IMPROVING VISION ATTENTION WITH RANDOM WALK GRAPH KERNEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision transformers, which propose to tokenize an image and introduce attention mechanism to learn cross-token relationship, have advanced many computer vision tasks. However, the attention module owns a quadratic computational complexity and hence suffers from slow computing speed and high memory cost, hindering it from handling long sequences of tokens. Some attempts optimize the quadratic attention with linear approximation yet observe undesired performance drop. This work balances the trade-off between modeling efficiency and capacity of vision attention. We notice that, by treating queries and keys as nodes in a graph, existing algorithms are akin to modeling one-step interaction between nodes. To strengthen the cross-node connection for a more representative attention, we introduce multi-step interaction, which is equivalent to solving an inverse matrix as in random walk graph kernel. We then come up with a new strategy to construct queries and keys, with the help of bipartite graph, to ease the calculation of matrix inversion. The effectiveness of our approach is verified on various visual tasks. We achieved the competitive results on the semantic segmentation task with 15% fewer parameters and 10-25% less computation. In addition, the vision transformer based quantization method can be applied to 512×512 or even 1024×1024 resolution images. Code will be made publicly available.

1 INTRODUCTION

Transformer, a powerful tool for natural language processing, has recently opened up favorable prospects for solving computer vision tasks (Dosovitskiy et al., 2021; Touvron et al., 2021). Such tremendous success mainly comes from the attention mechanism, which is primarily designed for sequential data modeling. Specifically, in vision transformers (Dosovitskiy et al., 2021; Touvron et al., 2021; Wu et al., 2021; Liu et al., 2021; Wang et al., 2021), an image is first divided into patches and then converted to a sequence of tokens. The attention module helps learn the relationship between tokens, resulting in an overall understanding of the given image.

Despite its capability in representation learning, a vision transformer is usually faced with quadratic computational complexity growth along with the sequence getting longer. That is because attention is typically asked to connect *every* two tokens to ensure a comprehensive interpretation of the input. Considering the forward speed and memory usage, such a drawback hinders the model from using a smaller patch size[1] or training on higher-resolution images. To cut the computing cost, existing attempts propose to replace full attention with non-generic sparse attention (*e.g.*, not relating all token pairs) (Child et al., 2019; Parmar et al., 2018; Ho et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020), or to simplify the attention computation with linear approximation (Choromanski et al., 2021; Wang et al., 2020; Xiong et al., 2021). However, linear attention may suffer from insufficient modeling capacity and cause performance deterioration accordingly (Zhu et al., 2021; Wu et al., 2021).

In this work, we target balancing the trade-off between modeling efficiency and capacity of vision attention. For this purpose, we revisit self-attention (*i.e.*, the set of query tokens is identical to the

[1]It is recently shown that smaller patch size is beneficial to the performance of vision transformers (Dosovitskiy et al., 2021; Wang et al., 2021; Xie et al., 2021).

set of key tokens) by forming tokens into a graph. We find that existing formulations (Beltagy et al., 2020; Xie et al., 2021; Kitaev et al., 2020; Shen et al., 2021) only characterize the one-step interaction between graph nodes, leaving space for strengthening the node connection via introducing other nodes as the intermediate steps. In other words, in addition to the one-edge (*i.e.*, direct) relationship in the graph, we would like to also learn the multi-edge (*i.e.*, indirect) relationship, encouraging the attention to capture more information from the input image. Motivated by this, we extend the conventional attention with multi-step node interaction. It turns out that such an improved version is equivalent to finding the inverse of a matrix as in random work graph kernel (Katz, 1953). However, matrix inversion own a cubic complexity. To speed up the calculation, we present a carefully designed strategy to construct queries and keys, drawing support from bipartite graph (Liu et al., 2010). That way, we are able to perform full attention in a linear time regarding the sequence length.

Extensive experiments demonstrated that our model is applicable on different kind of vision tasks. Without special design and tuning, our model can be directly used to replace the simplified attention design of the visual transformer for long sequence inputs. In particular, we achieved the comparable results on the semantic segmentation task with 15% fewer parameters and 10-25% less computation. Moreover, the quantization of high-resolution images with small patch becomes possible when the computational optimization of self-attention is performed using our model. Compared to other linear attention optimizations with a similar structure, our model has significant advantages in terms of speed or image reconstruction quality.

