Diffuser: Efficient Transformers with Multi-Hop Attention Diffusion for Long Sequences

Aosong Feng, Irene Li, Yuang Jiang, Rex Ying

Yale University, New Haven, CT, USA
aosong.feng@yale.edu, irene.li@yale.edu, yuang.jiang@yale.edu, rex.ying@yale.edu

Abstract

Efficient Transformers have been developed for long sequence modeling, due to their subquadratic memory and time complexity. Sparse Transformer is a popular approach to improving the efficiency of Transformers by restricting self-attention to locations specified by the predefined sparse patterns. However, leveraging sparsity may sacrifice expressiveness compared to full-attention, when important token correlations are multiple hops away. To combine advantages of both the efficiency of sparse transformer and the expressiveness of full-attention Transformer, we propose *Diffuser*, a new state-of-the-art efficient Transformer. Diffuser incorporates all token interactions within one attention layer while maintaining low computation and memory costs. The key idea is to expand the receptive field of sparse attention using *Attention Diffusion*, which computes multi-hop token correlations based on all paths between corresponding disconnected tokens, besides attention among neighboring tokens. Theoretically, we show the expressiveness of Diffuser as a universal sequence approximator for sequence-to-sequence modeling, and investigate its ability to approximate full-attention by analyzing the graph expander property from the spectral perspective. Experimentally, we investigate the effectiveness of Diffuser with extensive evaluations, including language modeling, image modeling, and Long Range Arena (LRA). Evaluation results show that Diffuser achieves improvements by an average of 0.94% on text classification tasks and 2.30% on LRA, with 1.67× memory savings compared to state-of-the-art benchmarks, which demonstrates superior performance of Diffuser in both expressiveness and efficiency aspects.

Introduction

Transformers (Vaswani et al. 2017) designed for sequential data have revolutionized the field of Natural Language Processing (NLP) (Liu et al. 2019; Zhu et al. 2020; Li et al. 2022), and have recently made tremendous impact in graph learning (Yang et al. 2021; Dwivedi and Bresson 2020) and computer vision (Dosovitskiy et al. 2020; Huynh 2022). The self-attention used by regular Transformer models comes with a quadratic time and memory complexity $\mathcal{O}(n^2)$ for input sequence of length n, which prevents the application of Transformers to longer sequences in practical settings with limited computational resources.

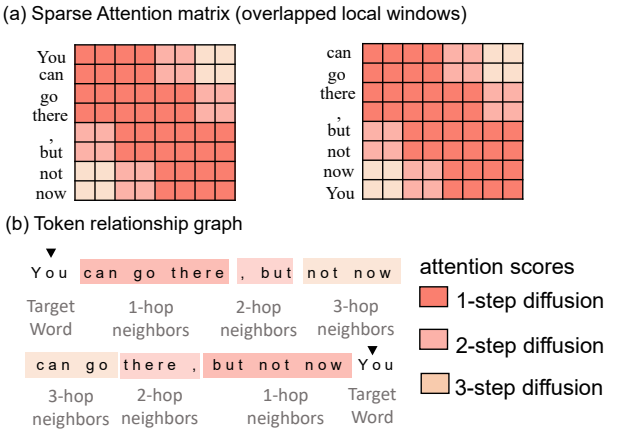

Figure 1: (a) Input token correlations follow predefined sparse pattern (b) The neighborhood structure for the target word completely change by rolling input tokens by 1.

Recently, many efficient Transformers that improve computational efficiency have emerged. One line of works approximates the $n \times n$ matrix multiplications by imposing a low-rank assumption on the attention structure, while the other line of works focuses on sparsification of the attention matrix. However, the improved computation efficiency always sacrifices expressiveness due to the following challenges:

Approximations of full-attention. The first line of works avoid explicitly computing $n \times n$ matrix through various approximations such as using the dot-product through kernalization (Wang et al. 2020; Katharopoulos et al. 2020) or random projections (Peng et al. 2021; Choromanski et al. 2020). However, such approximations are usually based on strict assumptions about the underlying attention structures such as the low-rank approximation (Shen et al. 2021; Tay et al. 2021a). There is currently a lack of rigorous guarantees for these assumptions to hold for potentially full-rank and dense self-attention matrices. Therefore these methods lead to empirically inferior results in sequence modeling tasks (Tay et al. 2021b; Ren et al. 2021), compared to the sparse Transformer approach.

Slow information propagation. The current state-of-the-art uses the sparse Transformer approach to approximate

the full self-attention (see a local pattern example in Figure 1)(Zaheer et al. 2020). However, such sparsity-based approach can be lossy or even misleading in capturing important token correlations when they are not directly connected. For example, as the sentence shown in Figure 1, one-hop attention scores of every pair of neighboring tokens can be misleading (caused by the word "can") when the real important correlations (the word "but", and "not") are two or three hops away. A sparse-attention layer only focuses on neighboring tokens, resulting in slower information propagation in the attention graph. Consequently, to model these crucial long-range correlations, sparse Transformers require more layers to expand the receptive field compared to full-attentions (Child et al. 2019; Ho et al. 2019; Dai et al. 2019). Some existing works (Ainslie et al. 2020; Beltagy, Peters, and Cohan 2020) deal with the slow propagation by introducing global attentions for important tokens, which alleviates the problem of long-range interactions. However, we cannot soly rely on global tokens for such propagation because of information loss when aggregating all tokens.

