Bidirectional Long-Range Parser for Sequential Data Understanding

George Leotescu LEOTEG@AMAZON.COM
Amazon Inc.

Daniel Voinea DVOINEA@AMAZON.COM
Amazon Inc.

Alin-Ionut Popa POPAALN@AMAZON.COM
Amazon Inc.

Abstract

The transformer is a powerful data modelling framework responsible for remarkable performance on a wide range of tasks. However, they are limited in terms of scalability as it is suboptimal and inefficient to process long-sequence data. To this purpose we introduce **BLRP** (**B**idirectional **L**ong-**R**ange **P**arser), a novel and versatile attention mechanism designed to increase performance and efficiency on long-sequence tasks. It leverages short and long range heuristics in the form of a local sliding window approach combined with a global bidirectional latent space synthesis technique. We show the benefits and versatility of our approach on vision and language domains by *demonstrating competitive results against state-of-the-art* methods on the Long-Range-Arena and CIFAR benchmarks together with ablations demonstrating the computational efficiency.

Keywords: long sequence understanding; bidirectional attention mechanism

1 Introduction

Statistical modeling of sequential data has witnessed increased scientific attention since the appearance of probabilistic learning approaches. Initially, notable approaches such as Norris (1998); Hochreiter and Schmidhuber (1997); Schuster and Paliwal (1997); Cho et al. (2014); Arjovsky et al. (2016) applied recurrent processing for this task. Usually they extract discriminative features and capture dominant patterns from the analysed data. Although this procedure has its merits, it is limited in terms of generability. For example, the recurrence aspect encourages sensitivity with respect to the ordering of the elements. With the emergence of transformer methodology Vaswani et al. (2017) and the introduction of the multi-head attention, the performance boundaries for sequential data understanding tasks were pushed further. It is a powerful mechanism which parses in parallel the sequence elements and performs an implicit embedding statistic, thus emphasizing global relationships among the elements.

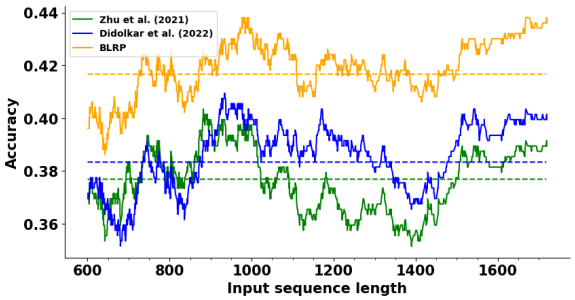

Figure 1: **Performance comparison for different sequence lengths.** We compare our proposed **BLRP** framework against Didolkar et al. (2022) and Zhu et al. (2021) for different sequence lengths on `ListOps`. **BLRP** brings scalable performance gains irrespective to sequence length, showing that the bi-directional mechanism increases the model's representative power on all input length ranges. Dotted lines represent average performance for each method.

©2024 Leotescu et al.

However, this benefit comes at the expense of scalability. The attention mechanism, in its primal form, scales quadratically with respect to the sequence size and as a direct consequence it is expensive and suboptimal to parse data structured as long sequences.

To address this limitation, we propose **B**idirectional **L**ong-**R**ange **P**arser, a novel attention framework designed to efficiently parse long sequences (*i.e.* $> 2,000$) which integrates *(I)* a local-window attention capturing *small scale* correlations between the elements of the sequence and *(II)* a bidirectional aggregation technique which captures recurrently the *large scale* context of the full-sequence into a temporal latent block representation. The intuition behind our approach is to efficiently capture proximal and distant relationships between elements at spatial level (*i.e.* sequence positioning), while taking into account their ordering interpreted as the temporal axis (*i.e.* going forward and backward along the sequence). Thus, we build a data-centric solution to efficiently address the long sequence scalability issue. From the linguistic point of view, our proposed approach is able to recover local discriminative vocabulary items, while successfully grasping the ample context where they are utilized and observe from a bidirectional perspective how they contribute to the textual flow. We experiment on the challenging benchmarks Long-Range-Arena (LRA) Tay et al. (2021a) and CIFAR Krizhevsky et al. (2009) proving competitive results against state-of-the-art approaches on multiple domains. An overview of our proposed mechanism is illustrated in Figure 2. Our approach is generic, in the sense that it can be applied on top of any sequential data parsing approach from multiple domains (*i.e.* language or vision).