2 Related Work

Vision transformer. Vision transformer (ViT) (Dosovitskiy et al., 2021; Touvron et al., 2021; Liu et al., 2021) have recently dominated a wide range of tasks (*e.g.*, classification, segmentation and generation) in computer vision community. It splits the images into discrete nonoverlapping patches that are treated as a sequence of tokens. For image classification task, the ViT has shown to outperform convolutional neural networks (e.g., ResNet (He et al., 2016)) with sufficient training data. In recent works (*e.g.*, PVT (Wang et al., 2021; 2022), Swin-Transformer (Liu et al., 2021), ViL (Zhu et al., 2021), CvT (Wu et al., 2021)), they have shown that smaller patch size (or longer sequences of image patches) is beneficial to the performance of vision transformers. Specifically, Convolutional Vision Transformer (CvT) (Wu et al., 2021), stack a pyramid of ViTs to form a multi-scale architecture and model long sequences of image patches at much higher resolution (*e.g.*, 96 $\times$96 = 9216 patches for images with 384^2 image size). Besides using vision transformer in classification, SETR (Zheng et al., 2021) adopts ViT-based encoder following with several CNN decoders to semantic segmentation and achieve good performance. After this, Segformer (Xie et al., 2021) design a pooling-like efficient attention mechanism and make an input image as a sequence of length $128 \times 128 = 16384$, obtained greater performance. In addition, vision transformer is also widely adopted in unsupervised learning for generative model or pre-train model. VIT-VQGAN applied vision transformer into a VQVAE architecture and achieved excellent image quantization ability. This vision transformer based VQVAE is chosen to be backbone in BeiT-2 (Peng et al., 2022) and used as the first stage model of 256-resolution image synthesis stage in Parti (Yu et al., 2022).

Efficient transformer. In recent years, many efficient transformers are proposed to deal with the quadratic cost of vanilla self-attention mechanism, which make improvements around computational and memory efficiency. They can be categorized as follows: 1) Sparse attention mechanism reduces the dense attention matrix to a sparse version by predefined patterns (*e.g.*, chunking paradigm),including Sparse Transformer (Child et al., 2019), Image Transformer (Parmar et al., 2018), Axial Transformer (Ho et al., 2019), Longformer (Beltagy et al., 2020), ETC (Ainslie et al., 2020) and Big Bird (Zaheer et al., 2020). 2) Low-rank projection attention mechanism is to improve efficiency by leveraging low-rank approximations of the full self-attention matrix, including Linformer (Wang et al., 2020), Nyströmformer (Xiong et al., 2021), Synthesizer (Tay et al., 2021). 3) Kernel-based approximation of the attention matrix is to view the attention mechanism through kernelization (*e.g.*, the unbiased estimation of the Gaussian kernel), including Performer (Choromanski et al., 2021), Linear Transformer (Katharopoulos et al., 2020), and Random Feature Attention (Peng et al., 2021). Besides that, some models utilize hybrid attention mechanisms, such as Long-Short Transformer (Zhu et al., 2021) seamlessly integrates both low-rank

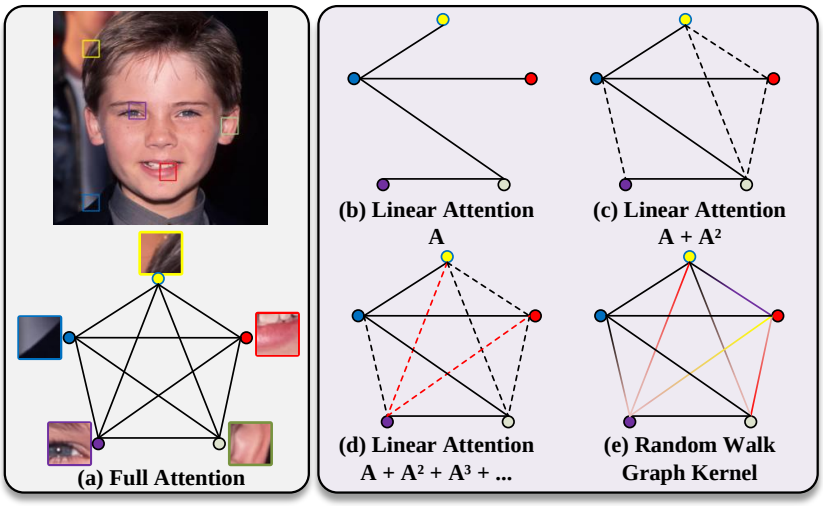

Figure 1: **Concept diagram** of our motivation. (a) A graph view of traditional self-attention. (b) A weak linear attention can only model limited attention relationship. (c) 2-step interactions based on a weak linear attention. (d) n-step interactions based on a weak linear attention. (e) Random Walk Graph Kernel $(\mathbf{I} - \lambda\mathbf{A})^{-1}$.

projection and sparse attentions. We notice that these existing efficient transformers are designed based on the one-step interaction between graph nodes, and they neglect to consider the multi-step interaction strategy. Thus, we introduce multi-step interaction into the attention mechanism via random walk graph kernel, encouraging the attention to capture more information from the input image.

