Paramixer: Parameterizing Mixing Links in Sparse Factors Works Better than Dot-Product Self-Attention (Supplemental Document)

Tong Yu * Ruslan Khalitov * Lei Cheng Zhirong Yang †
Norwegian University of Science and Technology

1. Synthetic Data Experiments

For both problems in the scalability test, we generated sequences using the setup described in the main paper. We ran the experiments on sequences with variable lengths: from 128 to 32k. The longer sequences, the more complex the retrieval process. There is a slight difference in the pre-processing part. For the Adding problem, the input data was only two-dimensional. To avoid using such a low-dimensional embedding space, we augmented the dimensionality with an additional linear layer to assure sufficient freedom for dot-product attention architectures. The training configuration and hyperparameters are the same for both the Adding problem and the Temporal Order problem. Their summary is in Table 1

2. Long Range Arena

The data set for the LRA benchmark is publicly available. The information about data and the download link can be found in the official GitHub repository: https://github.com/google-research/long-range-arena.

- **ListOps** The raw data for this problem is organized as three separate files `basic_train.tsv`, `basic_test.tsv`, `basic_val.tsv` for training, testing, and validation data, respectively. The split is fixed. In addition to the tokens described in the main paper, each sequence has "(" and ")" symbols, which should be removed. To equalize the lengths of the sequences, we used the built-in PyTorch padding functional. After the sequences are prepared, the embedding layer processes each unique value, thus mapping elements to the embedding space. The rest of the training process is straightforward.

- **Text Classification** We downloaded IMDB data set using the `tensorflow-dataset` package, and got 25000 instances for training and another 25000 for testing. We went through the whole corpus and extracted the character vocabulary. Then we mapped each sequence to a vector of indices using this vocabulary. Finally, we truncated or padded each sequence to a fixed length of 4096. For every review, we add ["CLS"] token to each sequence and use the embedding of ["CLS"] token for final classification. We used three blocks Paramixer for this task.

- **Image Classification** CIFAR10 is a well-known dataset, which can be downloaded from the `torchvision` package. The train/test splitting is fixed. To make images grayscaled, we used standard transformation `transforms|grayscale` from the same package. An image is flattened to a sequence of length 1024. Then each element is mapped to a dictionary of size 256 (all possible intensity values) and given to the embedding layer.

- **Pathfinder** The problem data consists of two types of files: images and metafiles. Metafiles store information about all the images and their corresponding labels (positive or negative). There are three classes of images: `curv_baseline` (easy), `curv_contour_length_9` (medium), `curv_contour_length_14` (hard). An image class corresponds to the distance between its endpoints (curve length), thus positively correlates with the difficulty level. The exact data split is not provided. To separate the data into three parts, we iterated over all metafiles from the catalogs and constructed the training/val/test (90%/5%/5%) sets such that all three types of images are present equally. The rest of the processing is similar to the Image Classification task.

3. Long Document Classification

The task is a four-class classification problem. The class of a paper is defined by its arxiv categorization, namely, cs.AI, cs.NE, math.AC, and math.GR. Each class in the data set is almost equally presented, with a slight class imbalance: 2995, 3012, 2885, and 3065 documents, respectively.

*Equal contribution.
†Corresponding author. zhirong.yang@ntnu.no

Table 1. Hyperparameters details for every task. N, B, V, E, H, lr refer to max sequence length, batch size, vocabulary size, embedding size, hidden states size, and learning rate, respectively. The vocabulary size includes padding index and ["CLS"].

Task	N	Protocol	n_links	lr	B	V	E	H	pos_embed	Pooling Type
Adding	32768	CHORD	15	0.001	40	-	32	32	True	FLAT
Temporal Order	16384	CHORD	14	0.001	40	6	32	32	True	FLAT
ListOps	2000	CHORD	12	0.001	48	16	32	32	True	FLAT
CIFAR10	1024	CDIL	3	0.001	64	256	32	32	True	FLAT
Text	4096	CDIL	9	0.0001	32	97	32	128	False	CLS
Pathfinder	1024	CHORD	11	0.001	64	256	32	32	True	FLAT
Long Document	16384	CHORD	15	0.0001	16	4290	100	128	False	CLS
Long Document	32768	CHORD	16	0.0001	16	4290	100	128	False	CLS
Genome Classification	16384	CHORD	15	0.0001	16	5	32	128	True	FLAT

To transform the raw articles into sequences, we first went through the whole corpus and extracted the character vocabulary. Then we mapped each character sequence to a vector of indices using this vocabulary. We fine-tuned Paramixer and X-formers to get the best results. The hyperparameters of Paramixer were selected using a similar process. Final configurations are shown in Table 1. For every document, we add ["CLS"] token to each sequence and use the result embedding of ["CLS"] token for the final classification.

4. Genome Classification

When building MTcDNA we downloaded cDNA sequences of Chelonoidis abingdonii and Gopherus agassizii, and merged them as a turtle data set. Following the same strategy, we built the mouse data set using Mus musculus and Mus spretus. More details can be found in the main paper. For the HFDNA classification task, one Paramixer block is enough to get 100% accuracy. However, ParamixerNn with two blocks result in the best test accuracy for MTcDNA. The selected hyperparameters are listed in Table 1.

5. Proof of Proposition 3.1 in the main paper

Definition 1. *An $N \times N$ circulant matrix C takes the form*

$$C = \begin{bmatrix} c_0 & c_{N-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{N-1} & \cdots & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{N-2} & \cdots & \ddots & \ddots & c_{N-1} \\ c_{N-1} & c_{N-2} & \ddots & c_1 & c_0 \end{bmatrix}$$

Definition 2. *The polynomial*

$$f(x) = c_0 + c_1 x + \cdots + c_{N-1} x^{N-1}$$

is called the associated polynomial of circulant matrix C.

We have the following theorem in the literature [1]:

Theorem 1. *The rank of a circulant matrix C is equal to $N - d$, where d is the degree of the polynomial GCD$(f(x), x^{N-1})$.*

Now we can prove Preposition 3.1 for the CHORD protocol. The proof for CDIL follows similarly.

Proof. The associated polynomial of $W^{(m)}$ is

$$f(x) = \sum_{k=0}^{\log_2 N - 1} x^k$$

Because GCD$(f(x), x^N - 1) = 1 = x^0$, the rank of $W^{(m)}$ is N. □

References

[1] A. W. Ingleton. The rank of circulant matrices. *Journal of the London Mathematical Society*, s1-31(4):445–460, 1956. 2