2 Related Work

Transformers Vaswani et al. (2017) made a significant impact across multiple machine learning fields, such as computer vision Carion et al. (2020); Radford et al. (2021), signal processing Che et al. (2021); Hu et al. (2022); Huang et al. (2023) or NLP Xu et al. (2020b,a); Tay et al. (2021b); Li et al. (2021). They are able to dynamically capture efficient dependencies across the sequence while analysing the entire content in parallel by leveraging the attention mechanism. While recent research efforts focused on multi-modal attention Li et al. (2021); Appalaraju et al. (2021); Dosovitskiy et al. (2020) and more efficient ways to extract information, the major limitation imposed by quadratic formulation of the attention matrix has been relatively unexplored. A number of approaches Didolkar et al. (2022); Hutchins et al. (2022); Zhu et al. (2021); Dai et al. (2019); Rae et al. (2019); Wu et al. (2020); Zhou et al. (2021); Zhang et al. (2021); Zaheer et al. (2020); Beltagy et al. (2020) have been proposed to address this problem by focusing on approximating the attention matrix. Our proposed **BLRP** parses the sequence in a bidirectional fashion by retaining a perceptual latent representation. Moreover, encouraged by the ideas of Zhu et al. (2021) we initialize the latent block representation using a projection of the entire input sequence. Another important aspect of our approach is that we are in line with the model requirements and hyperparameter configuration from the LRA benchmark[1] to have a fair comparison with current state-of-the-art.

Our approach is more similar to the work of Zhu et al. (2021). They consider analysing the sequence in two different ways: *locally*, using a sliding window attention heuristic and *globally*, by dynamically projecting the entire input sequence. The work of Didolkar et al. (2022) is also related to us as they consider splitting the input sequence into chunks and unidirectionaly parsing them into a canonical representation.

1. https://github.com/google-research/long-range-arena

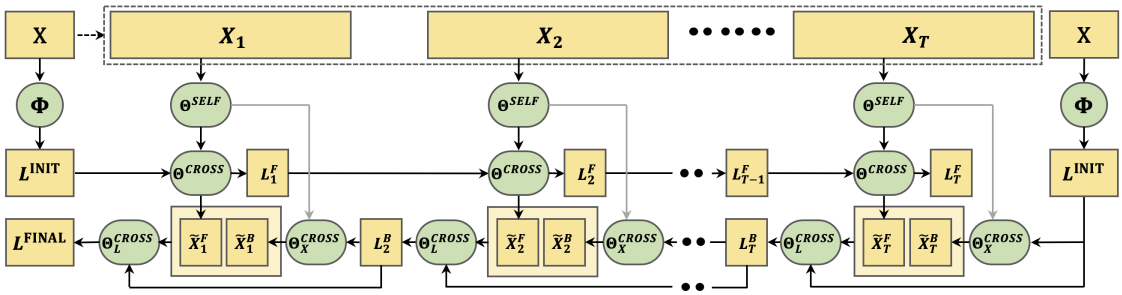

Figure 2: **Detailed overview of the proposed BLRP method.** The flow is from left to right for the *forward* pass, followed by right to left for the *backward* pass. The input sequence $\mathbf{X}$ is split into a list of non-overlaping segments $(\mathbf{X}_i)_{i=1}^T$, which is bidirectionally parsed while capturing the overall information into the latent block $\mathbf{L}$, originally initialized with $\mathbf{L}^{\texttt{INIT}}$ via function Φ. In turn, $\mathbf{L}^{\texttt{INIT}}$ is added at the start of both forward and backward passes and subsequently used as a residual connection. Thus, we obtain the temporal level state representations, $(\mathbf{L}_i^{\texttt{F}})_{i=1}^T$ and $(\mathbf{L}_i^{\texttt{B}})_{i=T}^1$, which synthesise the aggregated information from the entire sequence inside $\mathbf{L}^{\texttt{FINAL}}$. Notice that we optimally aggregate information at spatial level by iteratively conditioning the latent states on the segment embeddings, and at temporal level by utilizing the corresponding forward segment embeddings to update the backward states.

We are aware that there are multiple methods Gu et al. (2022); Smith et al. (2022); Hasani et al. (2022); Orvieto et al. (2023) outperforming the LRA benchmark by a large margin while not applying the above mentioned constraints. However, our goal is to ensure a fair comparison normalized with respect to the total number of the parameters of the model and within the boundaries of the original hyperparameter configuration proposed by the authors of the LRA benchmark.