3 METHOD

In this section, we first revisit the traditional self-attention mechanism, then discuss the basic idea of our random walk graph kernel, and finally give some structure and complexity analysis.

3.1 RETHINKING SELF-ATTENTION

The traditional self-attention mechanism outputs the feature of data by modeling the cross-token relationship among the input sequence. Specifically, an input sequence of N tokens of dimension d, $\mathbf{X} \in \mathbb{R}^{N\times d}$, is projected to $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V}$, called the query, key and value of input:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \mathbf{K} = \mathbf{X}\mathbf{W}_K, \mathbf{V} = \mathbf{X}\mathbf{W}_V. \tag{1}$$

Then, the popular scale dot-product self-attention (denoted as attention in the following paper) can be formulated as:

$$\text{Attention}(\mathbf{X}) = \mathbf{A}\mathbf{V} = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \tag{2}$$

where the $\text{Softmax}(\cdot)$ function normalizes the input matrix by row. Inspired by Zaheer et al. (2020), we interpret a general attention mechanism as a directed graph G that takes input tokens $[n] = \{1, 2, \ldots, N\}$ as vertices. A general attention is a well-defined weighted adjacent matrix $\mathbf{A}$ of G based on the attended-relationship between two corresponding vertices. Without loss of generality, the matrix $\mathbf{A}$ satisfies some basic properties, such as $\mathbf{A} \in [0, 1]^{N\times N}$ and for any $i = 1, 2, \ldots, N$, $\sum_{j=1}^{N} \mathbf{A}_{ij} = 1$. Formally, getting matrix $\mathbf{A}$ needs to consider inner product among all elements in query and key, which leads to quadratic computational cost. Meanwhile, considering the inherent 2D structure of the image, the applications of attention in vision are limited to big patch size or low resolution images.

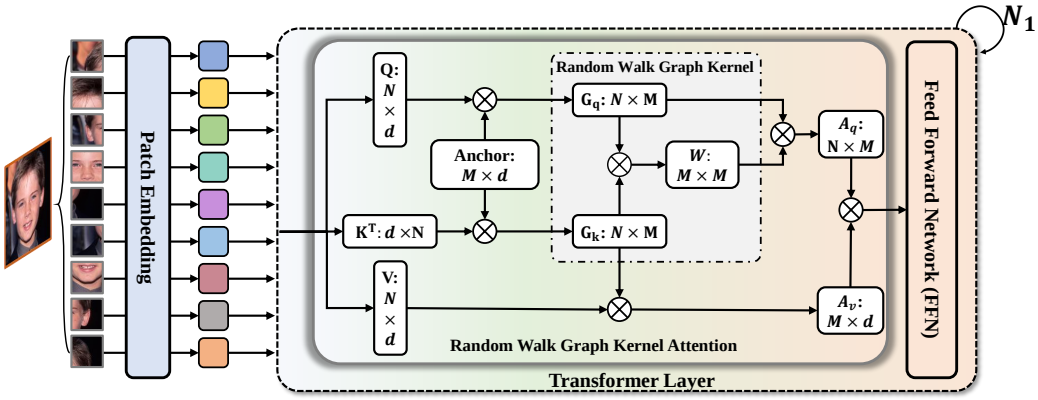

Figure 2: **Framework** of our proposed random walk graph kernel attention. $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ denotes the query, key and value of the input, respectively. Anchors with query and key form a bipartite graph structure. The gray dashed box contains the core part of the random walk graph kernel method.

3.2 RANDOM WALK GRAPH KERNEL AS ATTENTION

The key idea of our work is to handle long sequence input in vision tasks with an enhanced linear attention mechanism using random walk graph kernel (or Katz kernel (Katz, 1953)), which is a classical kernel formed by graph paths of different lengths.

Enhance Linear Attention with random walk graph kernel. Shen et al. (2021) proposed a simple linear attention $\mathbf{A} = \mathrm{Softmax}(\mathbf{Q})\,\mathrm{Softmax}(\mathbf{K})$ to deal with long sequence input in vision tasks. In order to maintain speed and feasible memory usage, this kind of linear attention sacrifices its representing capacity. We noticed that the simple linear attention only modeling one-step interaction between input tokens. However, for image-related problem, the representation of visual tokens in the embedding space is characterized by semantic coupling. To elaborate, when we split an image into small patches, the token embedding of a image patch may only has low-level semantic information. As shown in Fig. 1, given a weak adjacent matrix $\mathbf{A}$, considering hierarchical relative interactions based on $\mathbf{A}$ can clearly strengthen itself. Namely, we can get a better attention mechanism by calculating the weighted average of the powers of the matrix $\mathbf{A}$ of different orders:

$$\lambda_1\mathbf{A} + \lambda_2\mathbf{A}^2 + \cdots + \lambda_n\mathbf{A}^n + \ldots, s.t. \sum_{n=1}^{\infty}\lambda_n = 1, \lambda_i \in [0,1]. \tag{3}$$

