ClusTR: Exploring Efficient Self-attention via Clustering for Vision Transformers

Anonymous authors
Paper under double-blind review

Abstract

Although Transformers have successfully transitioned from their language modelling origins to image-based applications, their quadratic computational complexity remains a challenge, particularly for dense prediction. In this paper we propose a content-based sparse attention method, as an alternative to dense self-attention, aiming to reduce the computation complexity while retaining the ability to model long-range dependencies. Specifically, we cluster and then aggregate key and value tokens, as a content-based method of reducing the total token count. The resulting clustered-token sequence retains the semantic diversity of the original signal, but can be processed at a lower computational cost. Besides, we further extend the clustering-guided attention from single-scale to multi-scale, which is conducive to dense prediction tasks. We label the proposed Transformer architecture ClusTR, and demonstrate that it achieves state-of-the-art performance on various vision tasks but at lower computational cost and with fewer parameters. For instance, our ClusTR small model with 22.7M parameters achieves 83.2% Top-1 accuracy on ImageNet. Source code and ImageNet models will be made publicly available.

1 Introduction

Transformers have driven rapid progress in natural language processing, and have become the predominant model in the field as a result (Vaswani et al., 2017; Brown et al., 2020). The first Transformer to achieve image recognition performance comparable to the firmly established CNN models (*e.g.* ResNet (He et al., 2016) and EfficientNet (Tan & Le, 2019)) was ViT (Dosovitskiy et al., 2021). ViT splits images into 16×16 patches, resulting in a sequence of visual tokens. In contrast to the local receptive fields of CNNs, each token in ViT is able to interact with every other token, irrespective of location, thus enabling the modelling of long-range dependencies.

Although its strength has been demonstrated in various tasks, ViT still suffers from the quadratic complexity in both computation and memory due to the dense token-to-token self-attention. This particularly hinders the applications in dense prediction, such as semantic segmentation. Inspired by CNN models (Krizhevsky et al., 2012; Szegedy et al., 2015; He et al., 2016), recent research (Liu et al., 2021; Wang et al., 2021; Heo et al., 2021; Chu et al., 2021) has developed pyramid architectures for Transformers. The resultant variation in regulable token length and number of channels at various locations and scales enables greater computational and memory efficiency. To further reduce complexity, Swin Transformer (Liu et al., 2021) limited self-attention to a local window, and enabled cross-window connection through the window shifting. This means the computational burden scales linearly with the number of tokens, but at the cost of long-range dependencies. Pyramid Vision Transformer (PVT) (Wang et al., 2021) reduced the spatial dimension of queries and keys using the large-kernel and large-stride convolution. Such a spatial reduction attention suffers from the following two drawbacks. First, the reduced tokens are limited by the lack of fine-grained information. As shown in Figure 1, the downsampled token includes a wide range of content information. Taking the token located in the second row and second column for example, the object of “woman” only occupies a small part of the whole token, and the token also contains a small part of “child” and a large object of “sky”. This may lead to ambiguous semantics for these tokens. Second, the background tokens, like the sky and beach, take up quite a large portion of the entire sequence, which are full of redundant information whilst investing most of the computations. Hence, the aforementioned deficiencies may have a negative effect on the performance.

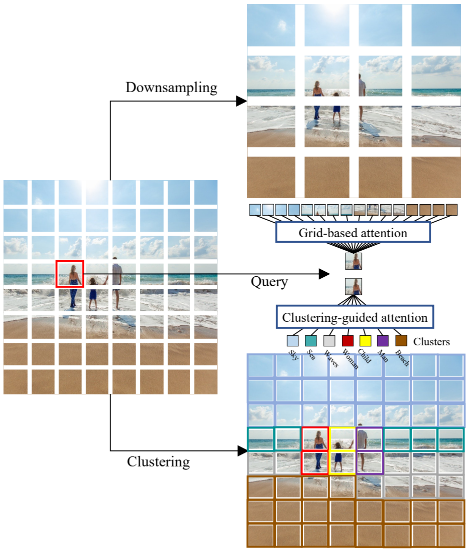


Figure 1: Comparison of grid-based self-attention and our clustering-guided self-attention. Downsampling obfuscates fine-grained image features, and mixes content types into larger tiles. Clustering, in contrast, eliminates *semantic* redundancy and is thus able to operate at higher resolution. This eliminates the over-representation of large objects and 'stuff' (like "sky").

