

KVT: k -NN Attention for Boosting Vision Transformers

Pichao Wang*, Xue Wang*, Fan Wang, Ming Lin, Shuning Chang, Hao Li, and
Rong Jin

Alibaba Group

{pichao.wang, xue.w, fan.w, ming.l, shuning.csn, lihao.lh,
jinrong.jr}@alibaba-inc.com

Abstract. Convolutional Neural Networks (CNNs) have dominated computer vision for years, due to its ability in capturing locality and translation invariance. Recently, many vision transformer architectures have been proposed and they show promising performance. A key component in vision transformers is the fully-connected self-attention which is more powerful than CNNs in modelling long range dependencies. However, since the current dense self-attention uses all image patches (tokens) to compute attention matrix, it may neglect locality of images patches and involve noisy tokens (e.g., clutter background and occlusion), leading to a slow training process and potential degradation of performance. To address these problems, we propose the k -NN attention for boosting vision transformers. Specifically, instead of involving all the tokens for attention matrix calculation, we only select the top- k similar tokens from the keys for each query to compute the attention map. The proposed k -NN attention naturally inherits the local bias of CNNs without introducing convolutional operations, as nearby tokens tend to be more similar than others. In addition, the k -NN attention allows for the exploration of long range correlation and at the same time filters out irrelevant tokens by choosing the most similar tokens from the entire image. Despite its simplicity, we verify, both theoretically and empirically, that k -NN attention is powerful in speeding up training and distilling noise from input tokens. Extensive experiments are conducted by using 11 different vision transformer architectures to verify that the proposed k -NN attention can work with any existing transformer architectures to improve its prediction performance. The codes are available at <https://github.com/damo-cv/KVT>.

1 Introduction

Traditional CNNs provide state of the art performance in vision tasks, due to its ability in capturing locality and translation invariance, while transformer [53] is the de-facto standard for natural language processing (NLP) tasks thanks to its advantages in modelling long-range dependencies. Recently, various vision

* The first two authors contribute equally.

transformers [16,51,70,55,24,69,41,59,52,71] have been proposed by building pure or hybrid transformer models for visual tasks. Inspired by the transformer scaling success in NLP tasks, vision transformer converts an image into a sequence of image patches (tokens), with each patch encoded into a vector. Since self-attention in the transformer is position agnostic, different positional encoding methods [16,11,14] have been developed, and in [59,8] their roles have been replaced by convolutions. Afterwards, all tokens are fed into stacked transformer encoders for feature learning, with an extra *CLS* token [16,51,14] or global average pooling (GAP) [41,8] for final feature representation. Compared with CNNs, transformer-based models explicitly exploit global dependencies and demonstrate comparable, sometimes even better, results than highly optimised CNNs [26,47].

Albeit achieving its initial success, vision transformers suffer from slow training. One of the key culprits is the fully-connected self-attention, which takes all the tokens to calculate the attention map. The dense attention not only neglects the locality of images patches, an important feature of CNNs, but also involves noisy tokens into the computation of self-attention, especially in the situations of cluttered background and occlusion. Both issues can slow down the training significantly [12,14]. Recent works [69,59,8] try to mitigate this problem by introducing convolutional operators into vision transformers. Despite encouraging results, these studies fail to resolve the problem fundamentally from the transformer structure itself, limiting their success. In this study, we address the challenge by directly attacking its root cause, i.e. the fully-connected self-attention.

To this end, we propose the k -NN attention to replace the fully-connected attention. Specifically, we do not use all the tokens for attention matrix calculation, but only select the top- k similar tokens from the sequence for each query token to compute the attention map. The proposed k -NN attention not only naturally inherits the local bias of CNNs as the nearby tokens tend to be more similar than others, but also builds the long range dependency by choosing the most similar tokens from the entire image. Compared with convolution operator which is an aggregation operation built on Ising model [43] and the feature of each node is aggregated from nearby pixels, in the k -NN attention, the aggregation graph is no longer limited by the spatial location of nodes but is adaptively computed via attention maps, thus, the k -NN attention can be regarded as a relieved version of local bias. The similar idea is proposed in [75] where the k -NN attention is mostly evaluated on NLP tasks. Despite the similarity in terms of the calculation of top- k , our work focuses on the recent vision transformers, makes a deep theoretical understanding and presents a thoroughly analysis by defining several metrics. We verify, both theoretically and empirically, that k -NN attention is effective in speeding up training and distilling noisy tokens of vision transformers. Eleven different available vision transformer architectures are adopted to verify the effectiveness of the proposed k -NN attention.

2 Related Work

2.1 Self-attention

Self-attention [53] has demonstrated promising results on NLP related tasks, and is making breakthroughs in speech and computer vision. For time series modeling, self-attention operates over sequences in a step-wise manner. Specifically, at every time-step, self-attention assigns an attention weight to each previous input element and uses these weights to compute the representation of the current time-step as a weighted sum of the past inputs. Besides the vanilla self-attention, many efficient transformers [50] have been proposed. Among these efficient transformers, sparse attention and local attention are one of the main streams, which are highly related to our work. Sparse attention can be further categorized into data independent (fixed) sparse attention [9,29,1,72] and content-based sparse attention [13,45,34,48]. Local attention [42,40,41] mainly considers attending only to a local window size. Our work is also content-based attention, but compared with previous works [13,45,34,48], our k -NN attention has its merits for vision domain. For example, compared with routing transformer [45] that clusters both queries and keys, our k -NN attention equals only clustering keys by assigning each query as the cluster center, making the quantization more continuous which is a better fitting of image domain; compared with reformer [34] which adopts complex hashing attention that cannot guarantee each bucket contain both queries and keys, our k -NN attention can guarantee that each query has number k keys for attention computing. In addition, our k -NN attention is also a generalized local attention, but compared with local attention, our k -NN attention not only enjoys the locality but also empowers the ability of global relation mining.