Among Eq. (3), $\mathbf{A}^n$ denotes n-length path adjacent matrix of graph $\mathcal{G}$. We would expect the relevance of longer paths to decay, thus, Eq. (3) has the same form as a random walk graph kernel:

$$\kappa(\mathbf{A}) = \sum_{n=1}^{\infty}\lambda^n\mathbf{A}^n = \sum_{n=0}^{\infty}\lambda^n\mathbf{A}^n - \mathbf{I} = (\mathbf{I} - \lambda\mathbf{A})^{-1} - \mathbf{I}. \tag{4}$$

Now we can construct a general attention mechanism using random walk graph kernel with a normalised parameter:

$$\mathrm{KernelAttention}(\mathbf{A}) = \frac{1-\lambda}{\lambda}\kappa(\mathbf{A}) = \frac{1-\lambda}{\lambda}((\mathbf{I} - \lambda\mathbf{A})^{-1} - \mathbf{I}). \tag{5}$$

Flexible Bipartite Anchor Graph. We briefly discuss how to construct an appropriate bipartite graph with a set of flexible anchors which can both improve the capacity of inner attention matrix $\mathbf{A}$ and make the random walk graph kernel attention more efficient. As shown in Fig. 2, given query $\mathbf{Q} \in \mathbb{R}^{N\times d}$ and key $K \in \mathbb{R}^{N\times d}$, we represent them with some base anchors in an embedding space with also dimension d, denoted as $\mathbf{B}_Q \in \mathbb{R}^{M\times d}$ and $\mathbf{B}_K \in \mathbb{R}^{M\times d}$:

$$\mathbf{G}_Q = \mathrm{Softmax}(\mathbf{Q}\mathbf{B}_K^T), \mathbf{G}_K = \mathrm{Softmax}(\mathbf{B}_Q\mathbf{K}^T). \tag{6}$$

In the actual implementation, there is no special restriction on the choice of anchor points. Those anchor points can either be obtained from query and key through a projection, or can simply be designed as a set of learnable parameters. Similar to simple linear attention, we let

$$\mathbf{A} = \mathbf{G}_Q\mathbf{G}_K = \mathrm{Softmax}(\mathbf{Q}\mathbf{B}_K^T)\,\mathrm{Softmax}(\mathbf{B}_Q\mathbf{K}^T). \tag{7}$$

Linearization by Woodbury Formula. The current random walk graph kernel attention is still suffering unacceptable computational cost from $\mathbf{A}$ and the matrix inversion in Eq. (5). Fortunately, combined with our bipartite anchor graph design, the Woodbury formula provide us a technique transposing Eq. (5) to:

$$\mathrm{RWKernelAttention}(\mathbf{A}) = (1-\lambda)\mathbf{G}_Q(\mathbf{I} - \lambda\mathbf{G}_K\mathbf{G}_Q)^{-1}\mathbf{G}_K. \tag{8}$$

The complete architecture is shown in Fig. 2 and the implementation pipeline in practice is shown in algorithm 1.

Algorithm 1: Pipeline for random walk graph kernel Attention

Input: Current feature sequence $\mathbf{X}$, parameter matrix $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W_V}, \mathbf{B}_Q, \mathbf{B}_K$
Result: Output feature sequence $\mathrm{RWKernelAttention}(\mathbf{X})$.

1 Compute query, key and value: $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, K = \mathbf{X}\mathbf{W}_K, \mathbf{V} = \mathbf{X}\mathbf{W}_V$;
2 Compute bipartite graph matrix: $\mathbf{G}_Q = \mathrm{Softmax}(\mathbf{Q}\mathbf{B}_K^T), \mathbf{G}_K = \mathrm{Softmax}(\mathbf{B}_Q\mathbf{K}^T)$;
3 Compute kernel matrix $\hat{\mathbf{W}} = \mathbf{G}_Q\mathbf{G}_K$ and the inversion: $(\mathbf{I} - \lambda\hat{\mathbf{W}})^{-1}$;
4 Normalise and Use associative law of matrix multiplication:

$$\mathrm{RWKernelAttention}(\mathbf{X}) = (1-\lambda)(\mathbf{G}_Q(\mathbf{I} - \lambda\mathbf{G}_K\mathbf{G}_Q)^{-1})(\mathbf{G}_K\mathbf{V})$$

3.3 Analysis of Random Walk Graph Kernel Attention.

Softmax Plays a Key Role. The following simple property(the proof is given in appendix) allows us give a simple analysis on why we choose softmax similarity in our bipartite graph structure.