Robustness to input perturbations. Since attention graph is built upon predefined topology pattern, the attention process can become very different even with minor input sequence changes. As shown in Figure 1, being shifted by one token, the attention structure and neighboring tokens of the target word will completely change, leading to inconsistent outputs. Compared to full-attention where every token is attended regardless of its position, sparse Transformers are less robust to such input perturbations. Slow information propagation of sparse Transformers will amplify such attention inconsistency, as the inconsistency accumulates when the attention receptive fields gradually expand.

Proposed work. To address the mentioned expressiveness issues and further improve Transformer efficiency, we propose *Diffuser*, a novel sparse Transformer that achieves state-of-the-art performance on sequence modeling with $1.67\times$ memory savings compared to state-of-the-art efficient Transformers. The key insight is to introduce *Attention Diffusion* mechanism based on the designed sparse pattern for enabling efficient full-attention and larger receptive field. Diffuser first calculates attention scores on edges of the attention graph as in most sparse Transformers, then computes attention scores between other node pairs through the attention diffusion process. Unlike all existing sparse Transformers, Diffuser can model correlations among all pairs of tokens in a single Transformer layer, which extends the attention receptive field to the entire sequence, with minimal runtime overhead. We theoretically show that Diffuser can be more efficient (requires fewer layers) universal approximators for sequence modeling compared to all existing sparse Transformers, and has good properties to approximate the full-attention.

We further demonstrate the performance of Diffuser with datasets from various domains. Experiments demonstrate Diffuser's superior performance in expressiveness and efficiency. Compared with state-of-the-art efficient Transformers, Diffuser improves state-of-the-art by an average of 0.94% on text classification and 2.30% on LRA for long sequence modeling, with $1.67\times$ memory savings and comparable running time. Furthermore, Diffuser achieves state-of-the-art on 2 questions answering tasks and 2 image density estimation tasks.

Related Work

Efficient Transformers. Many works aim to optimize Transformers for longer inputs. Notably, Bigbird (Zaheer et al. 2020) introduced a sparse attention method that considers random, windowed, and global attention, improving performance on tasks including question answering and summarization. Similarly, Longformer (Beltagy, Peters, and Cohan 2020) presented a combination of windowed self-attention and global attention to sparsify the dense attention. Sparse sinkhorn attention (Tay et al. 2020) and Reformer (Kitaev, Kaiser, and Levskaya 2020) adopted learnable patterns on the attention module. Vyas, Katharopoulos, and Fleuret (2020) proposed clustered attention that computes attention for only the centroids in clustered queries. Other works focus on kernel-based and feature mapping methods, like Performer (Choromanski et al. 2021), Reformer (Kitaev, Kaiser, and Levskaya 2020) and Linformer (Wang et al. 2020). Such methods improve self-attention efficiency by grouping, clustering or designing fix sparse patterns, at the expense of expressiveness. In contrast, Diffuser approximates full-attention using attention diffusion on a new sparse pattern, backed by a novel theoretically guaranteed graph expander perspective.

Diffusion on Graphs. In graph neural networks (GNNs), it is possible to increase number of layers to facilitate interactions with neighbors that are multiple hops away, but such indirect communication is less effective due to GNN aggregations and results in an increased computational cost. Another solution is to apply diffusion in each graph layer considering the multi-hop neighborhood (Xu et al. 2020; Atwood and Towsley 2016). (Klicpera, Bojchevski, and Günnemann 2019) proposed PPNP that applies personalized PageRank to propagate node predictions. (Klicpera, Weißenberger, and Günnemann 2019) propose GDC to allow propagation of multi-hop neighbors with generalized graph diffusion. Moreover, Wang et al. (2021) proposed MAGNA, which applies a diffusion based on the attention values in graph attention. Diffuser is inspired by the successful practice of diffusion in the graph domain, and utilizes it to improve the sparse Transformer expressiveness for general sequence modeling.

Diffuser: Multi-Hop Attention with Diffusion

In this section, we define the attention diffusion process and introduce the Diffuser model by integrating the attention diffusion into Transformers with sparse attention patterns.

Preliminaries

Multi-head Self-attention. Transformers and self-attention mechanism (Vaswani et al. 2017) are proposed for modeling sequences. The input sequence to the l-th layer with n tokens can be denoted as $H^{(l-1)} = [x_1, x_2, ..., x_n]$, where $H^{(l-1)} \in \mathbb{R}^{n\times d}$, and each token x_i is a d dimensional vector. The attention mechanism introduces matrices Q, K, V

as *queries*, *keys*, and *values*, which are linear projections of the input sequences:

$$Q = XW_Q, K = XW_K, V = XW_V. \quad (1)$$

The attention matrix A among tokens is then calculated as the scaled dot-product of queries and keys, and is used to calculate the updated token values:

$$Attn(X) = AV, A = \mathbf{softmax}\left(\frac{QK^\intercal}{\sqrt{d}}\right), \quad (2)$$

where softmax denotes the row-wise softmax normalization, and we omit the bias term for simplicity. To allow a token to attend to multiple aspects, self-attention can be further extended to multi-head self-attention as follows:

$$M\text{-}Attn(X) = \mathbf{cat}[Attn(X)_1, ..., Attn(X)_h]W_O, \quad (3)$$

where h is the number of heads in use.