3 Methodology

In this section we will elaborate the computational details behind our proposed **BLRP** framework. Let there be an input sequence $\mathbf{X} = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^{N \times d}$ containing N elements with dimensionality d. This is split into a list of T segments $\mathbf{X}_i = (x_{(i-1)\cdot t+1}, x_{(i-1)\cdot t+2}, \ldots, x_{i\cdot t}) \in \mathbb{R}^{t \times d}$, each of equal size t. The last segment $\mathbf{X}_T$ is padded to achieve the desired size t. Our model leverages a latent block representation $\mathbf{L} \in \mathbb{R}^{l \times d}$ which is used to retrieve relevant global information. This is achieved via a bidirectional flow over the list of segments $(\mathbf{X}_i)_{i=1}^T$. In essence, $\mathbf{L}$ transitions to a different state as it moves along the bidirectional loop. As a result of the forward and backward pass, we have the following state representations of $\mathbf{L}$: $(\mathbf{L}_i^{\texttt{F}})_{i=1}^T$ going forward and $(\mathbf{L}_i^{\texttt{B}})_{i=T}^1$ going backward. We retain the final state representation of $\mathbf{L}$ denoted with $\mathbf{L}^{\texttt{FINAL}}$ which synthesizes the information from the entire sequence $\mathbf{X}$. For each segment $\mathbf{X}_i$ we include the 1D positional embeddings following Didolkar et al. (2022).

In the following, we will detail the algorithmic steps of our proposed **BLRP** pipeline. Firstly, we initialize the latent state using a projection of the raw sequence into the latent space using the function $\Phi : \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{t \times d}$. Inspired from Zhu et al. (2021) we apply a dynamic projection of the input sequence into the latent space via a learned basis representation. This process is repeated independently for both forward and backward passes. Thus, we have the initial latent state denoted as $\mathbf{L}^{\texttt{INIT}} = \Phi(\mathbf{X})$ which is applied at the initial step of each pass, forward and backward. The attention operations play a crucial role in our framework. The self-attention operation denoted as $\Theta^{\texttt{SELF}}$ has

the following formula, $\Theta^{\texttt{SELF}}(\mathbf{X}) = softmax(\frac{q^{\texttt{S}}(\mathbf{X})k^{\texttt{S}}(\mathbf{X})^\top}{\sqrt{d}})v^{\texttt{S}}(\mathbf{X})$, where functions $k^{\texttt{S}}(\cdot)$, $q^{\texttt{S}}(\cdot)$ and $v^{\texttt{S}}(\cdot)$ represent the keys, queries and values, respectively, together with their embedded projection weights. $\Theta^{\texttt{SELF}}$ is used to identify local correlations between the elements within each segment.

For the forward pass, we process iteratively each input segment $\mathbf{X}_i$ to obtain the latent block states $(\mathbf{L}_i^{\texttt{F}})_{i=1}^T$ and the forward processed input segment embeddings $(\tilde{\mathbf{X}}_i^{\texttt{F}})_{i=1}^T$

$$\tilde{\mathbf{X}}_i^{\texttt{F}} = \begin{cases} \Theta_X^{\texttt{CROSS}}(\Theta^{\texttt{SELF}}(\mathbf{X}_i), \mathbf{L}^{\texttt{INIT}}), & i = 1 \\ \Theta_X^{\texttt{CROSS}}(\Theta^{\texttt{SELF}}(\mathbf{X}_i), \mathbf{L}_{i-1}^{\texttt{F}}), & i > 1 \end{cases}$$

$$\mathbf{L}_i^{\texttt{F}} = \begin{cases} \Theta_L^{\texttt{CROSS}}(\mathbf{L}^{\texttt{INIT}}, \tilde{\mathbf{X}}_i^{\texttt{F}}), & i = 1 \\ \Theta_L^{\texttt{CROSS}}(\mathbf{L}_{i-1}^{\texttt{F}}, [\tilde{\mathbf{X}}_i^{\texttt{F}}; \mathbf{L}^{\texttt{INIT}}]), & i > 1 \end{cases}$$

where,

$$\Theta_X^{\texttt{CROSS}}(\mathbf{X}, \mathbf{L}) = softmax(\frac{q_X^{\texttt{C}}(\mathbf{X})k_X^{\texttt{C}}(\mathbf{L})^\top}{\sqrt{d}})v_X^{\texttt{C}}(\mathbf{L})$$

$$\Theta_L^{\texttt{CROSS}}(\mathbf{X}, \mathbf{L}) = softmax(\frac{q_L^{\texttt{C}}(\mathbf{L})k_L^{\texttt{C}}(\mathbf{X})^\top}{\sqrt{d}})v_L^{\texttt{C}}(\mathbf{X})$$