Lemma 1 *We call a matrix $\mathbf{A} \in \mathbb{R}^{N\times d}$ is normalised if and only if the summation of all values in each row of the matrix is equal to 1, i.e. for any $i \in N, \sum_{j=1}^{d}\mathbf{A}_{ij} = 1$. Then we have the multiplication of any number normalized matrix (feasibility without violating matrix multiplication): $\mathbf{A}_1\mathbf{A}_2\ldots\mathbf{A}_n$ is still a normalised matrix.*

Keeping $(\mathbf{I} - \lambda\hat{\mathbf{W}})$ is invertible, similar to Eq. (3) we can expand it as:

$$(\mathbf{I} - \lambda\hat{\mathbf{W}})^{-1} = \mathbf{I} + \lambda\hat{\mathbf{W}} + \lambda^2\hat{\mathbf{W}}^2 + \lambda^3\hat{\mathbf{W}}^3 + \ldots \tag{9}$$

From lemma 1, we can easily get the summation of each row of the kernel is equal to $\sum_{n=0}^{\infty}\lambda^n = \frac{1}{1-\lambda}$. Thus, with the normalization factor $\frac{1-\lambda}{\lambda}$, the matrix get from our random walk graph kernel attention, the right hand side of Eq. (8) is a normalized matrix. As discussed in section 3.1, our random walk graph kernel attention forms a general attention mechanism. Moreover, the kernel matrix $\hat{\mathbf{W}}$ is a normalized matrix, means that $\rho(\hat{\mathbf{W}}) \leq 1$ ($\rho(\hat{\mathbf{W}})$ denotes the the spectral radius of $\hat{\mathbf{W}}$). Then we can choose the hyperparameter $\lambda \in (0, 1)$, which ensures the matrix invertible and the training stability.

Complexity analysis. We now provide a computational complexity analysis of our random walk graph kernel attention, which comprise the computation of bipartite graphs, random walk graph kernel and the final matrix multiplication. Two bipartite graphs respectively calculate $\mathbf{G}_Q = \mathrm{Softmax}(\mathbf{Q}\mathbf{B}_K^T)$ and $\mathbf{G}_K = \mathrm{Softmax}(\mathbf{B}_Q\mathbf{K}^T)$ takes $O(NMd + MNd)$. The kernel matrix $\mathbf{W} = \mathbf{G}_K\mathbf{G}_Q$ and the inversion among kernel calculate $(I - \lambda\mathbf{W})^{-1}$ takes $O(M^2N + M^3)$. The final matrix multiplication calculates $(\mathbf{G}_Q(I - \lambda\mathbf{W})^{-1})(\mathbf{G}_K\mathbf{V})$ takes $O(M^2N + MNd + NMd)$. And in terms of memory usage, our random walk graph kernel attention takes $O(Md)$ of anchor matrix, $O(NM + NM)$ of two bipartite graph matrix and M^2 for kernel matrix. The overall computational cost is thus $O(4NMd + 2M^2N + M^3)$ and the overall memory usage is thus

Table 1: Comparison to state-of-the-art real time methods on Cityscapes

Method	Encoder	#Param		Cityscapes		
		Encoder	Decoder	Image Size	GFlops	mIoU
FCN (Long et al., 2015)	MobileNetV2	9.8		1024×1024	317.1	61.5
ICNet (Zhao et al., 2018)	-	-		1024×1024	-	67.7
PSPNet (Zhao et al., 2017)	MobileNetV2	13.7		1024×1024	423.4	70.2
DeepLabV3 (Chen et al., 2018)	MobileNetV2	15.4		1024×1024	555.4	75.2
Segformer (Xie et al., 2021)	MiT-B0	3.4	0.4	1024×1024	44.3	**76.2**
		3.4	0.4	768 ×768	20.8	**75.3**
		3.4	0.4	640 ×1024	23.8	73.7
		3.4	0.4	512 ×1024	18.0	71.9
Segformer-RWGKA	MiT-B0-RWGKA	2.9	0.4	1024×1024	**33.4**	75.2
		2.9	0.4	768 ×768	**18.5**	75.2
		2.9	0.4	640 ×1024	**20.6**	**73.8**
		2.9	0.4	512× 1024	**16.4**	**72.7**
Segformer (Xie et al., 2021)	MiT-B1	13.1	0.6	1024×1024	86.8	**78.5**
	MiT-B2	24.2	3.3	1024×1024	291.0	81.0
Segformer-RWGKA	MiT-B1-RWKA	11.4	0.6	1024×1024	**64.9**	78.1
	MiT-B2-RWKA	20.2	3.3	1024×1024	**249.0**	**81.1**