Sparse attention. The runtime bottleneck of the standard self-attention is the attention matrix A with shape $n \times n$ in Equation 2, which has to be fully materialized in memory and scales quadratically as input length. This is impractical for long sequences with large n. To avoid the increased memory usage and speed up the attention calculations, we define the *sparse attention mechanism* described by a directed attention graph $G = (\mathcal{V}, \mathcal{E})$. In this graph, we have tokens to be the nodes, $\mathcal{V} = \{1, ..., n\}$, with the corresponding adjacency matrix $A \in \{0, 1\}^{n \times n}$. Each edge in the graph represents the query-key pair which we will include during computing sparse attention, i.e., $A_{i,j} = 1$ if query i attends to key j and is zero otherwise. Matrix A can be seen as a mask applied to the full-attention matrix by element-wise multiplication. The resulting sparse self-attention mechanism in Equation 2 can then be rewritten in the token-wise form as

$$Attn(x_i) = \mathbf{softmax}\left(\frac{Q_i K_{Ne(i)}^\intercal}{\sqrt{d}}\right) V_{Ne(i)}, \quad (4)$$

where x_i is the i-th input token to update value and $Ne(i)$ represents the neighbors of token i in the attention graph G.

Transformer Attention Diffusion

Similar to other sparse Transformers, the attention matrix A is first calculated on edges of the underlying graph G which is used to characterize the interaction strength between neighboring nodes on the graph, i.e.,

$$A_{i,j} = \frac{exp(Q_i K_j / \sqrt{d})}{\sum_{j \in Ne(i)} exp(Q_i K_j / \sqrt{d})}. \quad (5)$$

Each entry of attention matrix A is the attention score between 1-hop neighbors of G. Such 1-hop correlations in sparse Transformers cannot include all possible correlations compared to full-attention, which leads to limitations to capturing important correlations when the true dependencies are in several-hops away and not directly connected by edges in the graph as discussed in Figure 1.

The key idea of Diffuser is to apply the attention diffusion mechanism to calculate the multi-hop token relationships on the attention graph based on attention weights on edges. The multi-hop attention scores are calculated as entries of the graph diffusion matrix $\mathcal{A}$:

$$\mathcal{A} = \sum_{k=0}^{\infty} \theta_k A^k, \quad (6)$$

where A is the adjacency matrix or calculated sparse attention matrix, and the weighting coefficient θ_k satisfies $\sum_{k=0}^{\infty} \theta_k = 1, \theta_k \in [0, 1]$. The original receptive fields defined by the sparse attention pattern will be gradually expanded as k becomes larger. The resulting attention score $\mathcal{A}_{i,j}$ incorporates all paths between token i and j, weighted by the coefficient θ_k. We then multiply each value vector V by the diffusion attention matrix $\mathcal{A}$, which is equivalent to the message aggregation step in GNN.

Computing the power of attention matrices in Equation 6 can be inevitably expensive for long sequences, even when the sparsity is considered. To efficiently apply the diffusion mechanism in Transformers, we implement the graph diffusion process as Personalized PageRank (PPR) by specifying $\theta_k = \alpha(1 - \alpha)^k$ with teleport probability α. The resulting diffusion matrix $\mathcal{A} = \sum_{k=0}^{\infty} \alpha(1 - \alpha)^k A^k$ is the power expansion of the solution to the recursive equation $\mathcal{A} = \alpha I + (1 - \alpha)\mathcal{A}A$. We then adopt the power iteration method (Page et al. 1999) to achieve linear computational complexity by approximating PPR within the the first K diffusion steps. Each power iteration (diffusion) step is calculated as

$$Z_{(0)} = V = XW_V, \; Z_{(k+1)} = (1 - \alpha)AZ_{(k)} + \alpha V, \quad (7)$$

for $0 \leq k < K$. Z_K is output of the attention diffusion process, and will converge to the real output $\mathcal{A}V$ as $K \to \infty$ (shown in Appendix).

Sparse Pattern Design

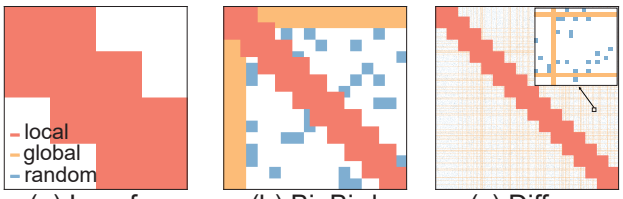

Figure 2: Comparison of sparse patterns (1024×1024) with different types of attentions.

Another important ingredient of Diffuser is the design of sparse attention pattern. It should be noted that attention diffusion is compatible with any sparse patterns. We design new sparse patterns to leverage the advantages of attention diffusion while maintaining computational efficiency. As shown in Figure 2, we consider a combination of local window attention, global attention, and random attention to capture token interactions without quadratic complexity dependency on the sequence length.

Local window attention. Local window attentions are constructed by the sliding window and are proposed to model

the information locality among neighboring tokens, e.g., the proximity of tokens in linguistic structure and the clustering coefficient in the graph. Given a fixed window size w, each token attends to $\frac{1}{2}w$ tokens on each side, and we also consider the cross-window attention by overlapping $\frac{1}{2}w$ tokens, resulting in computational complexity $\mathcal{O}(nw^2)$ which scales linearly with input sequence length n. The resulting receptive field (keys that each query looks up to calculate self-attention) is expanded linearly with more diffusion steps and Transformer layers. For example, the size of the receptive field grows as $(\frac{1}{2}+k)w$ with diffusion step k. Compared to Longformer and BigBird, Diffuser can achieve good expressiveness with smaller local window and therefore sparser attention, because of the fast receptive field expansion by attention diffusion given the same number of attention layers.

Global attention. We introduce global attention by extending the receptive field of tokens to the entire input sequence. Specifically, we randomly choose g tokens among input sequence as global tokens, such that for any global token i, $A_{i,:} = 1$ and $A_{:,i} = 1$, resulting in complexity $\mathcal{O}(gn)$. Global tokens share the same set of weight parameters with other types of attentions (in contrast to different weights used in Longformer). Furthermore, compared to BigBird which selects global attentions by grouping adjacent tokens, Diffuser constructs global attention with the unit of individual tokens.