2.2 Transformer for Vision

Transformer [53] is an effective sequence-to-sequence modeling network, and it has achieved state-of-the-art results in NLP tasks with the success of BERT [15]. Due to its great success, it has also be exploited in computer vision community, and ‘Transformer in CNN’ becomes a popular paradigm [2,58,7,80,36,37,25,4]. ViT [16] leads the other trend to use ‘CNN in Transformer’ paradigm for vision tasks [27,35,68,62,66]. Even though ViT has been proved compelling in vision recognition, it has several drawbacks when compared with CNNs: large training data, fixed position embedding, rigid patch division, coarse modeling of inner patch feature, single scale, unstable training process, slow speed training, easily fitting data and poor generalization, shallow & narrow architecture, and quadratic complexity. To deal with these problems, many variants have been proposed [74,32,57,19,30,20,44,67,78,17,56,64,79]. For example, DeiT [51] adopts several training techniques and uses distillation to extend ViT to a data-efficient version; CPVT [11] proposes a conditional positional encoding that is adaptable to arbitrary input sizes; CvT [59], CoaT [63] and Visformer [8] safely remove the position embedding by introducing convolution operations; T2T ViT [70], CeiT [69], and CvT [59] try to deal with the rigid patch division by introducing

convolution operation for patch sequence generation; Focal Transformer [65] makes each token attend its closest surrounding tokens at fine granularity and the tokens far away at coarse granularity; TNT [24] proposes the pixel embedding to model the inner patch feature; PVT [55], Swin Transformer [41], MViT [18], ViL [73], CvT [59], PiT [28], LeViT [22], CoaT [63], and Twins [10] adopt multi-scale technique for rich feature learning; DeepViT [77], CaiT [52], and PatchViT [21] investigate the unstable training problem, and propose the re-attention, re-scale and anti-over-smoothing techniques respectively for stable training; to accelerate the convergence of training, ConViT [14], PiT [28], CeiT [69], LocalViT [38] and Visformer [8] introduce convolutional bias to speedup the training; *convstem* is adopted in LeViT [22], EarlyConv [60], CMT [23], VOLO [71] and ScaledReLU [54] to improve the robustness of training ViTs; LV-ViT [31] adopts several techniques including MixToken and Token Labeling for better training and feature generation; T2T ViT [70], DeepViT [77] and CaiT [52] try to train deeper vision transformer models; T2T ViT [70], ViL [73] and CoaT [63] adopt efficient transformers [50] to deal with the quadratic complexity; To further exploit the capacities of vision transformer, OmniNet [49], CrossViT [6] and SoViT [61] propose the dense omnidirectional representations, coarse-fine-grained patch fusion and cross co-variance pooling of visual tokens, respectively. However, all of these works adopt the fully-connected self-attention which will bring the noise or irrelevant tokens for computing and slow down the training of networks. In this paper, we propose an efficient sparse attention, called k -NN attention, for boosting vision transformers. The proposed k -NN attention not only inherits the local bias of CNNs but also achieves the ability of global feature exploitation. It can also speed up the training and achieve better performance.

3 k -NN Attention

3.1 Vanilla Attention

For any sequence of length n , the vanilla attention in the transformer is the dot product attention [53]. Following the standard notation, the attention matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is defined as:

$$\mathbf{A} = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right),$$

where $\mathbf{Q} \in \mathbb{R}^{n \times d}$ denotes the queries while $\mathbf{K} \in \mathbb{R}^{n \times d}$ denotes the keys, and d represents the dimension. By multiplying the attention weights \mathbf{A} with the values $\mathbf{V} \in \mathbb{R}^{n \times d}$, the new values $\hat{\mathbf{V}}$ are calculated as:

$$\hat{\mathbf{V}} = \mathbf{A}\mathbf{V}.$$

The intuitive understanding of the attention is the weighted average over the old ones, where the weights are defined by the attention matrix \mathbf{A} . In this paper, we consider the \mathbf{Q} , \mathbf{K} and \mathbf{V} are generated via the linear projection of the input token matrix \mathbf{X} :

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

where $\mathbf{X} \in \mathbb{R}^{n \times d_m}$, $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d_m \times d}$ and d_m is the input token dimension.

One shortcoming with fully-connected self-attention is that irrelevant tokens, even though assigned with smaller weights, are still taken into consideration when updating the representation \mathbf{V} , making it less resilient to noises in \mathbf{V} . This shortcoming motivates us to develop the k -NN attention.

3.2 k -NN Attention

Instead of computing the attention matrix for all the query-key pairs as in vanilla attention, we select the top- k most similar keys and values for each query in the k -NN attention. There are two versions of k -NN attention, as described below.

Slow Version: For the i -th query, we first compute the Euclidean distance against all the keys, and then obtain its k -nearest neighbors \mathcal{N}_i^k and \mathcal{N}_i^v from keys and values, and lastly calculate the scaled dot product attention as:

$$\mathbf{A}_i = \text{softmax} \left(\frac{\langle \mathbf{q}_i, (\mathbf{k}_{j_1}, \dots, \mathbf{k}_{j_l}, \dots, \mathbf{k}_{j_k}) \rangle}{\sqrt{d}} \right), \mathbf{k}_{j_l} \in \mathcal{N}_i^k.$$

The shape of final attention matrix is $\mathbf{A}^{knn} \in \mathbb{R}^{n \times k}$, and the new values $\hat{\mathbf{V}}^{knn}$ is the same size of values $\hat{\mathbf{V}}$. The slow version is the exact definition of k -NN attention, but it is extremely slow because for each query it has to compute distances for different k keys.

Fast Version: As the computation of Euclidean distance against all the keys for each query is slow, we propose a fast version of k -NN attention. The key idea is to take advantage of matrix multiplication operations. Same as vanilla attention, all the queries and keys are calculated by the dot product, and then row-wise top- k elements are selected for softmax computing. The procedure can be formulated as:

$$\hat{\mathbf{V}}^{knn} = \text{softmax} \left(\mathcal{T}_k \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right) \right) \cdot \mathbf{V},$$

where $\mathcal{T}_k(\cdot)$ denotes the row-wise top- k selection operator:

$$[\mathcal{T}_k(\mathbf{A})]_{ij} = \begin{cases} A_{ij} & A_{ij} \in \text{top-}k(\text{row } j) \\ -\infty & \text{otherwise.} \end{cases}$$

3.3 Theoretical Analysis on k -NN Attention

In this section, we will show theoretically that despite its simplicity, k -NN attention is powerful in speeding up network training and in distilling noisy tokens. All the proof of the lemmas are provided in the supplementary.

Convergence Speed-up. Compared to CNNs, the fully-connected self-attention is able to capture long range dependency. However, the price to pay is that the dense self-attention model requires to mix each image patch with every other patch in the image, which has potential to mix irrelevant information together, e.g. the foreground patches may be mixed with background patches through the self-attention. This defect could significantly slow down the convergence as the goal of visual object recognition is to identify key visual patches relevant to a given class.