$O(2NM + Md + M^2)$. Both computational and memory cost are dominated by the number of anchors, i.e. when $M \ll N$, our random walk graph kernel attention has linear complexity of the input length. In fact, during the actual implementation the anchor serves the purpose of giving the token embedding a new representation in a same dimensional space. Our experiments in next section shows that choosing the number of anchors equal to dimension of token embedding is enough. With multi-head technique adopted in most attention mechanism, for each head the input tokens are embedded in a 64-dimension space which is indeed significantly less than the input length.

4 Experiments

The structure of our model is not designed for a specific vision task. To verify the modeling ability and generalization of our model for long sequence inputs, we directly replace the attention block of existing well-known methods with our random walk graph kernel attention with anchors (RWGKA).

4.1 Segmentation

Dataset and Implementation Details. We performed the semantic segmentation experiments on cityscapes (Cordts et al., 2016) datasets. In the training phase, the backbone is initialized with the weights pre-trained on ImageNet, and the newly added layers are initialized with Xavier (Kumar, 2017). We optimize our models using AdamW (Loshchilov & Hutter, 2018) with an initial learning rate of 1e-4. The learning rate is decayed following the polynomial decay schedule with a power of 0.9. We change all attention blocks in Segformer (Xie et al., 2021) with three different model size. We pretrain our model on ImageNet1K with Deit bachbone (Touvron et al., 2021). We report our model's semantic segmentation performance using mean Intersection over Union (mIoU).

Results and Analysis. Since our random walk graph kernel attention only introduce some learnable anchors but without projection needed to reduce the length of key and value, all the encoder using random walk graph kernel attention has only about 85% parameters compared with the original architecture. The linear computational complexity in random walk graph kernel also leads to about 10-25% flops reduce. For largest model B2, Segformer with our random walk graph kernel attention gets best mIoU result.

Table 2: Different backbones with supervised pre-trained in ImageNet-1K.

Model	#Param(M)	Flops	top-1 (%)
SegFormer-B0	3.58	**0.46**	**70.50**
SegFormer-B0-RWGKA	**3.17**	0.65	69.58
SegFormer-B1	13.66	**1.69**	**78.70**
SegFormer-B1-RWGKA	**11.93**	2.06	77.94
SegFormer-B2	24.71	**3.29**	81.60
SegFormer-B2-RWGKA	**20.75**	4.00	**82.07**

Table 3: Reconstruction Quality based on VITVQGAN

Model	Size	Length	GFlops	Batch Size	Train Steps	FID (↓) on Val.
ViT-VQGAN-SS (Yu et al., 2021)	256	1024	4.03	256	500K	**4.66**
ViT-VQGAN-SS-Nystrom	256	1024	8.94	64	300K	11.97
ViT-VQGAN-SS-RWGKA	256	1024	**2.78**	64	300K	5.72
ViT-VQGAN-SS-Slinear	512	4096	**10.08**	64	300K	27.27
ViT-VQGAN-SS-Nystrom	512	4096	21.09	64	300K	10.00
ViT-VQGAN-SS-RWGKA	512	4096	11.09	64	300K	**6.70**
ViT-VQGAN-SS-Nystrom	1024	4096	21.09	32	100K	27.80
ViT-VQGAN-SS-RWGKA	1024	4096	**11.09**	32	100K	**22.30**

4.2 Image Quantization with VITVQGAN

Dataset and Implementation Details. We performed the image quantization experiments with the ViTVQGAN backbone on CelebA-HQ (Huang et al., 2018) dataset. We train a ViT-VQGAN-SS model, the base first stage model used for image synthesis in Yu et al. (2021), on three resolutions (*i.e.*, 256, 512 and 1024) separately. We chose linear attention mentioned in Shen et al. (2021); Xiong et al. (2021) as our comparison backbone, since their architecture is closed to ours. Finding that CelebA-HQ dataset is relative easy to train and limited by computing resources, we train all model 300K steps with 64 batch size on 256/512 resolution and we train all model 100K steps with 32 batch size on 1024 resolution. All other training settings are following Yu et al. (2021).

Results and Analysis. As shown in Tab. 3, the quantitative results indicate that our method on image size 256^2 get the comparable performance with traditional self-attention mechanism while get fewer computing flops. When dealing with high resolution image quantization and reconstruction tasks, our method gets significantly better performance and keeps almost identical computing flops with the simplest softmax linear attention.