Random attention. We consider adding random attentions to accelerate the information flow between any pair of nodes. The intuition of introducing random attention is to enhance the graph expander properties for better full-attention approximation. From the graph theory perspective, random graph, e.g., Erdős–Rényi graph (Erdős, Rényi et al. 1960), has been shown to have good expander properties to approximate the complete graph (full-attention) spectrally (detailed in the next section). Therefore, for each input token i, we randomly select r tokens ($r \ll n$ and above the threshold $\mathcal{O}(log(n)/n)$), such that $A_{i,j} = 1$ for each selected token j, resulting in a total number of $\mathcal{O}(rn)$ random attentions. Compared to BigBird whose random attentions are based on the unit of blocks (e.g., 64 adjacent tokens as a block), Diffuser constructs random attention with the unit of individual tokens. Given the same number of global and random attention budget, the token-wise selection leads to more uniform attention distributions with weaker clustering, compared to block-wise selections, which improves the expander properties and accelerates the attention flows among tokens. It is noted that the reason BigBird adopts block-wise attentions is to blockify lookups for efficient implementations of attention calculations. In comparison, we implement token-wise attention using commercial graph packages with optimized GSpMM kernels and achieve similar efficiency.

Model Architecture

We introduce the building block of Diffuser, based on the proposed sparse pattern, regular self-attention mechanism, and attention diffusion process, as shown in Figure 3. At layer $l-1$, input $H^{(l-1)}$ is mapped to queries, keys, and values, and attention scores are calculated on edges of the

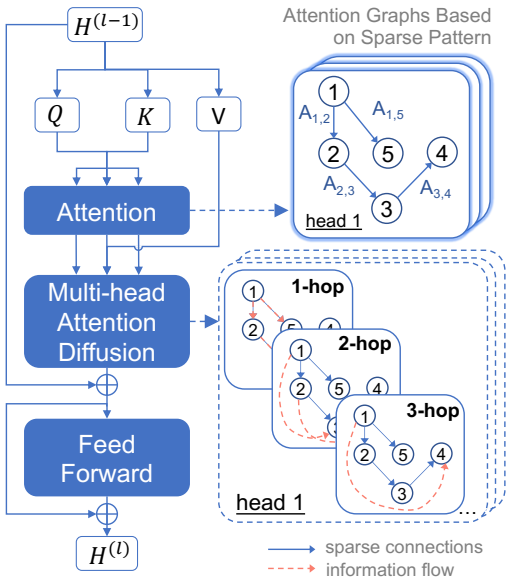

Figure 3: The layer architecture of Diffuser.

predefined attention graph using scaled dot-product. Then, attention diffusion procedure is calculated following Equation 7 up to K diffusion steps, which spreads the information of tokens to its multi-hop neighbors. The residual and feed-forward layers are then used to output $H^{(l)}$ for skip-connections and value mappings.

Theoretical Expressive Power of Diffuser

In this section, we investigate the expressiveness of the proposed model from two perspectives. First, we show Diffuser's capability to *approximate sequence modeling* by proving that the model with sparse connections and diffusion is a universal approximator to sequence-to-sequence function, and it requires less layers to achieve the same expressivity compared with sparse attentions without diffusion. Second, we show Diffuser's capability to *approximate full-attention*. From the spectral graph perspective, we show that the proposed sparse patterns combined with diffusion induces better graph expander properties, enabling approximations of the complete graph.

Diffuser as Universal Approximators

We follow the proof of Yun et al. (2019) and show Diffuser can approximate arbitrary sequence-to-sequence functions (mapping sequential input X from $\mathbb{R}^{n\times d}$ to $\mathbb{R}^{n\times d}$). Given one family of Diffuser structure $\mathcal{D}^{h,m,r}$ with h attention

Length	Longformer	BigBird	Diffuser			
			tot	loc	glob	rand
1024	62.5	55.7	**24.0**	18.0	4.2	1.9
2048	34.4	32.5	**15.5**	9.2	4.2	2.1
4096	18.0	16.9	**11.2**	4.6	4.3	2.2

Table 1: The percentage of attentions: `tot`, `loc`, `glob`, and `rand` represent total, local, global, and random attentions, respectively.

heads of head size m and hidden layers of width r, we state the main theorem as follows, which shows that if the sparse pattern in use satisfies the assumptions below, there exists Diffuser belonging to $\mathcal{D}^{h,m,r}$ that is a universal approximator of continuous sequence-to-sequence function.

Theorem 1. *Consider any continuous function $f \in \mathcal{F}$, and the class of Diffuser $\mathcal{D}^{h,m,r}$ with sparse attention graph satisfying Assumption 1. Then, for any $\epsilon > 0$ and $1 \leq q < \infty$, there exists a function $g \in \mathcal{D}^{h,m,r}$ such that*

$$d_q(f, g) := \left(\int_{\mathbb{D}} \|f(\boldsymbol{X}) - g(\boldsymbol{X})\|_q^q \, d\boldsymbol{X} \right)^{1/q} \leq \epsilon. \quad (8)$$

Compared to other works discussing expressiveness of sparse Transformers (Zaheer et al. 2020; Yun et al. 2020), we show Diffuser can achieve contextual mappings using fewer layers based on attention diffusion mechanism. Intuitively, the improved efficiency can be understood as the expanded attention receptive field through diffusion, which includes more attentions without stacking attention layers. We then specify a set of conditions on the sparse attention patterns A of the attention graph G in study.