Functions $k_L^{\texttt{C}}(\cdot)$, $q_L^{\texttt{C}}(\cdot)$, $v_L^{\texttt{C}}(\cdot)$, $k_X^{\texttt{C}}(\cdot)$, $q_X^{\texttt{C}}(\cdot)$ and $v_X^{\texttt{C}}(\cdot)$ are equivalent to $k^{\texttt{S}}(\cdot)$, $q^{\texttt{S}}(\cdot)$ and $v^{\texttt{S}}(\cdot)$. This updating step is illustrated in Figure 3. In essence, at each time step i we update the segment embeddings, $\tilde{\mathbf{X}}_i^{\texttt{F}}$, via cross-attention against the current latent block state. Next, we repeat the update process in the same manner to generate the backward states, $(\mathbf{L}_i^{\texttt{B}})_{i=T}^1$, for the latent block. When updating the backward latent states, we let the cross-attention keys attend to the union between the encoded forward segment (*i.e.* $\tilde{\mathbf{X}}_i^{\texttt{F}}$), the encoded backward segment (*i.e.* $\tilde{\mathbf{X}}_i^{\texttt{B}}$) and $\mathbf{L}^{\texttt{INIT}}$, with

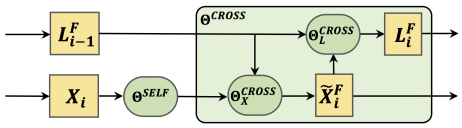

Figure 3: **Module $\Theta^{\texttt{CROSS}}$ at step *i*.** Forward embeddings $\tilde{\mathbf{X}}_i^{\texttt{F}}$ and latent state $\mathbf{L}_i^{\texttt{F}}$ are obtained by an interleave usage of $\Theta_L^{\texttt{CROSS}}$ and $\Theta_X^{\texttt{CROSS}}$.

$$\tilde{\mathbf{X}}_i^{\texttt{B}} = \begin{cases} \Theta_X^{\texttt{CROSS}}(\Theta^{\texttt{SELF}}(\mathbf{X}_i), \mathbf{L}^{\texttt{INIT}}), & i = T \\ \Theta_X^{\texttt{CROSS}}(\Theta^{\texttt{SELF}}(\mathbf{X}_i), \mathbf{L}_{i+1}^{\texttt{B}}), & i < T \end{cases}$$

$$\mathbf{L}_i^{\texttt{B}} = \begin{cases} \Theta_L^{\texttt{CROSS}}(\mathbf{L}_T^{\texttt{F}}, [\tilde{\mathbf{X}}_i^{\texttt{F}}; \tilde{\mathbf{X}}_i^{\texttt{B}}]), & i = T \\ \Theta_L^{\texttt{CROSS}}(\mathbf{L}_{i+1}^{\texttt{B}}, [\tilde{\mathbf{X}}_i^{\texttt{F}}; \tilde{\mathbf{X}}_i^{\texttt{B}}; \mathbf{L}^{\texttt{INIT}}]), & i < T \end{cases}$$

Intuitively, this acts as an intertwined spatio-temporal computational grid operating at micro and macro level designed to efficiently extract relevant information encoded within a long sequence. Finally, the last state of the latent block (*i.e.* $\mathbf{L}_1^{\texttt{B}}$), denoted with $\mathbf{L}^{\texttt{FINAL}}$, retrieved after successfully passing the backward pass is incorporated as a bidirectional embedding for the entire sequence $\mathbf{X}$ for other downstream tasks. In our experiments, we couple it as input to an MLP head for a classification task. However, it can be incorporated into any other downstream bottom-up task which involves sequential data understanding (*e.g.* Question Answering, Named-Entity Recognition).

