enc}(x_1, x_2, \ldots, x_T)$$

Such that:

- $r_{1:T}$ are effective for downstream task (sequence classification or sequence to sequence)
- The encoder itself is computationally (GPU-memory) efficient

# Long Sequence Encoding

This is not decoding long sequences

Encoding is about finding effective and efficient representation, current challenge is (still) neural architecture engineering

Decoding long sequence is more about planning and generation

# Guidelines

Long Sequence Encoding

The NLP Approach

State-space Models

Mathematical Principles of State-space Models

Challenges

# Transformers attention complexity: $O(T^2)$
# Efficient Xformer $O(T)$ or $O(T \log T)$

| Model | ListOps | Text | Retrieval | Image | Pathfinder | Path-X | Avg |
|---|---|---|---|---|---|---|---|
| Transformer | 36.37 | 64.27 | 57.46 | 42.44 | 71.40 | FAIL | <u>54.39</u> |
| Local Attention | 15.82 | 52.98 | 53.39 | 41.46 | 66.63 | FAIL | 46.06 |
| Sparse Trans. | 17.07 | 63.58 | **59.59** | **44.24** | 71.71 | FAIL | 51.24 |
| Longformer | 35.63 | 62.85 | 56.89 | 42.22 | 69.71 | FAIL | 53.46 |
| Linformer | 35.70 | 53.94 | 52.27 | 38.56 | <u>76.34</u> | FAIL | 51.36 |
| Reformer | **37.27** | 56.10 | 53.40 | 38.07 | 68.50 | FAIL | 50.67 |
| Sinkhorn Trans. | 33.67 | 61.20 | 53.83 | 41.23 | 67.45 | FAIL | 51.39 |
| Synthesizer | <u>36.99</u> | 61.68 | 54.67 | 41.61 | 69.45 | FAIL | 52.88 |
| BigBird | 36.05 | 64.02 | <u>59.29</u> | 40.83 | 74.87 | FAIL | **55.01** |
| Linear Trans. | 16.13 | **65.90** | 53.09 | 42.34 | 75.30 | FAIL | 50.55 |
| Performer | 18.01 | <u>65.40</u> | 53.82 | <u>42.77</u> | **77.05** | FAIL | 51.41 |
| Task Avg (Std) | 29 (9.7) | 61 (4.6) | 55 (2.6) | 41 (1.8) | 72 (3.7) | FAIL | 52 (2.4) |

Table 1: Experimental results on Long-Range Arena benchmark. Best model is in boldface and second best is underlined. All models do not learn anything on Path-X task, contrary to the Pathfinder task and this is denoted by FAIL. This shows that increasing the sequence length can cause seriously difficulties for model training. We leave Path-X on this benchmark for future challengers but do not include it on the Average score as it has no impact on relative performance.

# The NLP Approach

## Efficient implementation

Rasley et. Al. 2020. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters
Dao et. al. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness
Dettmers et. al. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale.

## Hierarchical encoding

Seo et. al. 2017. Bidirectional Attention Flow for Machine Comprehension
Wu. et. al. 2021. Recursively Summarizing Books with Human Feedback

## Non-transformer Architectures

LSTM?
Hutchins et. al. 2022. Block-Recurrent Transformers.

No absolute conclusion which one is the best

# The NLP Approach

| Model | ListOps | Text | Retrieval | Image | Pathfinder | Path-X | Avg |
|---|---|---|---|---|---|---|---|
| Transformer | 36.37 | 64.27 | 57.46 | 42.44 | 71.40 | FAIL | <u>54.39</u> |
| Local Attention | 15.82 | 52.98 | 53.39 | 41.46 | 66.63 | FAIL | 46.06 |
| Sparse Trans. | 17.07 | 63.58 | **59.59** | **44.24** | 71.71 | FAIL | 51.24 |
| Longformer | 35.63 | 62.85 | 56.89 | 42.22 | 69.71 | FAIL | 53.46 |
| Linformer | 35.70 | 53.94 | 52.27 | 38.56 | <u>76.34</u> | FAIL | 51.36 |
| Reformer | **37.27** | 56.10 | 53.40 | 38.07 | 68.50 | FAIL | 50.67 |
| Sinkhorn Trans. | 33.67 | 61.20 | 53.83 | 41.23 | 67.45 | FAIL | 51.39 |
| Synthesizer | <u>36.99</u> | 61.68 | 54.67 | 41.61 | 69.45 | FAIL | 52.88 |
| BigBird | 36.05 | 64.02 | <u>59.29</u> | 40.83 | 74.87 | FAIL | **55.01** |
| Linear Trans. | 16.13 | **65.90** | 53.09 | 42.34 | 75.30 | FAIL | 50.55 |
| Performer | 18.01 | <u>65.40</u> | 53.82 | <u>42.77</u> | **77.05** | FAIL | 51.41 |
| Task Avg (Std) | 29 (9.7) | 61 (4.6) | 55 (2.6) | 41 (1.8) | 72 (3.7) | FAIL | 52 (2.4) |