Assumption 1. *Sparsity pattern A satisfies the following:*

1. *All tokens attend to themselves, i.e., for all $k \in [n]$, we have $k \in Ne(k)$.*
2. *The graph G is connected and has a Hamiltonian path connecting all nodes, i.e., there exists a permutation $\gamma : [n] \to [n]$ such that, for all $i \in [n-1]$, $\gamma(i) \in Ne(\gamma(i+1))$.*

The detailed proof is shown in Appendix, and the key innovation here is that the introduction of attention diffusion allows sequence ID computation to involve all token values within one attention layer.

Diffuser as Expander Graphs

Expander graphs are sparse and robust graphs with strong connectivity, and have several nice properties to improve the expressiveness of Diffuser while keeping the computational efficiency. In this subsection, we show the sparse attention graph in Diffuser has good expander graph property, and then highlight three advantages of constructing attention graph as an expander graph, including ensuring sparsity, mixing diffusion rapidly, and approximating full-attention.

We consider the family of d-regular graphs G with adjacency matrix A, which require all vertices to have the same degree d, and we then define the (ϵ, d)-expander:

Definition 1. *A graph G is a (d, ϵ)-expander if it is d-regular and its adjacency matrix eigenvalues satisfy $|\mu_i| \leq \epsilon d$ for $i \geq 2$.*

As the Laplacian eigenvalues of regular graph are given by $\lambda_i = d - \mu_i$, this is equivalent to $|d - \lambda_i| \leq \epsilon d$. We show in Appendix the equivalent definitions using expansion ratio and the properties of eigenvalues of d-regular graph. One common random graph model used to build such expander graphs is Erdős–Rényi $\mathcal{G}_{n,p}$ model where each edge is included in the graph with probability p, and we consider the variant $\mathcal{G}_{n,m}$ model where m edges are randomly drawn, further constrained to the regularity d. These two models are very similar if $p \geq logn/n$ which is satisfied in the long-sequence scenario. It can be proved that such randomly built d-regular is an expander with high probability (Friedman 2008). To ensure good expander graph properties, we follow such random models to build the random attention graph which can be thought of as a (r, ϵ)-expander, as discussed in the previous section (additional connections from local and global pattern will not harm expander properties).

The next theorem shows that such sparse random attention with (r, ϵ)-expander properties can approximate the full-attention complete graph (proved in Appendix).

Definition 2. *For two graph G and H, we say G is an ϵ-approximation to H if $(1+\epsilon)H \succeq G \succeq (1-\epsilon)H$, where $G \succeq H$ means the corresponding Laplacian matrix $L_G - L_H$ is positive-semidefinite.*

Theorem 2. *For every $\epsilon > 0$, there exists d such that for all sufficiently large n, there is a d-regular graph G which is an ϵ approximation of the complete graph K_n.*

Further we notice the nice properties from diffusion transformation as low-pass filters can further enhance the expander properties. The eigenvalues $\tilde{\mu}_i$ of diffusion matrix $\mathcal{A}$ can be computed as $\tilde{\mu}_i = \alpha \sum_{k=0}^{\infty} (1-\alpha)^k \mu_i^k = \frac{\alpha}{1-(1-\alpha)\mu_i}$ in the PPR case, which amplifies low Laplacian eigenvalues while suppressing high eigenvalues (shown in Appendix).

Another essential reason for Diffuser designing expander graphs is that they achieve rapid mixing for random walks and diffusion, which accelerates the information propagation on the attention graph of Diffuser.

Theorem 3. *Given d-regular graph with adjacency matrix A and transition matrix $\hat{A} = \frac{1}{d}A$ of random walk, assume the spectral gap σ is defined by $\sigma = max(|\mu_2|, |\mu_n|) \triangleq \beta d$. Then,*

$$||\hat{A}^t v - u||_1 \leq \sqrt{n}\beta^t, \quad (9)$$

where u is the stationary distribution and v is an arbitrary initial distribution.

The theorem shows that PPR (or general random walk) approaches its limiting probability distribution rapidly on expander graph which has large spectral gap (proved in Appendix). The fast convergence of PPR on expander graph indicates accelerated information propagation in Diffuser.

Experiments

We evaluate the performance of Diffuser with a rich set of sequence modeling tasks, including language modeling, image modeling and Long-Range Arena (LRA) tasks. We then analyze expressiveness and efficiency through extensive ablation studies.

Model Implementations

We implement Diffuser using the graph library DGL , which offers optimized kernels for sparse matrix operations. We first build the graph according to the sparse pattern, then follow the message passing framework defined in DGL by calculating the sparse attention as *message functions*, and attention diffusion as *update and reduce function*. The remaining

components follow the regular Transformer architecture in PyTorch. The detailed experimental settings, hyperparameters and baseline setup are discussed in Appendix.

Efficiency. We show the GPU memory usage and runtime comparison in Figure 4. Compared to benchmarks, Diffuser achieves 1.67× memory savings compared to the best baseline Performer, with comparable running time. It should be noted that the runtime can be further improved with better diffusion sparse operation support from hardware.

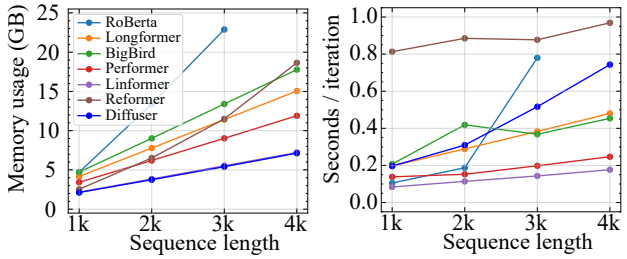

Figure 4: Comparisons of computational efficiency: memory usage and seconds/iteration.