We then show a visualization results in Fig. 3 to compare the relationship between model capabilities and multi-step attention considerations for image tokens. In this experiments we choose the simplest softmax linear attention $\mathbf{A}_{1-Slinear} = \mathbf{Q}\,\mathbf{K}^{\mathrm{T}}$ as a 1-step concerned attention method. On this basis, we compared it with 2-steps concerned method $\mathbf{A}_{2-Slinear}$ and ∞-step concerned method $\mathbf{A}_{\infty-Slinear}$. We trained a VIT-VQGAN-SS model with the above three kind of attention mechanism on Celeb-AHQ with 64 batch size and 100 epochs. In Fig. 3, we found that the attention who concerned more steps relationships between the input tokens gives out a stronger model capacibility.

4.3 Image Classification

Dataset and Implementation Details. We performed the Image classification experiments on the ImageNet-1K dataset (Deng et al., 2009), which consists of 1.3M training images and 50K validation images from 1,000 categories. We use CvT (Wu et al., 2021) and ViL (Zhu et al., 2021), the state-of-the-art vision transformer architectures, as our backbones and replace their attention mechanisms with our random walk graph kernel attention equipped by pooling tokens (RWGKP) or

Table 4: Test accuracies of models trained on ImageNet-1K.

Model	#Param (M)	Image Size	FLOPs (G)	ImageNet top-1 (%)	ImageNet top-5 (%)
DeiT-S (Touvron et al., 2021)	22	224^2	4.6	79.8	95.0
DeiT-B (Touvron et al., 2021)	86	224^2	17.6	81.8	95.6
PVT-Medium (Wang et al., 2021)	44	224^2	6.7	81.2	-
PVT-Large (Wang et al., 2021)	61	224^2	9.8	81.7	-
Swin-S (Liu et al., 2021)	50	224^2	8.7	83.2	96.2
Swin-B (Liu et al., 2021)	88	224^2	15.4	83.5	96.5
PVTv2-B4 (Wang et al., 2022)	63	224^2	10.1	83.6	-
PVTv2-B5 (Wang et al., 2022)	82	224^2	11.8	83.8	-
ViT-B/16 (Dosovitskiy et al., 2021)	86	384^2	55.5	77.9	-
ViT-L/16 (Dosovitskiy et al., 2021)	307	384^2	191.1	76.5	-
DeiT-B (Touvron et al., 2021)	86	384^2	55.5	83.1	96.2
Swin-B (Liu et al., 2021)	88	384^2	47.1	84.5	97.0
CvT-13 (Wu et al., 2021)	20	224^2	4.6	81.6	95.7
CvT-13-RWGKP	20	224^2	**4.4**	81.3	95.6
CvT-13-RWGKA	20	224^2	4.5	81.3	94.2
CvT-13 (Wu et al., 2021)	20	384^2	16.3	83.0	96.4
CvT-13-RWGKP	20	384^2	**13.5**	82.5	96.2
CvT-13-RWGKA	20	384^2	13.6	82.7	96.4
CvT-21 (Wu et al., 2021)	31.6	384^2	25.0	83.3	96.2
CvT-21-RWGKP	31.6	384^2	**21.8**	82.9	96.0
ViL-Tiny (Zhu et al., 2021)	6.4	224^2	**1.3**	**76.3**	-
ViL-Tiny-RWGKP	6.4	224^2	**1.3**	76.0	93.2
ViL-Medium (Zhu et al., 2021)	39.8	224^2	11.0	**83.3**	-
ViL-Medium-RWGKP	39.8	224^2	**10.7**	**83.3**	96.4

anchors (RWGKA), denoted as CvT-RWGKP, CvT-RWGKA and ViL-RWGKP in the Tab. 4. For All models are trained for 300 epochs. We apply a center crop on the validation set to benchmark, where we adopt 224×224 and 384×384 resolution to evaluate the classification accuracy.

Results and Analysis. To demonstrate the generalization and capabilities dealing with long sequences of our models, we compare our models with two notable method (*i.e.*, CvT and ViL) on the classification tasks. CvT models a long sequence length in the early stages (*e.g.*, $96 \times 96 = 9216$ patches for images with 384^2 image size). As shown in the Tab. 4, our random walk graph kernel attention with CvT-21 on 384^2 image size achieves competitive results with CvT, while using the same amount of parameters and 87.2% of its FLOPs. ViL uses window attention and global tokens to improve the efficiency. Our ViL-Medium-GKP with random walk graph attention saves 0.3 FLOPs compared with the vanilla ViL-Medium, while maintaining the same perfomance. Thanks to our designed the random walk graph kernel, our method has shown superior cost-benefit trade-off over these two approaches.