Table 1: Experimental results on Long-Range Arena benchmark. Best model is in boldface and second best is underlined. All models do not learn anything on Path-X task, contrary to the Pathfinder task and this is denoted by FAIL. This shows that increasing the sequence length can cause seriously difficulties for model training. We leave Path-X on this benchmark for future challengers but do not include it on the Average score as it has no impact on relative performance.

But one apparent baseline is the full transformer

Table 4: (**Long Range Arena**) (*Top*) Original Transformer variants in LRA. Full results in Appendix D.2. (*Bottom*) Other models reported in the literature. *Please read Appendix D.5 before citing this table.*

| Model | ListOps | Text | Retrieval | Image | Pathfinder | Path-X | Avg |
|---|---|---|---|---|---|---|---|
| Transformer | 36.37 | 64.27 | 57.46 | 42.44 | 71.40 | ✗ | 53.66 |
| Reformer | <u>37.27</u> | 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 | <u>65.90</u> | 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 | <u>77.80</u> | ✗ | 54.42 |
| Nyströmformer | 37.15 | 65.52 | <u>79.56</u> | 41.58 | 70.94 | ✗ | 57.46 |
| Luna-256 | 37.25 | 64.57 | 79.29 | <u>47.38</u> | 77.72 | ✗ | <u>59.37</u> |
| **S4** | **59.60** | **86.82** | **90.90** | **88.65** | **94.20** | **96.35** | **86.09** |

# State-space Models

Table 4: (**Long Range Arena**) (*Top*) Original Transformer variants in LRA. Full results in Appendix D.2. (*Bottom*) Other models reported in the literature. *Please read Appendix D.5 before citing this table.*

| 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** |

# State-space Models

## Early years

Zhang et. al. ICML 2018. Learning long term dependencies via Fourier recurrent Units
Voelker et. al. NeurIPS 2019. Legendre memory units: Continuous-time representation in recurrent neural networks

## State-space models

(HiPPO): Gu et. al. NeurIPS 2020. HiPPO: Recurrent Memory with Optimal Polynomial Projections
(LSSL): Gu et. al. NeurIPS 2021. Combining Recurrent, Convolutional, and Continuous-time Models with Linear State-Space Layers
(S4): Gu et. al. ICLR 2022. Efficiently Modeling Long Sequences with Structured State Spaces

## Diagonal simplification

(DSS): Gupta et. al. 2022. Diagonal State Spaces are as Effective as Structured State Spaces
(S4D): Gu. et. al. 2022. On the Parameterization and Initialization of Diagonal State Space Models

## Further Development

(GSS): Mehta et. al. 2022. Long Range Language Modeling via Gated State Spaces
(SaShiMi): Goel. et. al. 2022. It's Raw! Audio Generation with State-Space Models
(S5): Smith et. al. 2022. Simplified State Space Layers for Sequence Modeling
Gu et. al. 2022. How to Train Your HiPPO: State Spaces with Generalized Orthogonal Basis Projections

## NLP Relatives

Martins et. al. ACL 2022. ∞-former: Infinite Memory Transformer
Lee-Thorp et. al. NAACL 2022. FNet: Mixing Tokens with Fourier Transforms

# We highlight the following four papers

## Early years

Zhang et. al. ICML 2018. Learning long term dependencies via Fourier recurrent Units
Voelker et. al. NeurIPS 2019. Legendre memory units: Continuous-time representation in recurrent neural networks

## State-space models

(HiPPO): Gu et. al. NeurIPS 2020. HiPPO: Recurrent Memory with Optimal Polynomial Projections
(LSSL): Gu et. al. NeurIPS 2021. Combining Recurrent, Convolutional, and Continuous-time Models with Linear State-Space Layers
(S4): Gu et. al. ICLR 2022. Efficiently Modeling Long Sequences with Structured State Spaces