To see this, we consider the model with only learnable parameters $\mathbf{W}_Q, \mathbf{W}_K$ in attention layers and adopting Adam optimizer [33]. According to Theorem 4.1 in [33], Adam’s convergence is proportional to $\mathcal{O}(\alpha^{-1}(G_\infty + 1) + \alpha G_\infty)$, where α is the learning rate and G_∞ is an element-wise upper bound on the magnitude of the batch gradient¹. Let f_i be the loss function corresponding to batch i . Via chain rule of derivative, the gradient w.r.t the \mathbf{W}_Q in a self-attention block can be represented as $\nabla_{\mathbf{W}_Q} f_i = F_i(\hat{\mathbf{V}}^{knn}) \cdot \frac{\partial \hat{\mathbf{V}}^{knn}}{\partial \mathbf{W}_Q}$, where $F_i(\hat{\mathbf{V}}^{knn})$ is a matrix output function. Since the possible value of $\hat{\mathbf{V}}^{knn}$ is a subset of its fully-connected counterpart, the upper bound of on the magnitude of $F_i(\hat{\mathbf{V}}^{knn})$ is no larger than the full attention. We then introduce the weighed covariate matrix of patches to characterize the scale of $\frac{\partial \hat{\mathbf{V}}^{knn}}{\partial \mathbf{W}_Q}$ in the following lemma.

Lemma 1. (Informal) Let $\hat{\mathbf{V}}_l^{knn}$ be the l -th row of the $\hat{\mathbf{V}}^{knn}$. We have

$$\frac{\partial \hat{\mathbf{V}}_l^{knn}}{\partial \mathbf{W}_Q} \propto \text{Var}_{\mathbf{a}_l}(\mathbf{x}) \text{ and } \frac{\partial \hat{\mathbf{V}}_l^{knn}}{\partial \mathbf{W}_K} \propto \text{Var}_{\mathbf{a}_l}(\mathbf{x}),$$

where $\text{Var}_{\mathbf{a}_l}(\mathbf{x})$ is the covariate matrix on patches $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with probability from l -th row of the attention matrix.

The same is true for $\hat{\mathbf{V}}$ of the fully-connected self-attention.

Since k -NN attention only uses patches with large similarity, its $\text{Var}_{\mathbf{a}_l}(\mathbf{x})$ will be smaller than that computed from the fully-connected attention. As indicated in Lemma 1, $\frac{\partial \hat{\mathbf{V}}^{knn}}{\partial \mathbf{W}_Q}$ is proportional to variance $\text{Var}_{\mathbf{a}_l}(\mathbf{x})$ and thus the scale of $\nabla_{\mathbf{W}_Q} f_i$ becomes smaller in k -NN attention. Similarly, the scale of $\nabla_{\mathbf{W}_K} f_i$ is also smaller in k -NN attention. Therefore, the element-wise upper bound on batch gradient G_∞ in Adam analysis is also smaller for k -NN attention. For the same learning rate, the k -NN attention yields faster convergence. It is particularly significant at the beginning of training. This is because, due to the random initialization, we expect a relatively small difference in similarities between patches, which essentially makes self-attention behave like “global average”. It will take multiple iterations for Adam to turn the “global average” into the real function of self-attention. In Table 2 and Figure 2, we numerically verify the training efficiency of k -NN attention as opposed to the fully-connected attention.

¹ Theorem 4.1 in [33] describes the upper bound for regrets (the gap on loss function value between the current step parameters and optimal parameters). One can telescope it to the average regrets to consider the Adam’s convergence.

Noisy patch distillation. As already mentioned before, the fully-connected self-attention model may mix irrelevant patches with relevant ones, particularly at the beginning of training when similarities between relevant patches are not significantly larger than those for irrelevant patches. k -NN attention is more effective in identifying noisy patches by only considering the top k most similar patches. To formally justify this point, we consider a simple scenario where all the patches are divided into two groups, the group of relevant patches and the group of noisy patches. All the patches are sampled independently from unknown distributions. We assume that all relevant patches are sampled from distributions with the same shared mean, which is different from the means of distributions for noisy patches. It is important to know that although distributions for the relevant patches share the mean, those relevant patches can look quite differently, due to the large variance in stochastic sampling. In the following Lemma, we will show that the k -NN attention is more effective in distilling noises for the relevant patches than the fully-connected attention.

Lemma 2 (informal). *We consider the self-attention for query patch l . Let's assume the patch \mathbf{x}_i are bounded with mean $\boldsymbol{\mu}_i$ for $i = 1, 2, \dots, n$, and ρ_k is the ratio of the noisy patches in all selected patches. Under mild conditions, the follow inequality holds with high probability:*

$$\left\| \hat{\mathbf{V}}_l^{knn} - \boldsymbol{\mu}_l \mathbf{W}_V \right\|_{\infty} \leq \mathcal{O}(k^{-1/2} + c_1 \rho_k),$$

where c_1 is a positive number.

In the above lemma, the quantity $\left\| \hat{\mathbf{V}}_l^{knn} - \boldsymbol{\mu}_l \mathbf{W}_V \right\|_{\infty}$ measures the distance between $\hat{\mathbf{V}}_l^{knn}$, representation vector updated by the k -NN attention, and its mean $\boldsymbol{\mu}_l \mathbf{W}_V$. We now consider two cases: the normal k -NN attention with appropriately chosen k , and fully-connected attention with $k = n$. In the first case, with appropriately chosen k , we should have most of the selected patches coming from the relevant group, implying a small ρ_k . By combining with the fact that k is decently large, we expect a small upper bound for the distance $\left\| \hat{\mathbf{V}}_l^{knn} - \boldsymbol{\mu}_l \mathbf{W}_V \right\|_{\infty}$, indicating that k -NN attention is powerful in distilling noise. For the case of fully-connected attention model, i.e. $k = n$, it is clearly that $\rho_n \approx 1$, leading to a large distance between transformed representation $\hat{\mathbf{V}}_l$ and its mean, indicating that fully-connected attention model is not very effective in distilling noisy patches, particularly when noise is large.

Besides the instance with low signal-noise-ratio, the instance with a large volume of backgrounds can also be hard. In the next lemma, we show that under a proper choice of k , with a high probability the k -NN attention will be able to select all meaningful patches.