We propose here a form of content-based sparse attention, and a corresponding efficient and versatile vision Transformer. The aim is to reduce the computational complexity of self-attention by reducing the numbers of key and value tokens. In contrast to grid-based downsampling solutions used by (Wang et al., 2021; 2022b) (see Figure 1) we merge tokens according to the similarity of their content rather than their location.

We label the proposed approach clustering-guided self-attention, and the corresponding Transformer as ClusTR. It is comparable to other self-attention models but with value-clustering applied to the key and value tokens (not query tokens). This method has the advantage that the clustered tokens contain not only rich but also explicit semantic information. It is less affected by the background or other large-size objects than grid-based methods. Clustering is also more flexible than grid-based methods in that it allows more control over the final number of tokens. Clustering can also be applied to patches at varying scales, thus exploiting the demonstrated value of multi-scale information in vision (Zhang et al., 2021; Chen et al., 2021a; Ren et al., 2022).

We demonstrate the effectiveness of our clustering-based self-attention on tasks including classification, segmentation, detection, and pose estimation. The experimental results show that ClusTR outperforms its CNN-based and Transformer-based counterparts. For instance, ClusTR achieves the 83.2% and 84.1% Top-1 accuracy on ImageNet with 22.7M and 40.3M parameters, respectively. Our contributions are summarized as follows:

- We propose a content-based self-attention Transformer that clusters and aggregates visual tokens according to their semantic information. Our clustering-guided self-attention reduces computational complexity without compromising long-range context modelling.
- We introduce multi-scale clustering-guided self-attention that is particularly well suited to dense prediction tasks.
- Our ClusTR, as a versatile Transformer backbone, outperforms the current state-of-the-art on four key vision tasks including classification, segmentation, detection, and pose estimation.

2 Related work

Vision Transformer. Transformers have become the dominant architecture in language modelling and have recently demonstrated competitive performance in computer vision. Their ability to exploit long-range interactions between tokens is particularly appealing. (Dosovitskiy et al., 2021) proposed the Vision Transformer (ViT) which achieved superior performance in image recognition tasks over its CNN counterparts. Transformers have since been applied to various vision tasks, including segmentation (Zheng et al., 2021), detection (Carion et al., 2020), low-level vision (Chen et al., 2021b), image generation (Jiang et al., 2021), and pose estimation (Zeng et al., 2022a). ViT requires large volumes of training data due to its weak inductive bias (d'Ascoli et al., 2021). DeiT (Touvron et al., 2021a) utilizes an efficient Transformer optimization strategy that distils another strong classifier to reduce data consumption. T2T-ViT(Yuan et al., 2021) models the local image structure via a Tokens-to-Token (T2T) transformation. CaiT (Touvron et al., 2021b) uses layer scaling to increase the stability of the optimization when training large-scale Transformers. Although achieving record performance on ImageNet (Deng et al., 2009), these methods suffer from the quadratic complexity of dense self-attention. In weight for weight comparisons they are often outperformed by CNNs on dense-prediction tasks, or high-resolution images. Inspired by CNN models, the pyramid Transformer structure (Wang et al., 2021; Liu et al., 2021; Heo et al., 2021; Li et al., 2021; Chu et al., 2021; Chen et al., 2022a; Ren et al., 2022) breaks with the ViT architecture, and particularly its fixed number of tokens and fixed number of channels. These methods have a pyramid structure that can be used as a versatile backbone for both image classification and dense prediction tasks. These pyramid Transformer variants downsample tokens at each stage by convolution with strides (Wang et al., 2021; 2022b), patch merging with linear projections (Liu et al., 2021), or clustering-based patch embedding (Zeng et al., 2022a).