Model	ListOps	Text	Retrieval
Wang et al. (2020)	37.38	56.12	79.37
Xiong et al. (2021)	37.34	65.75	81.29
Zhu et al. (2021)	37.5	66.0	81.79
Didolkar et al. (2022)	38.2	82.08	76.91
BLRP	**41.43**	**82.83**	**83.43**

Model	CIFAR10		CIFAR100	
	64×64	128×128	64×64	128×128
Dosovitskiy et al. (2020)	93.75	73.18	69.53	47.4
Zhu et al. (2021)	93.58	83.27	74.47	57.11
Liu et al. (2022)	**97.66**	84.9	79.95	58.59
Didolkar et al. (2022)	94.79	84.38	79.17	59.19
BLRP	95.04	**86.85**	**83.85**	**61.06**

Table 1: ***(Left)*** **LRA Results.** Our proposed approach surpasses state-of-the-art methods on all 3 tasks from LRA benchmark. ***(Right)*** **CIFAR Results.** The models were trained on the 64×64 setting and transferred to 128×128. Results are averaged across 5 random seeds. **BLRP** outperforms the comparing baselines for all setups.

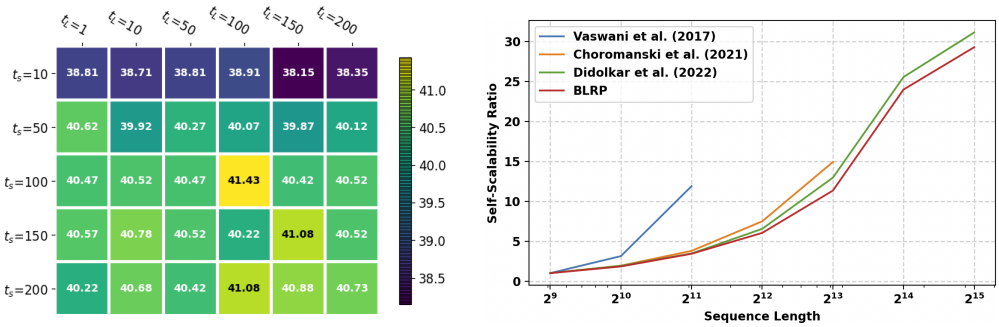

Figure 4: ***(Left)*** **Impact of latent block size versus segment size.** The rows correspond to segment sizes and are denoted with t_S and the columns correspond to latent block sizes and are denoted with t_L. The highest performance (*i.e.* 41.43) is obtained with segment size and latent block sizes equal to the value of 100. The poorest performance is obtained with a segment size of 1 as the window context is very limited, thus the model not being able to infer the global information within the input sequence. Increasing the latent size is not enough to reach optimal performance. The local context has to be large enough to capture meaningful correlations. ***(Right)*** **Scalability Analysis in Terms of GPU Memory Usage.** We tested the self-scalability in terms of GPU memory consumption for different sequence lengths (*i.e.* 512, 1024, etc). For Transformer Vaswani et al. (2017) and Performer Choromanski et al. (2021) we were unable to test on extremely long sequences due to GPU memory limitations. All measurements are realised on an NVidia A10G machine with 24 GB of memory.

4 Experiments

We demonstrate the effectiveness of **BLRP** on multiple modalities - textual modality by using LRA Tay et al. (2021a) and visual modality by using the CIFAR Krizhevsky et al. (2009) benchmark. For all of our experiments we used the PyTorch Paszke et al. (2019) library on a system with an Intel Xeon QuadCore 2.5 GHz with 32 GB system RAM and a single NVidia Tesla A10G GPU card with 24 GB of GPU RAM. For training our proposed ensemble, we use AdamW optimiser Loshchilov and Hutter (2019) with a starting learning rate of $4e^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e^{-9}$, $\gamma = 0.8$ with linear learning rate decay. A detailed view on hyperparameters and architectural design choices for each task is available in Table 2.

Benchmark	Embedding Size	Hidden Size	# of Heads	Segment Size	Batch Size
ListOps	64	128	8	100	32
Text	256	256	4	10	24
Retrieval	256	736	4	100	24
CIFAR10	368	736	6	10	128
CIFAR100	368	736	6	10	64

Table 2: **Hyperparameter details for BLRP on all benchmark.** This helps with reproducibility of our experimental results and provides the reader with meaningful insights on the internal settings of our best models.