## Diagonal simplification

(DSS): Gupta et. al. 2022. Diagonal State Spaces are as Effective as Structured State Spaces
(S4D): Gu. et. al. 2022. On the Parameterization and Initialization of Diagonal State Space Models

## Further Development

(GSS): Mehta et. al. 2022. Long Range Language Modeling via Gated State Spaces
(SaShiMi): Goel. et. al. 2022. It's Raw! Audio Generation with State-Space Models
(S5): Smith et. al. 2022. Simplified State Space Layers for Sequence Modeling
Gu et. al. 2022. How to Train Your HiPPO: State Spaces with Generalized Orthogonal Basis Projections

# Guidelines

Long Sequence Encoding

The NLP Approach

State-space Models

Mathematical Principles of State-space Models

Challenges

# Why S4 is Good at Long Sequence: Remembering a Sequence with Online Function Approximation

Yao Fu, University of Edinburgh. https://franxyao.github.io/

yao.fu@ed.ac.uk

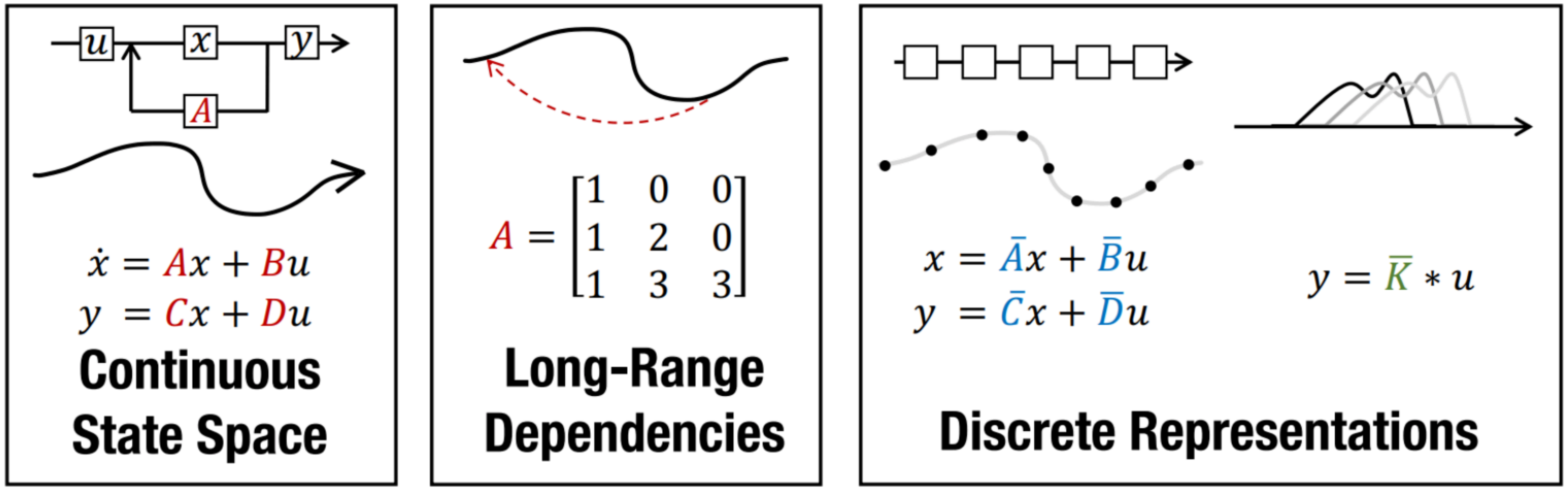
The Structured State Space for Sequence Modeling (S4) model achieves impressive results on the Long-range Arena benchmark with a substantial margin over previous methods. However, it is written in the language of control theory, ordinary differential equation, function approximation, and matrix decomposition, which is hard for a large portion of researchers and engineers from a computer science background. This post aims to explain the math in an intuitive way, providing an approximate feeling/ intuition/ understanding of the S4 model: *Efficiently Modeling Long Sequences with Structured State Spaces*. ICLR 2022

https://yaofu.notion.site/Why-S4-is-Good-at-Long-Sequence-Remembering-a-Sequence-with-Online-Function-Approximation-836fc54a49aa413b84997a265132f13f

# The Annotated S4

Efficiently Modeling Long Sequences with Structured State Spaces

Albert Gu, Karan Goel, and Christopher Ré.