Efficient sparse self-attention. Self-attention is the mechanism by which transformers preform long-range interactions between tokens, and simple dense self-attention methods naturally scale quadratically with the number of tokens. Methods for improving the efficiency of self-attention through sparsifying the set of possible interactions can be categorized as either content-based or location-based.

Location-based sparse attention assumes that not all token-token interactions are equally likely to be valuable. Examples of this selective-attention approach from language modelling include local window sliding attention, global attention, and combinations thereof (Beltagy et al., 2020; Zaheer et al., 2020).In computer vision, (Liu et al., 2021) achieved efficient self-attention by limiting self-attention to a local region, and enabling regions to interact through sliding windows. (Wang et al., 2021) reduced the number of key and value tokens by aggregating the local region to a single token through convolution with large kernels and large strides.

Predefined sparsity patterns do not necessarily match the empirical characteristics of data. Content-based sparse attention methods partition the tokens according to their content. (Roy et al., 2021), for example, clustered the tokens using the k-means algorithm and performed the self-attention in each cluster. (Kitaev et al., 2019) presented an efficient locality sensitivity hashing clustering to divide tokens into chunks. (Wang et al., 2022a) proposed the kNN attention to select the top-k tokens from keys and ignored the rest for each query when computing the attention matrix, thus filtering out noisy tokens and speeding up training. Although spare attention has been studied in these attempts, our ClusTR is different in the following aspects: 1) Compared with Wang et al. (2021); Liu et al. (2021), ClusTR breaks the rigid rules of grid-based token aggregation and makes full use of token representation for efficient vision modelling. 2) (Roy et al., 2021; Kitaev et al., 2019; Wang et al., 2022a) limited the range of self-attention to achieve efficiency, in which only similar tokens in the same cluster can communicate with each other. In contrast, our ClusTR breaks the constraints of limited self-attention range, and encourages to explore global attention patterns from the diverse clustered tokens. 3) Moreover, with the proposed multi-scale attention, ClusTR is superior to these single-scale attention methods when processing dense prediction tasks.

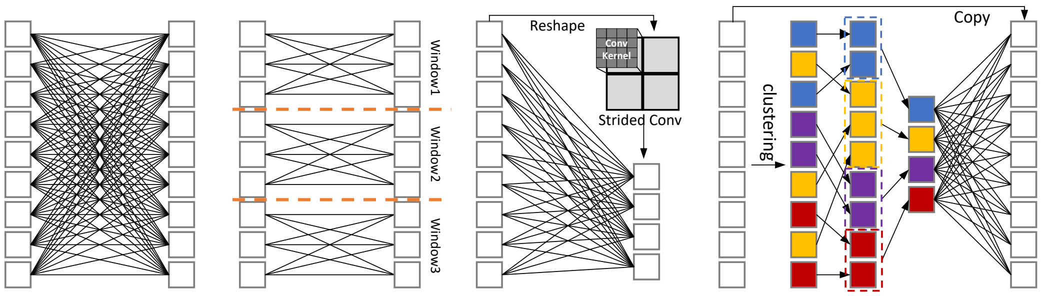
3 Method

As an efficient vision Transformer, ClusTR is different from other counterparts in terms of the self-attention mechanism. As shown in Figure 2, we group vision tokens and aggregate the semantic-similar tokens in the same cluster, aiming to reduce the computational complexity of self-attention. Based on the clustering-guided self-attention, we can easily extend it to a multi-scale version which is benefited from the multi-scale aggregation. In the following, we delve into the ClusTR self-attention and architecture details.