LRA consists of three major subtasks, **ListOps**, **Text** and **Retrieval** addressing classification objectives which encompass similarity, long-range dependencies and structural representation. It involves sequential data with input sequences ranging from 500 up to $4,000$ elements. We use the recommended evaluation protocol described in Tay et al. (2021a). Moreover, our model has a total of $258K$ parameters which is in line with the requirements from Tay et al. (2021a) for a fair comparison. In Table 1 *(Left)* we showcase comparison results in terms of accuracy for all LRA tasks. On all 3 tasks we achieve superior performance. The reported results are averaged over 4 random runs. Similar to Didolkar et al. (2022), for all experiments we use two self-attention layers, one cross-attention layer to update the segment embedding and one cross-attention layer to update the latent block states. For CIFAR we evaluate on the image classification task using **CIFAR10** and **CIFAR100** datasets. The models are trained using ViT Dosovitskiy et al. (2020) backbone on a resolution of 64×64. The input image is split into patches of size 4×4 and fed in raster order to the model. Performance results are available in Table 1 *(Right)*. The model is evaluated on the default 64×64 as well as the 128×128 configurations to stress the generalization capabilities for long sequences. **BLRP** outperforms both comparing baselines demonstrating the effectiveness of our proposed approach. Furthermore, with this experiment we demonstrate the versatility aspect of our attention-mechanism as it can be adapted to other backbone type using a different learning modality.

4.1 Ablation Studies

To better understand the limitations and the intuition behind our approach, we performed extensive ablation studies. In Figure 1 we plot the average performance of our model against Didolkar et al. (2022) for different sequence sizes for ListOps benchmark. It is noticeable that we achieve consistent superior performance for all ranges of sequence lengths (standard sequence lengths - leftmost part of the figure and extemely long sequences - rightmost part of the figure). In Table 4.1 we study the impact of the bidirectional sequence parsing heuristic as well as different alternatives for the usage of the latent block $\mathbf{L}$. We trained different variants of **BLRP** on ListOps benchmark by going on a unidirectional / bidirectional flow, and with various initializations of the latent block $\mathbf{L}$. We use 3 different alternatives for $\mathbf{L}^{\texttt{INIT}}$: *(a)* 1D positional embedding of segment elements denoted with $[\texttt{1DPosEmb}]$, *(b)* $\Theta(\mathbf{X})^{\texttt{INIT}}$ corresponding to the learned dynamic projection of input sequence $\mathbf{X}$ which is added only at initialization of each pass (*i.e.* forward or backward) and *(c)* $\Theta(\mathbf{X})$ corresponding to the learned dynamic projection added throughout the sequence processing as a residual connection and within the cross-attention operation. This process is applied in a unidirectional fashion, going forward, rows $2-4$, going backward, rows $5-7$ and bidirectionally, rows $8-14$, where we use different initialisations for the latent block of each pass. The best

performance is achieved while using the dynamic projection function for both passes and adding the initialization as a skip connection throughout the process. This validates the importance of the bidirectional flow combined with information gain brought by using the dynamic projection.

In Figure 4 *(Left)* we analysed the impact of the latent block size against the segment size on the `ListOps` task. We trained and evaluated the performance of the model on all the combinations of latent block and segment sizes from the following set of values $[1, 10, 50, 100, 150, 200]$ and $[10, 50, 100, 150, 200]$, respectively. The rows correspond to segment sizes and are denoted with t_S and the columns correspond to latent block sizes and are denoted with t_L. We notice a performance drop correlated with the decrease of segment size. The highest performance (*i.e.* 41.43) is obtained with segment size and latent block sizes equal to the value of 100. The worst performance is obtained with a segment size of 10 as the window context is very limited, thus the model not being able to capture powerful local correlations that would enable learning discriminative global representations. In essence, a limitation of the context window leads to poorer understanding of the entire sequence. Moreover, we observe that the performance is higher when the latent size is equal to the segment size. Another important study is emphasized in Table 4. We use different alternatives for building $\mathbf{L}^{\texttt{INIT}}$ with Θ function. Firstly, we use the same variable generated by Θ function to instantiate the forward and backward pass. Secondly, we use the Θ function as a Siamese component to instantiate 2 separate variables for the bidirectional pass. Lastly, we learn 2 separate representations of the Θ function, one for the forward and one for the backward pass. The best outcome is obtained with different initialisations and different learned dynamic projection heads. This is a result of the fact that we strengthen the signal on both sides of the sequence pass using global information synthesised from the entire sequence.