Lemma 3 (informal). *Let \mathcal{M}^* be the index set contains all patches relevant to query \mathbf{q}_l . Under mild conditions, there exist $c_2 \in (0, 1)$ such that with high probability, we have*

$$\sum_{i=1}^n \mathbb{1}(\mathbf{q}_i \mathbf{k}_i^\top \geq \min_{j \in \mathcal{M}^*} \mathbf{q}_i \mathbf{k}_j^\top) \leq \mathcal{O}(nd^{-c_2}).$$

The above lemma shows that if we select the top $\mathcal{O}(nd^{-c_2})$ elements, with high probability, we will be able to eliminate almost all the irrelevant noisy patches, without losing any relevant patches. Numerically, we verify the proper k gains better performance (e.g., Figure 1) and for the hard instance k -NN gives more accurate attention regions. (e.g., Figure 4 and Figure 5).

4 Experiments for Vision Transformers

In this section, we replace the dense attention with k -NN attention on the existing vision transformers for image classification to verify the effectiveness of the proposed method. The recent DeiT [51] and its variants, including T2T ViT [70], TNT [24], PiT [28], Swin [41], CvT [59], So-ViT [61], Visformer [8], Twins [10], Dino [3] and VOLO [71], are adopted for evaluation. These methods include both supervised methods [51,70,24,28,41,59,61,8,10,71] and self-supervised method [3]. Ablation studies are provided to further analyze the properties of k -NN attention.

4.1 Experimental Settings

We perform image classification on the standard ILSVRC-2012 ImageNet dataset [46]. In our experiments, we follow the experimental setting of original official released codes. For fair comparison, we only replace the vanilla attention with proposed k -NN attention. Unless otherwise specified, the fast version of k -NN attention is adopted for evaluation. To speed up the slow version, we develop the CUDA version k -NN attention. As for the value k , different architectures are assigned with different values. For DeiT [51], So-ViT [61], Dino [3], CvT [59], TNT [24] PiT [28] and VOLO [71], as they directly split an input image into rigid tokens and there is no information exchange in the token generation stage, we suppose the irrelevant tokens are easy to filter, and tend to assign a smaller k compared with these complicated token generation methods [70,41,8,10]. Specifically, we assign k to approximate $\frac{n}{2}$ at each scale stage; for these complicated token generation methods [70,41,8,10], we assign a larger k which is approximately $\frac{2}{3}n$ or $\frac{4}{5}n$ at each scale stage.

4.2 Results on ImageNet

Table 1 shows top-1 accuracy results on the ImageNet-1K validation set by replacing the dense attention with k -NN attention using eleven different vision transformer architectures. From the Table we can see that the proposed k -NN attention improves the performance from 0.2% to 0.8% for both global and local vision transformers. It is worth noting that on ImageNet-1k dataset, it is very hard to improve the accuracy after 85%, but our k -NN attention can still consistently improve the performance even without model size increase.

Table 1. The k -NN attention performance on ImageNet-1K validation set. "!" means we pretrain the model with 300 epochs and finetune the pretrained model for 100 epoch for linear eval, following the instructions of Dino training and evaluation; " $\rightarrow k$ -NN Attn" represents replacing the vanilla attention with proposed k -NN attention; $\rightarrow k$ -NN Attn-slow means adopting the slow version.

Arch.	Model	Input	Params	GFLOPs	Top-1
Transformers (Supervised)	DeiT-Tiny [51]	224^2	5.7M	1.3	72.2%
	DeiT-Tiny [51] $\rightarrow k$ -NN Attn	224^2	5.7M	1.3	73.0%
	DeiT-Tiny [51] $\rightarrow k$ -NN Attn-slow	224^2	5.7M	1.3	73.0%
	So-ViT-7 [61]	224^2	5.5M	1.3	76.2%
	So-ViT-7 [61] $\rightarrow k$ -NN Attn	224^2	5.5M	1.3	77.0%
Transformers (Supervised)	Visformer-Tiny [8]	224^2	10M	1.3	78.6%
	Visformer-Tiny [8] $\rightarrow k$ -NN Attn	224^2	10M	1.3	79.0%
Transformers (Supervised)	CvT-13 [59]	224^2	20M	4.6	81.6%
	CvT-13 [59] $\rightarrow k$ -NN Attn	224^2	20M	4.6	81.9%
	DeiT-Small [51]	224^2	22M	4.6	79.8%
	DeiT-Small [51] $\rightarrow k$ -NN Attn	224^2	22M	4.6	80.1%
	TNT-Small [24]	224^2	24M	5.2	81.5%
	TNT-Small [24] $\rightarrow k$ -NN Attn	224^2	24M	5.2	81.9%
	VOLO-D1 [71]	384^2	27M	22.8	85.2%
	VOLO-D1 [71] $\rightarrow k$ -NN Attn	384^2	27M	22.8	85.4%
	Swin-Tiny [41]	224^2	28M	4.5	81.2%
	Swin-Tiny [41] $\rightarrow k$ -NN Attn	224^2	28M	4.5	81.3%
	T2T-ViT-t-19 [70]	224^2	39M	9.8	82.2%
	T2T-ViT-t-19 [70] $\rightarrow k$ -NN Attn	224^2	39M	9.8	82.7%
Transformer (Self-supervised)	Dino-Small [3!]	224^2	22M	4.6	76.0%
	Dino-Small [3!] $\rightarrow k$ -NN Attn	224^2	22M	4.6	76.2%
Transformers (Supervised)	Twins-SVT-Base [10]	224^2	56M	8.3	83.2%
	Twins-SVT-Base [10] $\rightarrow k$ -NN Attn	224^2	56M	8.3	83.4%
	PiT-Base[28]	224^2	74M	12.5	82.0%
	PiT-Base[28] $\rightarrow k$ -NN Attn	224^2	74M	12.5	82.6%
	VOLO-D3 [71]	448^2	86M	67.9	86.3%
	VOLO-D3 [71] $\rightarrow k$ -NN Attn	448^2	86M	67.9	86.5%

4.3 The Impact of Number k

The only parameter for k -NN attention is k , and its impact is analyzed in Figure 1. As shown in the figure, for DeiT-Tiny, $k = 100$ is the best, where the total number of tokens $n = 196$ (14×14), meaning that k approximates half of n ; for CvT-13, there are three scale stages with the number of tokens $n_1 = 3136$, $n_2 = 784$ and $n_3 = 196$, and the best results are achieved when the k in each stage is assigned to 1600/400/100, which also approximate half of n in each stage; for Visformer-Tiny, there are two scale stages with the number of tokens $n_1 = 196$

and $n_2 = 49$, and the best results are achieved when k in each stage is assigned to 150/45, as there are more than 21 conv layers for token generation and the information in each token are already mixed, making it hard to distinguish the irrelevant tokens, thus larger values of k are desired; for PiT-Base, there are three scale stages with the number of tokens $n_1 = 961$, $n_2 = 256$ and $n_3 = 64$, and the optimal values of k also approximate the half of n . Please note that, we do not perform exhaustive search for the optimal choice of k , instead, a general rule as below is sufficient: $k \approx \frac{n}{2}$ at each scale stage for simple token generation methods and $k \approx \frac{2}{3}n$ or $\frac{4}{5}n$ for complicated token generation methods at each scale stage.