3.1 kNN-based Density Peaks Clustering

We denote the set of vision tokens as $X = [x_1, x_2, ..., x_N]^\intercal \in \mathbb{R}^{N \times C}$, where N and C represent the number of tokens and dimension of the token channel, respectively. Following (Rodriguez & Laio, 2014), we characterize token clusters by a higher density than their neighbors and by a relatively large distance from other tokens with higher densities. As for a token $x_i \in X$, its local density is defined as

$$\rho_i = \exp\,(-\frac{1}{k}\sum_{j \in k\mathrm{NN}(x_i)} d(x_i, x_j)^2) \tag{1}$$



(a) ViT (b) Swin (c) PVT (d) ClusTR

Figure 2: Comparison of self-attention in ViT, Swin, PVT and our proposed method. ViT performs the dense token-to-token self-attention; Swin Transformer divides all tokens into several windows and performs the window-based self-attention; PVT aggregates tokens in a grid by using strided convolution. The proposed method groups vision tokens according to the feature similarity, resulting in compact but semantic tokens for efficient self-attention.

where $d(x_i, x_j)$ refers to the Euclidean distance between x_i and x_j, $k\text{NN}(x_i) = \{j \in X | d(x_i, x_j) \leq d(x_i, x_k)\}$, x_k is the k-th neighbor of x_i. Here, we also define another variable δ_i for the token x_i, which measures the distance between x_i and other high-density tokens.

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j > \rho_i} (d(x_i, x_j)) & if\ \exists \rho_j > \rho_i \\ \max\limits_{j} (d(x_i, x_j)) & if\ \nexists \rho_j > \rho_i \end{cases} \tag{2}$$

If x_i is characterized as a cluster, its local density should be higher than that of its neighbors. Besides, it should also have a relatively large distance from other higher-density tokens. To this end, a decision value $\gamma_i = \rho_i * \delta_i$ can be computed to locate the density peaks efficiently. The token clusters are specialized with both large density ρ and large distance δ. After that, the remaining tokens are assigned to the same cluster as their nearest tokens with higher density. Based on the cluster index, we can partition all tokens in X into M clusters, denoted by $G = \{G_1, G_2, ..., G_M\}$.

The tokens in the same cluster are aggregated to generate a cluster representative token, formulated by

$$\texttt{Cluster}(X; \lambda) = [h_1, h_2, ..., h_{N/\lambda}] \in \mathbb{R}^{M \times C} \tag{3}$$

where $\lambda = N/M$ is the token reduction ratio, $h_i = \sum_{x_i \in G_i} w_i \cdot x_i$, and w_i is the learnable parameter for each token x_i. Note that the number of aggregated cluster representative tokens is far smaller than that of the original visual tokens X, *i.e.*, $N >> N/\lambda$. Such a clustering-guided token aggregation condenses a lot of visual tokens, benefiting the efficient self-attention process.

3.2 Clustering-Guided Self-Attention

The attention module is one of the core components of the Transformer. Following (Vaswani et al., 2017), most of Transformers and their variants apply the multi-head self-attention mechanism to model the long-range dependencies. For each head, the query Q, key K, and value V have the size of $N \times C$. The scaled dot-product attention can be formulated as

$$\texttt{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^\mathsf{T}}{\sqrt{s}})V \tag{4}$$

where s is the scaling factor. Although the above self-attention can be implemented in a fast manner by using highly optimized matrix multiplication, it still suffers from the high computation complexity, *i.e.*, $O(N^2)$, especially for the abundant vision tokens. To address this issue, we propose a clustering-guided efficient self-attention that clusters and aggregates the semantic-similar tokens in the same cluster to reduce the computation complexity. Based on the clustering algorithm in Sec. 3.1, the proposed efficient self-attention is reformulated as

$$\texttt{ClusAtt}(Q, K, V; \lambda) = \text{Softmax}(\frac{Q \cdot \texttt{Cluster}(K; \lambda)^\mathsf{T}}{\sqrt{s}}) \cdot \texttt{Cluster}(V; \lambda) \tag{5}$$