Direction	Forward L Usage	Backward L Usage	Acc.
Forward	[1DPosEmb]	N/A	39.84
	$\Theta(\mathbf{X})^{\texttt{INIT}}$		39.91
	$\Theta(\mathbf{X})$		37.00
Backward	N/A	[1DPosEmb]	40.38
		$\Theta(\mathbf{X})^{\texttt{INIT}}$	40.47
		$\Theta(\mathbf{X})$	40.94
Bidirectional	[1DPosEmb]	[1DPosEmb]	39.44
	[1DPosEmb]	$\Theta(\mathbf{X})^{\texttt{INIT}}$	39.44
	$\Theta(\mathbf{X})^{\texttt{INIT}}$	[1DPosEmb]	39.55
	$\Theta(\mathbf{X})^{\texttt{INIT}}$	$\Theta(\mathbf{X})^{\texttt{INIT}}$	39.40
	[1DPosEmb]	$\Theta(\mathbf{X})$	40.86
	$\Theta(\mathbf{X})$	[1DPosEmb]	40.66
	$\Theta(\mathbf{X})$	$\Theta(\mathbf{X})$	**41.43**

Table 3: **Importance of bidirectional flow.** To validate the impact of the bidirectional flow, we trained different variants of **BLRP** on `ListOps` benchmark by going on a unidirectional / bidirectional flow, and with various initializations of the latent block $\mathbf{L}$. The best performance is achieved for a bidirectional flow, using dynamic projection initialization combined with skip connections.

To consolidate the long-range context understanding claim, we performed a sequence length augmentation study on `ListOps` task in table 5. We artificially multiplied the original sequences by self-concatenating them via the `MAX` operator. Thus, we ensure the output is invariant w.r.t. the sequence length. This process was achieved with a multiplication factor of between 1 and 4, thus forcing the model to process sequences of up to 8192 tokens. For testing, we use the models trained on the original `ListOps` containing sequences of up to 2048 elements. Our approach is able to achieve high performance, although it is tested out-of-domain sequences in terms of length. Comparatively, the model of Didolkar et al. (2022) manifests a high performance drop for all sequence lengths. This observation is in line with the rightmost section of the plot from Figure 1, showcasing the capability of our approach to maintain a long-context understanding despite the high length of the processed sequence.

Latent Variant	Accuracy
Identical $\mathbf{L}^{\texttt{INIT}}$ for backward / forward pass	40.86
Separate $\mathbf{L}^{\texttt{INIT}}$ obtained with Siamese $\Theta(\mathbf{X})$ for backward / forward pass	41.32
Separate $\mathbf{L}^{\texttt{INIT}}$ obtained with separate $\Theta(\mathbf{X})$ for backward / forward pass	**41.43**

Table 4: **Ablation on usage of dynamic projection function Θ.** We experimented using different alternatives for Θ function for the bidirectional setup. The best results are obtained when initializing the forward and backward pass latent blocks using separately learned dynamic projection functions. The second best is when we have different representations obtained with a Siamese setup of Θ.

4.2 Computational Efficiency and Limitations

Our proposed approach is incomplete without a study in terms of computational efficiency and scalability. To demonstrate the capabilities for our proposed **BLRP** with respect to this demand, we conducted an experiment in terms of self-scalability of the GPU memory consumption correlated with an increase in terms of sequence size. We measured how the model scales in terms of memory consumption, as the input sequences are increased considerably in length (*i.e.* exponential increase as a power of 2). Results are illustrated in Figure 4 *(Right)*. Basically, this study highlights how the model scales with respect to the length of the sequential data being processed. Please note that the vanilla transformer Vaswani et al. (2017) is limited to sequences of up to 4096 elements due to GPU memory limitations. It is worth noticing that our scalability is directly proportional against TLB Didolkar et al. (2022), achieving almost linear scalability, however at a higher performance gain (see Figure 1). In terms of inference speed measurements, we have averaged the run times normalized per sequence length and compared against Didolkar et al. (2022) baseline (0.39582 seconds for TLB Didolkar et al. (2022) compared with 0.41743 seconds for our method, both methods measured with a batch with 32 elements with $10,000$ elements) demonstrating that we achieve similar time efficiency for extremely long sequences at an overall higher performance gain (see Figure 1).