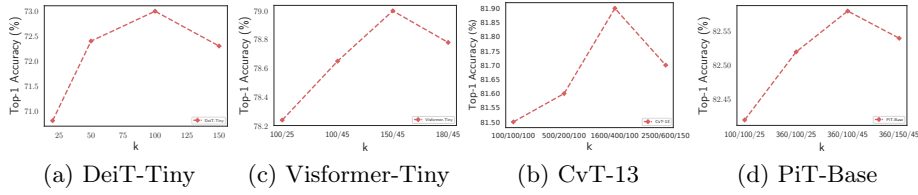


Fig. 1. The impact of k on DeiT-Tiny, Visformer-Tiny, CvT-13 and PiT-Base.

4.4 Convergence Speed of k -NN Attention

In Table 2, we investigate the convergence speed of k -NN attention. Three methods are included for comparison, i.e. DeiT-Small [51], CvT-13 [59] and T2T-ViT-t-19 [70]. From the Table we can see that the convergence speed of k -NN attention is faster than full-connected attention, especially in the early stage of training. These observations reflect that removing the irrelevant tokens benefits the convergence of neural networks training.

Table 2. Ablation study on the convergence speed of k -NN attention.

Epoch	Top-1 accuracy					
	DeiT-S	DeiT-S $\rightarrow k$	CvT-13	CvT-13 $\rightarrow k$	T2T-ViT-t-19	T2T-ViT-t-19 $\rightarrow k$
10	29.1%	31.3%	51.4%	54.2%	0.52%	0.68%
30	54.4%	55.4%	65.4%	68.1%	63.0%	63.2%
50	60.9%	62.0%	68.1%	70.5%	73.8%	74.4%
70	65.0%	65.8%	69.9%	72.2%	76.9%	77.3%
90	67.7%	68.2%	71.0%	73.0%	78.4%	78.6%
120	69.9%	70.7%	72.4%	73.7%	79.7%	80.0%
150	72.4%	72.4%	74.4%	74.9%	80.7%	80.9%
200	75.5%	75.7%	77.3%	77.7%	82.0%	82.3%
300	79.8%	80.0%	81.6%	81.9%	81.3%	81.7%

4.5 Other properties of k -NN attention

To analyze other properties of k -NN attention, four quantitative metrics are defined as follows.

Layer-wise cosine similarity between tokens: following [21] this metric is defined as:

$$\text{CosSim}(\mathbf{t}) = \frac{1}{n(n-1)} \sum_{i \neq j} \frac{t_i^T t_j}{\|t_i\| \|t_j\|},$$

where t_i represents the i -th token in each layer and $\|\cdot\|$ denotes the Euclidean norm. This metric implies the convergence speed of the network.

Layer-wise standard deviation of attention weights: Given a token t_i and its softmax attention weight $\text{sfm}(t_i)$, the standard deviation of the softmax attention weight $\text{std}(\text{sfm}(t_i))$ is defined as the second metric. For multi-head attention, the standard deviations over all heads are averaged. This metric represents the degree of training stability.

Ratio between the norms of residual activations and main branch: The ratio between the norm of the residual activations and the norm of the activations of the main branch in each layer is defined as $\|f_i(t)\|/\|t\|$, where $f_i(t)$ can be the attention layer or the FFN layer. This metric denotes the information preservation ability of the network.

Nonlocality: following [14], the nonlocality is defined by summing, for each query patch i , the distances $\|\delta_{ij}\|$ to all the key patches j weighted by their attention score A_{ij} . The number obtained over the query patch is averaged to obtain the nonlocality metric of head h , which can be averaged over the attention heads to obtain the nonlocality of the whole layer l :

$$D_{loc}^{l,h} := \frac{1}{L} \sum_{ij} A_{ij}^{h,l} \|\delta_{ij}\|, D_{loc}^l := \frac{1}{N_h} \sum_h D_{loc}^{l,h},$$

where D_{loc} is the number of patches between the center of attention and the query patch; the further the attention heads look from the query patch, the higher the nonlocality.

Comparisons of the four metrics on DeiT-tiny without distillation token are shown in Figure 2 and Figure 3. From Figure 2 (a) we can see that by using k -NN attention, the averaged cosine similarity is larger than that of using dense self-attention, which reflects that the convergence speed is faster for k -NN attention. Figure 2 (b) shows that the averaged standard deviation of k -NN attention is smoother than that of fully-connected self-attention, and the smoothness will help make the training more stable. Figure 2 (c) and (d) show the ratio between the norms of residual activations and main branch are consistent with each other for k -NN attention and dense attention, which indicates that there is nearly no information lost in k -NN attention by removing the irrelevant tokens. Figure 3 shows that, with k -NN attention, lower layers tend to focus more on the local areas (with more lines being pushed toward the bottom area in Figure 3), while the higher layers still maintain their capability of extracting global