Language Modeling

Pretraining. We evaluate the model on language tasks following the standard pretraining and finetuning pipeline (Liu et al. 2019). Diffuser is pretrained with masked language modeling (MLM) task, which involves predicting a random subset of tokens that have been masked out. We pretrain the model with three standard datasets (detailed in Appendix) and evaluate the pretraining performance with bits per character (BPC) as in Zaheer et al. (2020). The training is conducted with the maximum sequence length of 4,096 and linear warmup from the RoBERTa checkpoint. As shown in Table 2, Diffuser achieves lower BPC compared to benchmarks after training for 50K steps. The significant difference between initialization and training 10K steps for Diffuser model indicates the RoBERTa weights are not working well because of the change of the updating rule, and the model is learning to better utilize the attention diffusion.

Model	BPC	Model	BPC
RoBERTa	2.02	Diffuser-init	3.52
Longformer	1.86	Diffuser-10K steps	1.96
BigBird	1.82	Diffuser-50K steps	**1.68**

Table 2: MLM BPC for Diffuser and baselines.

Text classification. We first evaluate Diffuser on text classification tasks with five datasets. **Hyperpartisan** (Kiesel et al. 2019) and **20NewsGroups** (Lang 1995) are news datasets with different scales. **IMDB** (Maas et al. 2011) is a collection of movie reviews for sentiment classification. Moreover, we select and propose two new benchmarks with longer documents based on an existing large-scale corpus, Amazon product reviews (He and McAuley 2016), to conduct long document classification. **Amazon-512** contains all reviews that are longer than 512 words from the *Electronics* category; **Amazon-2048** contains 10,000 randomly sampled

	HYP	20NG	IMDB	A-512	A-2048	Avg.
95pt.	2,030	1,229	771	1,696	5,216	-
BERT	85.7	85.3	91.3	59.2	50.3	74.36
RoBERTa	87.4	85.7	95.3	65.0	57.9	78.26
BigBird	92.2	82.3	95.2	67.4	63.6	80.14
Longformer	93.8	86.3	**95.7**	67.3	61.2	80.86
BigBird_D	93.1	84.5	95.0	**68.2**	63.4	80.84
Longformer_D	93.5	**87.3**	95.4	67.0	62.5	81.24
Diffuser	**94.4**	86.8	95.2	67.8	**64.8**	**81.80**

Table 3: Text classification results on five datasets: Hyperpartisan (HYP), 20NewsGroups (20NG), IMDB, Amazon-512 (A-512) and Amazon-2048 (A-2048). `95pt.` indicates 95th percentile of token number. We report average F1 scores (Avg.). We underscore the best among baselines, and bold the best overall models.

reviews that are longer than 2,048 words from the *Books* category. We randomly split 8/1/1 as train/dev/test sets for both datasets (statistics detailed in Appendix). We finetuned the pretrained Diffuser on each dataset and compare the average F1 score with benchmark models. To investigate the influence of attention diffusion, we also apply attention diffusion to Longformer and BigBird models based on their respective sparse patterns (*BigBird_D* and *Longformer_D*). As shown in Table 3, diffusion-based methods consistently achieve better average score, indicating the importance of attention diffusion. Among them, Diffuser achieves the best average performance, showing the effect of the proposed sparse pattern on attention diffusion. Especially, Diffuser outperforms BigBird by 0.4% and 1.2% on two long Amazon datasets, which shows its stronger ability to model long sentences.

Question answering. We choose two benchmarks for question answering: WikiHop (Welbl, Stenetorp, and Riedel 2018) and TriviaQA (Joshi et al. 2017). WikiHop is a dataset collected based on Wikipedia articles for multi-hop question answering across documents. TriviaQA is a large-scale dataset of question-answer-evidence pairs for reading comprehension. Both datasets are in a reasonable scale and length, as in Table 4. We follow Beltagy, Peters, and Cohan (2020) and concatenate the question, answer, and candidates into one input sequence with special tokens along the context. Task specific projection layers are then adopted to classify the correct answers for WikiHop and predict the answer span for TriviaQA. From Table 4, we see that Diffuser achieves the best results for TriviaQA and has comparable performance for WikiHop.

Model	WikiHop	TriviaQA	
Metric	Acc	F1	EM
RoBERTa	71.82	74.02	66.87
Longformer	75.30	74.82	67.24
BigBird	74.54	73.16	68.26
Diffuser	**75.80**	**75.84**	**70.20**

Table 4: Comparison of WikiHop and TriviaQA, and model performances. We report Accuracy for WikiHop, and F1, EM score for TriviaQA.

Image Generative Modeling

We then evaluate the performance of Diffuser on image density modeling task with CIFAR-10 and ImageNet-64. The sequence lengths are 3,072 and 12,288, respectively. We follow the setting of (Child et al. 2019) and adopt an 8-layer model with 512 hidden dimensions which is trained until the validation errors stop decreasing. As shown in Table 5, Diffuser achieves lower bits per dimension (BPD) on CIFAR-10 datasets and converges to similar BPD on ImageNet64 dataset, demonstrating the effectiveness of the model in the image domain. Similar results are obtained with different layers and hidden dimensions.