Model	Sequence Multiplication Factor			
	$\times 1$	$\times 2$	$\times 3$	$\times 4$
Didolkar et al. (2022)	38.2	16.98	15.70	14.86
BLRP	41.43	36.04	35.84	35.79

Table 5: **Performance analysis over long-range context.** We report the average performance on artificially augmented sequences from `ListOps`. The augmentation is performed by self-concatenating input sequences via the `MAX` operator, thus ensuring we obtain the same output. Our approach is able to maintain a high performance threshold, whereas Didolkar et al. (2022) gets a large performance gap due to the domain-shift distribution caused by sequence length.

As shown in our ablation studies, the proposed method is extremely sensitive w.r.t. the right combination of latent block size and segment size. Thus, making this type of methodology unsuitable for adapting pre-trained models on new data domains. For extremely long sequences we do not have an explicit gating mechanism, thus we cannot provide an insurance that forgetting irrelevant information will naturally occur via the cross-attention operations. This might explain why increasing the segment size is correlated with increasing the latent size. Moreover, we do not have a natural flow of combining multi-modal information flows.

5 Conclusion

With our proposed **BLRP** framework we demonstrate the importance of bidirectionally modelling close and distant dependencies across long sequences. We prove the optimality with respect to the long-sequence parsing aspect of **BLRP** on challenging benchmarks from multiple domains (visual and textual) using different learnable backbones. Furthermore, through our ablation studies, we emphasize the innovative aspects, the limitations and the intuition behind our work and the individual contribution of all architectural design choices.

References

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. *arXiv preprint arXiv:2106.11539*, 2021.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *ICML*, pages 1120–1128. PMLR, 2016.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020.

Chao Che, Peiliang Zhang, Min Zhu, Yue Qu, and Bo Jin. Constrained transformer network for ecg signal processing and arrhythmia classification. *BMC Medical Informatics and Decision Making*, 21(1):1–13, 2021.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Aniket Rajiv Didolkar, Kshitij Gupta, Anirudh Goyal, Nitesh Bharadwaj Gundavarapu, Alex Lamb, Nan Rosemary Ke, and Yoshua Bengio. Temporal latent bottleneck: Synthesis of fast and slow processing mechanisms in sequence learning. In *NeurIPS*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022.

Ramin Hasani, Mathias Lechner, Tsun-Hsuan Wang, Makram Chahine, Alexander Amini, and Daniela Rus. Liquid structural state-space models. *arXiv preprint arXiv:2209.12951*, 2022.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Rui Hu, Jie Chen, and Li Zhou. A transformer-based deep neural network for arrhythmia detection using continuous ecg signals. *Computers in Biology and Medicine*, 144:105325, 2022.

Feiqing Huang, Kexin Lu, CAI Yuxi, Zhen Qin, Yanwen Fang, Guangjian Tian, and Guodong Li. Encoding recurrence into transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. Block-recurrent transformers. In *NeurIPS*, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Mateusz Krubinski, Stefan Matcovici, Diana Grigore, Daniel Voinea, and Alin-Ionut Popa. Watermark text pattern spotting in document images. *arXiv preprint arXiv:2401.05167*, 2024.

Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5652–5660, 2021.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, pages 12009–12019, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Stefan Matcovici, Daniel Voinea, and Alin-Ionut Popa. k-nn embeded space conditioning for enhanced few-shot object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 401–410, 2023.

James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. *arXiv preprint arXiv:2303.06349*, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance

deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*. 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

Ionut-Catalin Sandu, Daniel Voinea, and Alin-Ionut Popa. Large sequence representation learning via multi-stage latent transformers. In *COLING*, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

Ioana-Sabina Stoian, Ionut-Catalin Sandu, Daniel Voinea, and Alin-Ionut Popa. Unstructured object matching using co-salient region segmentation. In *CVPR*, pages 5051–5060, 2022.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *ICLR*, 2021a.

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*, 2021b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. *arXiv preprint arXiv:2004.11886*, 2020.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14138–14148, 2021.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020a.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1192–1200, 2020b.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020.

Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *CVPR*, pages 2998–3008, 2021.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, volume 35, pages 11106–11115, 2021.

Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. Long-short transformer: Efficient transformers for language and vision. *NeurIPS*, 34:17723–17736, 2021.