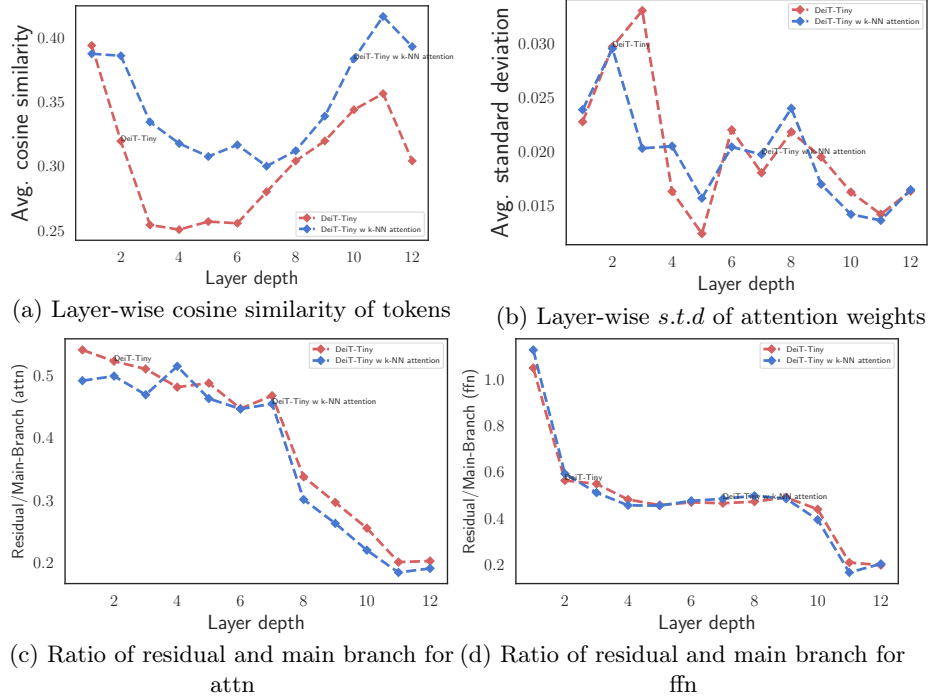


Fig. 2. The properties of k -NN attention. Blue and red dotted lines represent the metrics for k -NN attetion and the original fully-connected self-attention, respectively.

information. Additionally, it is also observed that the non-locality of different layers is spreading more evenly, indicating that they can explore a larger variety of dependencies at different ranges.

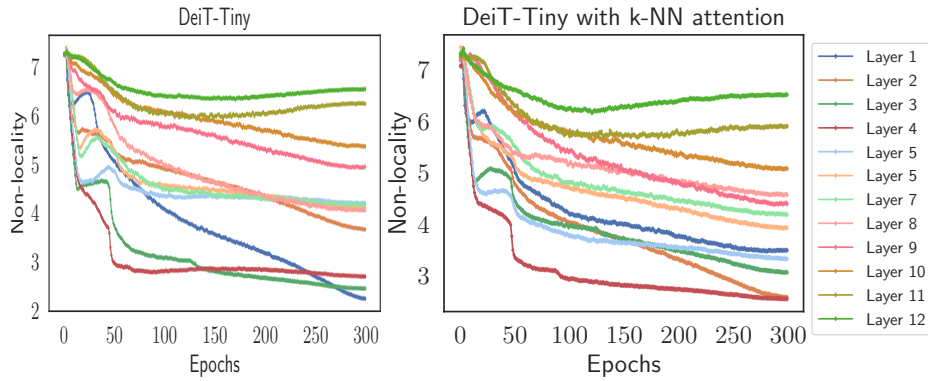


Fig. 3. The nonlocality of DeiT-Tiny. It is plotted averaged over all the images from training set of ImageNet-1k.

4.6 Comparisons with temperature in softmax

k -NN attention effectively zeros the bottom $N - k$ tokens out of the attention calculation. How does this compare with introducing a temperature parameter to softmax over the attention values? We compare our k -NN attention with temperature t in softmax as $\text{softmax}(\text{attn}/t)$. The performance over the t is shown in Table 3. From the Table we can see that small t makes the training crash due to large value of attention values; the performance increases a little bit to 72.5 (baseline 72.2) with t assigned to appropriate values. The k -NN attention is more robust compared with temperature in softmax, and achieves much better performance, 73.0 (k -NN attention) vs 72.5 (best performance for temperature in softmax).

Table 3. The Top-1 (%) over the temperature t in softmax.

t	0.05	0.1	0.25	0.75	2	4	8	16
Top-1 (%)	crash	crash	72.0	72.5	72.5	72.5	72.5	72.1

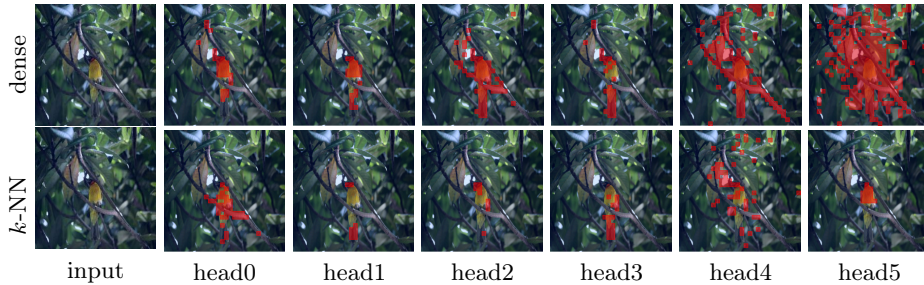
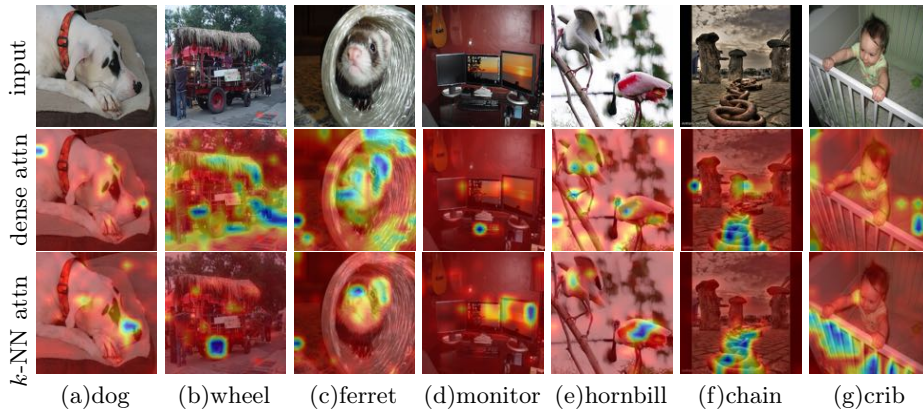


Fig. 4. Self-attention heads from the last layer.

4.7 Visualization

Figure 4 visualizes the self-attention heads from the last layer on Dino-Small [3]. We can see that different heads attend to different semantic regions of an image. Compared with dense attention, the k -NN attention filters out most irrelevant information from background regions which are similar to the foreground, and successfully concentrates on the most informative foreground regions. Images from different classes are visualized in Figure 5 using Transformer Attribution method [5] on DeiT-Tiny. It can be seen that the k -NN attention is more concentrated and accurate, especially in the situations of cluttered background and occlusion.

**Fig. 5.** Visualization using Transformer Attribution [5].**Table 4.** Object detection and Segmentation results for Swin-Tiny and Twins-SVT-Base with/without k -NN attention on the COCO and ADE20K validation sets. All the models are pretrained on ImageNet-1k.