CIFAR-10	BPD	ImageNet-64	BPD
PixcelCNN	3.03	PixcelCNN	3.57
PixcelCNN+	2.92	Parallel Multiscale	3.70
PixelSNAIL	2.85	SPN	3.52
Sparse Trans.	2.80	Sparse Trans.	**3.44**
Diffuser	**2.78**	Diffuser	**3.44**

Table 5: Bits per Dimension (Bits/Dim) on CIFAR-10 and ImageNet-64. We list baseline details in Appendix.

Long-Context Sequence Modeling

Long Range Arena (LRA) (Tay et al. 2021b) is a unified benchmark for evaluating efficient Transformer models with five multi-class classification tasks from different domains, including ListOps, byte-level text classification, byte-level document retrival, image classification, and image-based path finder. All the tasks are multi-class classification with input sequences of different lengths. As shown in Table 6, Diffuser achieves the best results on ListOps (2K), Retrieval (4K), and Image (1K), improving average accuracy by 2.30% compared to the best benchmark BigBird.

Models	ListOps	Retrieval	Image	Pathfinder	Avg
Full	36.37	57.46	42.44	71.40	54.39
Local Att	15.82	53.39	41.46	66.63	46.06
Linear	16.13	53.09	42.34	75.30	50.55
Reformer	37.27	53.40	38.07	68.50	50.67
Sparse	17.07	59.59	44.24	71.71	51.24
Sinkhorn	33.67	53.83	41.23	67.45	51.48
Linformer	35.70	52.27	38.56	76.34	51.36
Performer	18.01	53.82	42.77	**77.05**	51.41
Synthesizer	36.99	54.67	41.61	69.45	52.88
Combiner	36.65	59.81	41.67	71.52	54.93
Longformer	35.63	56.89	42.22	69.71	53.46
BigBird	36.05	59.29	40.83	74.87	55.01
Diffuser	**37.52**	**61.28**	**45.20**	76.58	**57.31**

Table 6: Classification accuracy on LRA datasets with three best performing benchmarks on average. Underline values are best among baselines, while bold are the best.

Ablation Studies

We first study the influence of different mechanisms used in Diffuser by ablating the corresponding components. Table 7 shows that the diffusion (#4) and local patterns (#1) have the biggest influence on the performance while random (#2) and global attentions (#3) result in similar performance drop.

Model	ListOps	Retrieval	Image	Pathfinder	Avg
#0 Diffuser	**37.52**	61.28	**45.20**	**76.58**	**57.31**
#1 w/o loc.	35.28	58.05	38.07	73.25	52.65
#2 w/o rand.	36.38	60.60	43.48	73.36	55.74
#3 w/o glob.	36.52	61.07	42.35	72.47	55.19
#4 w/o diff.	33.48	58.25	39.28	71.08	52.36
#5 w uni.	37.39	**61.42**	44.98	76.30	57.07

Table 7: Ablation studies of each component in Diffuser.

We also notice that changing the random attention into uniform distribution (#5) does not substantially affect the performance as the expansion properties are retained (shown in Appendix). We then investigate the effect of the diffusion parameter using A-2048 datasets as shown in Figure 5. We observe significant improvement in performance as K increases, and the saturated performance under $K \geq 5$ indicates the convergence to the stationary distribution. We also observe the performance is significantly influenced by the teleport parameter α, and we choose $K = 5$ and $\alpha = 0.1$ in practice. We also show the influence of different types of attentions in Figure 6. The performance improvements slow down as we increase the number of attentions for all three types of attentions, and we choose the number of attentions considering the balance between expressiveness and efficiency. We also observe that there exists an optimal ratio to combine the random and global attentions and improve performance upon random-attentions-only or global-attentions-only scenarios. More ablation studies (e.g., input robustness analysis) are shown in Appendix.

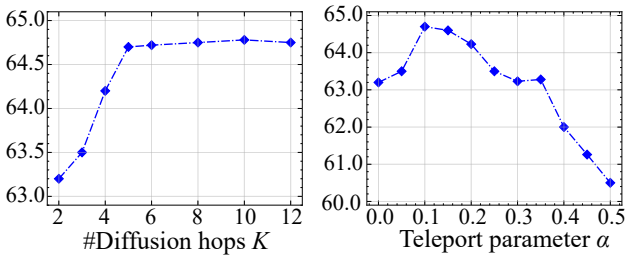

Figure 5: The influence of diffusion parameters on accuracy.

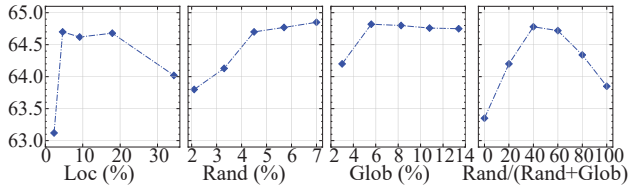

Figure 6: The influence of different attention on accuracy.

Conclusion

In this work, we proposed Diffuser, an efficient Transformer for long sequence modeling that applies multi-hop attention diffusion. We theoretically showed that Diffuser is a more efficient universal approximator for sequence modeling, with better expander properties from the graph spectral perspective. Experimentally, we showed that Diffuser achieves superior performance in language modeling, image modeling, and other long sequence modeling tasks.

References

Ainslie, J.; Ontanon, S.; Alberti, C.; Cvicek, V.; Fisher, Z.; Pham, P.; Ravula, A.; Sanghai, S.; Wang, Q.; and Yang, L. 2020. ETC: Encoding long and structured inputs in transformers. *arXiv preprint arXiv:2004.08483*.

Atwood, J.; and Towsley, D. 2016. Diffusion-Convolutional Neural Networks. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 1993–2001.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. *CoRR*, abs/2004.05150.