4.4 Ablation Study

Thus We do some ablation studies on the long path decay factor λ and the number of anchors in semantic segmentation with the backbone Segformer-B2 whose attention layers are changed by ours. We compare the mIoU accuracy on Cityscapes dataset with resolution 1024×1024.

The impact of λ in random walk graph kernel attention. As results shown in Tab. 5, we see that in semantic segmentation the model performance drops with the long path decay factor λ increasing. This is not counter-intuitive; in fact, as λ gets larger, the random walk graph kernel attention mechanism gives more weight to modeling relative relationships over more steps. This

Table 5: **Ablation studies** on Segformer

Model	#Heads	#Anchors	λ	#Param	GFlops	mIoU
SegFormer-B2-RWGKA	1×	64	0.5	23.5	249	80.96
	1×	64	0.1	23.5	249	**81.10**
	1×	64	0.9	23.5	249	80.89
	1×	32	0.5	23.2	243	81.01
	1×	128	0.5	23.7	268	80.77
	2×	32	0.5	23.2	242	80.62
	2×	64	0.5	23.4	251	81.03

Figure 3: 256×256 image reconstruction comparison of 1-step, 2-step weak linear attention and random walk graph kernel attention.

means that the model focuses more on deep cross-token interactions, which may not be necessary or even harmful for semantic segmentation task.

The impact of anchor size and the number of head. In the last four rows in Tab. 5, we show the relationship between the model performance and the number of anchor points. As we mentioned in Sec. 3.3, Keeping the number of anchors in the same dimension as the token embedding gives good performance. Increasing the number of anchors does not improve the model performance significantly, but even makes it more difficult to learn and makes it slower. In addition, we can improve the performance of the model by increasing the number of attention heads without significantly increasing the computational consumption. This technique also allows us to control the number of anchor points ≤ 64. In practical applications, this makes the number of anchor points much smaller than the length of the input sequence, which guarantees the linear complexity of our proposed method in terms of computation and storage.

5 Conclusion

Due to the inherent 2D structure of the image itself, the vision transformer needs to simplify the traditional self-attention when dealing with long sequence input. Previous methods based on sparse attention tend to have insufficient generalization, while those based on linear attention tend to have limited performance. We approach a novel attention mechanism based on random walk graph kernel, can be widely used in vision transformer with long sequence inputs. The random walk graph kernel enhances a weak linear attention by pushing it focus on multi-step interactions between tokens. By a special design of a bipartite anchor graph, we make the random walk graph kernel attention still maintain the structure of linear attention, i.e. the linear complexity is maintained. This plug-and-play attention mechanism achieves competitive performance with significant speed up on various vision tasks, especially for settings with long sequences as input.

REFERENCES

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. Etc: Encoding long and structured inputs in transformers. In *Conference on Empirical Methods in Natural Language Processing*, pp. 268–284, 2020.

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

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Eur. Conf. Comput. Vis.*, 2018.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *Int. Conf. Learn. Represent.*, 2021.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Huaibo Huang, Ran He, Zhenan Sun, Tieniu Tan, et al. Introvae: Introspective variational autoencoders for photographic image synthesis. *Adv. Neural Inform. Process. Syst.*, 31, 2018.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Int. Conf. Mach. Learn.*, 2020.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *Int. Conf. Learn. Represent.*, 2020.

Siddharth Krishna Kumar. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017.

Wei Liu, Junfeng He, and Shih-Fu Chang. Large graph construction for scalable semi-supervised learning. In *Int. Conf. Mach. Learn.*, 2010.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, 2021.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent.*, 2018.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *Int. Conf. Mach. Learn.*, pp. 4055–4064, 2018.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *Int. Conf. Learn. Represent.*, 2021.

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *IEEE Winter Conf. Appl. Comput. Vis.*, pp. 3531–3539, 2021.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention in transformer models. In *Int. Conf. Mach. Learn.*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In *Int. Conf. Mach. Learn.*, volume 139, pp. 10347–10357, July 2021.

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

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inform. Process. Syst.*, 34:12077–12090, 2021.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. *Assoc. Adv. Artif. Intell.*, 2021.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. In *Int. Conf. Learn. Represent.*, 2021.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

Manzil Zaheer, Guru Guruganesh, 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 *Adv. Neural Inform. Process. Syst.*, 2020.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Eur. Conf. Comput. Vis.*, 2018.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 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. *Adv. Neural Inform. Process. Syst.*, 34:17723–17736, 2021.