*Blog Post and Library by Sasha Rush and Sidd Karamcheti*, v3

The Structured State Space for Sequence Modeling (S4) architecture is a new approach to very long-range sequence modeling tasks for vision, language, and audio, showing a capacity to capture dependencies over tens of thousands of steps. Especially impressive are the model's results on the challenging Long Range Arena benchmark, showing an ability to reason over sequences of up to **16,000+** elements with high accuracy.

https://srush.github.io/annotated-s4/

# Guidelines

Long Sequence Encoding

The NLP Approach

State-space Models

Mathematical Principles of State-space Models

Challenges

# Challenges

Table 4: (**Long Range Arena**) (*Top*) Original Transformer variants in LRA. Full results in Appendix D.2. (*Bottom*) Other models reported in the literature. *Please read Appendix D.5 before citing this table.*

| 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** |

Text: Byte-level text classification, imdb binary classification

Retrieval: Byte-level document retrieval, ACL Anthology Network if a paper cites another, binary classification

Pathfinder and PathX: whether two points are connect by a path. Binary classification

# Challenges

Table 8: (**WikiText-103 language modeling**) S4 approaches the performance of Transformers with much faster generation. (*Top*) Transformer baseline which our implementation is based on, with attention replaced by S4. (*Bottom*) Attention-free models (RNNs and CNNs).

| Model | Params | Test ppl. | Tokens / sec |
|---|---|---|---|
| Transformer | 247M | **20.51** | 0.8K (1×) |
| GLU CNN | 229M | 37.2 | - |
| AWD-QRNN | 151M | 33.0 | - |
| LSTM + Hebb. | - | 29.2 | - |
| TrellisNet | 180M | 29.19 | - |
| Dynamic Conv. | 255M | 25.0 | - |
| TaLK Conv. | 240M | 23.3 | - |
| **S4** | 249M | **20.95** | **48K (60×)** |

# Challenges

## SCROLLS: Standardized CompaRison Over Long Language Sequences

**What is SCROLLS?**

SCROLLS is a suite of datasets that require synthesizing information over long texts. The benchmark includes seven natural language tasks across multiple domains, including summarization, question answering, and natural language inference.

Read the paper (Shaham et al., 2022)

**Citing SCROLLS**

Please use the following bibliography to cite SCROLLS:

```
@misc{shaham2022scrolls,
    title={SCROLLS: Standardized CompaRison Over Long Language Sequences},
     author={Uri Shaham and Elad Segal and Maor Ivgi and Avia Efrat and Ori Yoran and Adi Haviv and
Ankit Gupta and Wenhan Xiong and Mor Geva and Jonathan Berant and Omer Levy},
    year={2022},
    eprint={2201.03533},
    archivePrefix={arXiv},
    primaryClass={cs.CL}
}
```

When citing SCROLLS, please make sure to cite all of the original dataset papers. [bibtex]

TAU NLP

**Contact Us**

scrolls-benchmark-contact@googlegroups.com

# Comparison Between Scrolls and LRA

| | Task nature | Input length | input vocabulary | output length | output vocab | Spasity |
|---|---|---|---|---|---|---|
| LRA (no recency bias) | seq. classification | 1K / 2K / 4K / 8K / 16000 | 2 / 32 / 256 | 1 | 2 | Sparse: most zeros in the seq., little non-zero elements |
| Scrolls (strong recency bias) | seq2seq | 16K | 52K | 1024 | 52K | Dense: every element in seq is different |

# Summary of Challenges

Fitness: is state-space models fit for NLP tasks?

- Strength: extracting highly sparse signals for long sequences
- NLP: Encoding dense information

Math background

- Real Analysis, Functional Analysis (3rd year math undergrad)
- Signal Processing, Fourier Analysis (2nd year EE undergrad)
- Parallel computation and complexity (3nd year CS undergrad)
- Matrix decomposition (which major teach this??)

Engineering

- Multiple packages interacting: state-space, transformers, torchtext, torch-lightning, fairseq, scrolls (many conflictions to each other)
- Custom CUDA kernel: require certain version of g++ and cuda and pytorch (again, many conflictions)

Infrastructure

- Nvidia A100

This is indeed a challenging problem

But in many times
people cannot solve a challenging problem
not because of lack of knowledge
but lack of courage

Are you brave enough to challenge the dragon?

Thank you