Child, R.; Gray, S.; Radford, A.; and Sutskever, I. 2019. Generating Long Sequences with Sparse Transformers. *CoRR*, abs/1904.10509.

Choromanski, K.; Likhosherstov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlos, T.; Hawkins, P.; Davis, J.; Mohiuddin, A.; Kaiser, L.; et al. 2020. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*.

Choromanski, K. M.; Likhosherstov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlós, T.; Hawkins, P.; Davis, J. Q.; Mohiuddin, A.; Kaiser, L.; Belanger, D. B.; Colwell, L. J.; and Weller, A. 2021. Rethinking Attention with Performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dwivedi, V. P.; and Bresson, X. 2020. A Generalization of Transformer Networks to Graphs. *CoRR*, abs/2012.09699.

Erdős, P.; Rényi, A.; et al. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1): 17–60.

Friedman, J. 2008. *A proof of Alon's second eigenvalue conjecture and related problems*. American Mathematical Soc.

He, R.; and McAuley, J. J. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In Bourdeau, J.; Hendler, J.; Nkambou, R.; Horrocks, I.; and Zhao, B. Y., eds., *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, 507–517. ACM.

Ho, J.; Kalchbrenner, N.; Weissenborn, D.; and Salimans, T. 2019. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*.

Huynh, E. 2022. Vision Transformers in 2022: An Update on Tiny ImageNet. *CoRR*, abs/2205.10660.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Katharopoulos, A.; Vyas, A.; Pappas, N.; and Fleuret, F. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, 5156–5165. PMLR.

Kiesel, J.; Mestre, M.; Shukla, R.; Vincent, E.; Adineh, P.; Corney, D.; Stein, B.; and Potthast, M. 2019. SemEval-2019 Task 4: Hyperpartisan News Detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 829–839. Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Kitaev, N.; Kaiser, L.; and Levskaya, A. 2020. Reformer: The Efficient Transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Klicpera, J.; Weißenberger, S.; and Günnemann, S. 2019. Diffusion Improves Graph Learning. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 13333–13345.

Lang, K. 1995. NewsWeeder: Learning to Filter Netnews. In Prieditis, A.; and Russell, S., eds., *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, 331–339. Morgan Kaufmann.

Li, I.; Feng, A.; Wu, H.; Li, T.; Suzumura, T.; and Dong, R. 2022. LiGCN: Label-interpretable Graph Convolutional Networks for Multi-label Text Classification. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, 60–70. Seattle, Washington: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Peng, H.; Pappas, N.; Yogatama, D.; Schwartz, R.; Smith, N. A.; and Kong, L. 2021. Random feature attention. *arXiv preprint arXiv:2103.02143*.

Ren, H.; Dai, H.; Dai, Z.; Yang, M.; Leskovec, J.; Schuurmans, D.; and Dai, B. 2021. Combiner: Full attention transformer with sparse computation cost. *Advances in Neural Information Processing Systems*, 34: 22470–22482.

Shen, Z.; Zhang, M.; Zhao, H.; Yi, S.; and Li, H. 2021. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 3531–3539.

Tay, Y.; Bahri, D.; Metzler, D.; Juan, D.-C.; Zhao, Z.; and Zheng, C. 2021a. Synthesizer: Rethinking self-attention for transformer models. In *International conference on machine learning*, 10183–10192. PMLR.

Tay, Y.; Bahri, D.; Yang, L.; Metzler, D.; and Juan, D. 2020. Sparse Sinkhorn Attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 9438–9447. PMLR.

Tay, Y.; Dehghani, M.; Abnar, S.; Shen, Y.; Bahri, D.; Pham, P.; Rao, J.; Yang, L.; Ruder, S.; and Metzler, D. 2021b. Long Range Arena : A Benchmark for Efficient Transformers. In *International Conference on Learning Representations*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Vyas, A.; Katharopoulos, A.; and Fleuret, F. 2020. Fast Transformers with Clustered Attention. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wang, G.; Ying, R.; Huang, J.; and Leskovec, J. 2021. Multi-hop Attention Graph Neural Networks. In Zhou, Z., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, 3089–3096. ijcai.org.

Wang, S.; Li, B. Z.; Khabsa, M.; Fang, H.; and Ma, H. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Welbl, J.; Stenetorp, P.; and Riedel, S. 2018. Constructing Datasets for Multi-hop Reading Comprehension Across Documents. *Trans. Assoc. Comput. Linguistics*, 6: 287–302.

Xu, B.; Shen, H.; Cao, Q.; Cen, K.; and Cheng, X. 2020. Graph Convolutional Networks using Heat Kernel for Semi-supervised Learning. *CoRR*, abs/2007.16002.

Yang, J.; Liu, Z.; Xiao, S.; Li, C.; Lian, D.; Agrawal, S.; Singh, A.; Sun, G.; and Xie, X. 2021. GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 28798–28810.

Yun, C.; Bhojanapalli, S.; Rawat, A. S.; Reddi, S. J.; and Kumar, S. 2019. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*.

Yun, C.; Chang, Y.-W.; Bhojanapalli, S.; Rawat, A. S.; Reddi, S.; and Kumar, S. 2020. O (n) connections are expressive enough: Universal approximability of sparse transformers. *Advances in Neural Information Processing Systems*, 33: 13783–13794.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontañón, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; and Ahmed, A. 2020. Big Bird: Transformers for Longer Sequences. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhu, J.; Xia, Y.; Wu, L.; He, D.; Qin, T.; Zhou, W.; Li, H.; and Liu, T. 2020. Incorporating BERT into Neural Machine Translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.