Backbone	COCO		ADE20K	
	Method	mAP(box)	Method	mIoU
Swin-T	Mask R-CNN 3x	46.0	UPerNet	44.5
Swin-T- k -NN	Mask R-CNN 3x	46.2	UPerNet	44.7
Twins-SVT-Base	Mask R-CNN 1x	45.2	UPerNet	47.4
Twins-SVT-Base- k -NN	Mask R-CNN 1x	45.6	UPerNet	47.9

4.8 Object Detection and Semantic Segmentation

To verify the effects of k -NN attention on object detection and semantic segmentation tasks, the widely-used COCO [39] and ADE20K [76] are adopted for evaluation. We adopt Swin-Tiny [41] and Twins-SVT-Base [10] for comparisons due to the well released codes, and the results are shown in Table 4. From the Table we can see that by replacing the vanilla attention with our k -NN attention, the performance increases with almost no overhead.

5 Conclusion

In this paper, we propose an effective k -NN attention for boosting vision transformers. By selecting the most similar keys for each query to calculate the attention, it screens out the most ineffective tokens. The removal of irrelevant tokens speeds up the training. We theoretically prove its properties in speeding up training, distilling noises without losing information, and increasing the performance by choosing a proper k . Several vision transformers are adopted to verify the effectiveness of the k -NN attention.

References

1. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 (2020) [3](#)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020) [3](#)
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294 (2021) [8](#), [9](#), [13](#)
4. Chang, S., Wang, P., Wang, F., Li, H., Feng, J.: Augmented transformer with adaptive graph for temporal action proposal generation. arXiv preprint arXiv:2103.16024 (2021) [3](#)
5. Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: CVPR (2021) [13](#), [14](#)
6. Chen, C.F., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. arXiv preprint arXiv:2103.14899 (2021) [4](#)
7. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: CVPR (2021) [3](#)
8. Chen, Z., Xie, L., Niu, J., Liu, X., Wei, L., Tian, Q.: Visformer: The vision-friendly transformer. arXiv preprint arXiv:2104.12533 (2021) [2](#), [3](#), [4](#), [8](#), [9](#)
9. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019) [3](#)
10. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting spatial attention design in vision transformers. arXiv preprint arXiv:2104.13840 (2021) [4](#), [8](#), [9](#), [14](#)
11. Chu, X., Zhang, B., Tian, Z., Wei, X., Xia, H.: Do we really need explicit position encodings for vision transformers? arXiv preprint arXiv:2102.10882 (2021) [2](#), [3](#)
12. Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers. In: ICLR (2019) [2](#)
13. Correia, G.M., Niculae, V., Martins, A.F.: Adaptively sparse transformers. In: EMNLP. pp. 2174–2184 (2019) [3](#)
14. d’Ascoli, S., Touvron, H., Leavitt, M., Morcos, A., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. arXiv preprint arXiv:2103.10697 (2021) [2](#), [4](#), [11](#)
15. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) [3](#)
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) [2](#), [3](#)
17. El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: Cross-covariance image transformers. arXiv preprint arXiv:2106.09681 (2021) [3](#)
18. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. arXiv preprint arXiv:2104.11227 (2021) [4](#)
19. Fang, J., Xie, L., Wang, X., Zhang, X., Liu, W., Tian, Q.: Msg-transformer: Exchanging local spatial information by manipulating messenger tokens. arXiv preprint arXiv:2105.15168 (2021) [3](#)

20. Gao, P., Lu, J., Li, H., Mottaghi, R., Kembhavi, A.: Container: Context aggregation network. arXiv preprint arXiv:2106.01401 (2021) [3](#)
21. Gong, C., Wang, D., Li, M., Chandra, V., Liu, Q.: Improve vision transformers training by suppressing over-smoothing. arXiv preprint arXiv:2104.12753 (2021) [4](#), [11](#)
22. Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., Douze, M.: Levit: a vision transformer in convnet’s clothing for faster inference. arXiv preprint arXiv:2104.01136 (2021) [4](#)
23. Guo, J., Han, K., Wu, H., Xu, C., Tang, Y., Xu, C., Wang, Y.: Cmt: Convolutional neural networks meet vision transformers. arXiv preprint arXiv:2107.06263 (2021) [4](#)
24. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. arXiv preprint arXiv:2103.00112 (2021) [2](#), [4](#), [8](#), [9](#)
25. Han, L., Wang, P., Yin, Z., Wang, F., Li, H.: Exploiting better feature aggregation for video object detection. In: ACM MM (2020) [3](#)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) [2](#)
27. He, S., Luo, H., Wang, P., Wang, F., Li, H., Jiang, W.: Transreid: Transformer-based object re-identification. arXiv preprint arXiv:2102.04378 (2021) [3](#)
28. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. arXiv preprint arXiv:2103.16302 (2021) [4](#), [8](#), [9](#)
29. Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T.: Axial attention in multidimensional transformers. arXiv preprint arXiv:1912.12180 (2019) [3](#)
30. Huang, Z., Ben, Y., Luo, G., Cheng, P., Yu, G., Fu, B.: Shuffle transformer: Rethinking spatial shuffle for vision transformer. arXiv preprint arXiv:2106.03650 (2021) [3](#)
31. Jiang, Z., Hou, Q., Yuan, L., Zhou, D., Jin, X., Wang, A., Feng, J.: Token labeling: Training a 85.5% top-1 accuracy vision transformer with 56m parameters on imagenet. arXiv preprint arXiv:2104.10858 (2021) [4](#)
32. Jonnalagadda, A., Wang, W., Eckstein, M.P.: Foveater: Foveated transformer for image classification. arXiv preprint arXiv:2105.14173 (2021) [3](#)
33. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) [6](#)
34. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: The efficient transformer. arXiv preprint arXiv:2001.04451 (2020) [3](#)
35. Li, W., Liu, H., Ding, R., Liu, M., Wang, P.: Lifting transformer for 3d human pose estimation in video. arXiv preprint arXiv:2103.14304 (2021) [3](#)
36. Li, X., Hou, Y., Wang, P., Gao, Z., Xu, M., Li, W.: Transformer guided geometry model for flow-based unsupervised visual odometry. Neural Computing and Applications pp. 1–12 (2021) [3](#)
37. Li, X., Hou, Y., Wang, P., Gao, Z., Xu, M., Li, W.: Trear: Transformer-based rgb-d egocentric action recognition. arXiv preprint arXiv:2101.03904 (2021) [3](#)
38. Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021) [4](#)
39. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) [14](#)
40. Liu, P.J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., Shazeer, N.: Generating wikipedia by summarizing long sequences. In: ICLR (2018) [3](#)

41. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021) [2](#), [3](#), [4](#), [8](#), [9](#), [14](#)
42. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: EMNLP (2015) [3](#)
43. Pai, S.: Convolutional neural networks arise from ising models and restricted boltzmann machines [2](#)
44. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. arXiv preprint arXiv:2106.02034 (2021) [3](#)
45. Roy, A., Saffar, M., Vaswani, A., Grangier, D.: Efficient content-based sparse attention with routing transformers. Transactions of the Association for Computational Linguistics **9**, 53–68 (2021) [3](#)
46. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015) [8](#)
47. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML (2019) [2](#)
48. Tay, Y., Bahri, D., Yang, L., Metzler, D., Juan, D.C.: Sparse sinkhorn attention. In: ICML (2020) [3](#)
49. Tay, Y., Dehghani, M., Aribandi, V., Gupta, J., Pham, P., Qin, Z., Bahri, D., Juan, D.C., Metzler, D.: Omninert: Omnidirectional representations from transformers. arXiv preprint arXiv:2103.01075 (2021) [4](#)
50. Tay, Y., Dehghani, M., Bahri, D., Metzler, D.: Efficient transformers: A survey. arXiv preprint arXiv:2009.06732 (2020) [3](#), [4](#)
51. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. arXiv preprint arXiv:2012.12877 (2020) [2](#), [3](#), [8](#), [9](#), [10](#)
52. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. arXiv preprint arXiv:2103.17239 (2021) [2](#), [4](#)
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS (2017) [1](#), [3](#), [4](#)
54. Wang, P., Wang, X., Luo, H., Zhou, J., Zhou, Z., Wang, F., Li, H., Jin, R.: Scaled relu matters for training vision transformers. arXiv preprint arXiv:2109.03810 (2021) [4](#)
55. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122 (2021) [2](#), [4](#)
56. Wang, W., Yao, L., Chen, L., Cai, D., He, X., Liu, W.: Crossformer: A versatile vision transformer based on cross-scale attention. arXiv preprint arXiv:2108.00154 (2021) [3](#)
57. Wang, Y., Huang, R., Song, S., Huang, Z., Huang, G.: Not all images are worth 16x16 words: Dynamic vision transformers with adaptive sequence length. arXiv preprint arXiv:2105.15075 (2021) [3](#)
58. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: CVPR (2021) [3](#)
59. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. arXiv preprint arXiv:2103.15808 (2021) [2](#), [3](#), [4](#), [8](#), [9](#), [10](#)

60. Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., Girshick, R.: Early convolutions help transformers see better. arXiv preprint arXiv:2106.14881 (2021) [4](#)
61. Xie, J., Zeng, R., Wang, Q., Zhou, Z., Li, P.: So-vit: Mind visual tokens for vision transformer. arXiv preprint arXiv:2104.10935 (2021) [4](#), [8](#), [9](#)
62. Xu, T., Chen, W., Wang, P., Wang, F., Li, H., Jin, R.: Cdtrans: Cross-domain transformer for unsupervised domain adaptation. arXiv preprint arXiv:2109.06165 (2021) [3](#)
63. Xu, W., Xu, Y., Chang, T., Tu, Z.: Co-scale conv-attentional image transformers. arXiv preprint arXiv:2104.06399 (2021) [3](#), [4](#)
64. Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., Sun, X.: Evo-vit: Slow-fast token evolution for dynamic vision transformer. arXiv preprint arXiv:2108.01390 (2021) [3](#)
65. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers. arXiv preprint arXiv:2107.00641 (2021) [4](#)
66. Yin, Z., Wang, P., Wang, F., Xu, X., Zhang, H., Li, H., Jin, R.: Transfgu: A top-down approach to fine-grained unsupervised semantic segmentation. In: European conference on computer vision (2022) [3](#)
67. Yu, Q., Xia, Y., Bai, Y., Lu, Y., Yuille, A., Shen, W.: Glance-and-gaze vision transformer. arXiv preprint arXiv:2106.02277 (2021) [3](#)
68. Yu, Z., Li, X., Wang, P., Zhao, G.: Transrppg: Remote photoplethysmography transformer for 3d mask face presentation attack detection. arXiv preprint arXiv:2104.07419 (2021) [3](#)
69. Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W.: Incorporating convolution designs into visual transformers. arXiv preprint arXiv:2103.11816 (2021) [2](#), [3](#), [4](#)
70. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. arXiv preprint arXiv:2101.11986 (2021) [2](#), [3](#), [4](#), [8](#), [9](#), [10](#)
71. Yuan, L., Hou, Q., Jiang, Z., Feng, J., Yan, S.: Volo: Vision outlooker for visual recognition. arXiv preprint arXiv:2106.13112 (2021) [2](#), [4](#), [8](#), [9](#)
72. Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al.: Big bird: Transformers for longer sequences. arXiv preprint arXiv:2007.14062 (2020) [3](#)
73. Zhang, P., Dai, X., Yang, J., Xiao, B., Yuan, L., Zhang, L., Gao, J.: Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. arXiv preprint arXiv:2103.15358 (2021) [4](#)
74. Zhang, Z., Zhang, H., Zhao, L., Chen, T., Pfister, T.: Aggregating nested transformers. arXiv preprint arXiv:2105.12723 (2021) [3](#)
75. Zhao, G., Lin, J., Zhang, Z., Ren, X., Su, Q., Sun, X.: Explicit sparse transformer: Concentrated attention through explicit selection. arXiv preprint arXiv:1912.11637 (2019) [2](#)
76. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV **127**(3), 302–321 (2019) [14](#)
77. Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Hou, Q., Feng, J.: Deepvit: Towards deeper vision transformer. arXiv preprint arXiv:2103.11886 (2021) [4](#)
78. Zhou, D., Shi, Y., Kang, B., Yu, W., Jiang, Z., Li, Y., Jin, X., Hou, Q., Feng, J.: Refiner: Refining self-attention for vision transformers. arXiv preprint arXiv:2106.03714 (2021) [3](#)

- 79. Zhou, J., Wang, P., Wang, F., Liu, Q., Li, H., Jin, R.: Elsa: Enhanced local self-attention for vision transformer. arXiv preprint arXiv:2112.12786 (2021) 3
- 80. Zou, C., Wang, B., Hu, Y., Liu, J., Wu, Q., Zhao, Y., Li, B., Zhang, C., Zhang, C., Wei, Y., et al.: End-to-end human object interaction detection with hoi transformer. In: CVPR (2021) 3