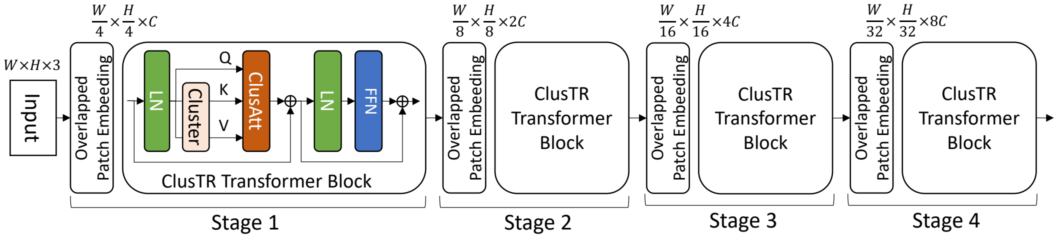

$W \times H \times 3$
Input
$\frac{W}{4} \times \frac{H}{4} \times C$
Overlapped Patch Embeeding
LN
Cluster
Q
K
V
ClusAtt
LN
FFN
ClusTR Transformer Block
Stage 1
$\frac{W}{8} \times \frac{H}{8} \times 2C$
Overlapped Patch Embeeding
ClusTR Transformer Block
Stage 2
$\frac{W}{16} \times \frac{H}{16} \times 4C$
Overlapped Patch Embeeding
ClusTR Transformer Block
Stage 3
$\frac{W}{32} \times \frac{H}{32} \times 8C$
Overlapped Patch Embeeding
ClusTR Transformer Block
Stage 4

Figure 3: The architecture of our ClusTR.

After clustering, the tokens of key and value are decreased by λ times, reducing the computation complexity from $O(N^2)$ to $O(\frac{N^2}{\lambda})$. Based on the single-head attention, the multi-head attention can be implemented in parallel as

$$\texttt{MH-ClusAtt}(X;\lambda) = \Phi(\bigcup_{i=1}^{H} \texttt{ClusAtt}(XW_i^Q, XW_i^K, XW_i^V;\lambda)) \tag{6}$$

where $\cup$ refers to the concatenation operation, Φ aggregates the feature representation of H attention heads through a linear projection function. W_i^Q, W_i^K, and W_i^V are linear projections to generate query, key, and value tokens.

3.3 Multi-Scale Self-Attention

Here we extend the proposed clustering-guided self-attention from single-scale to multi-scale. For the multi-scale aggregation, we replace the single λ in Eq. 3 with a set of factors $\lambda_1, ..., \lambda_L$, where L refers to the number of scales. Then, the multi-scale clustering can be described as

$$\texttt{Cluster}(X;\lambda_1,...,\lambda_L) = [h_1^{\lambda_1}, ..., h_{N/\lambda_1}^{\lambda_1}; ...; h_1^{\lambda_L}, ..., h_{N/\lambda_L}^{\lambda_L}] \in \mathbb{R}^{(\frac{N}{\lambda_1}+...+\frac{N}{\lambda_L})\times C} \tag{7}$$

The computational complexity of multi-scale attention is $O(N^2(\frac{1}{\lambda_1}+...+\frac{1}{\lambda_L}))$. And the multi-head multi-scale clustering-guided self-attention can be described as

$$\texttt{MHMS-ClusAtt}(X;\lambda_1,...,\lambda_L) = \Phi(\sum_{j=1}^{L}\bigcup_{i=1}^{H} \texttt{ClusAtt}(XW_i^Q, XW_i^K, XW_i^V;\lambda_j)) \tag{8}$$

where the linear projection Φ is used to aggregate the feature representation of H attention heads and L scales.

3.4 ClusTR Transformer Architecture

The basic ClusTR model is composed of four stages, as shown in Figure 3. We follow (Ren et al., 2022) and employ the overlapped patch embedding at the beginning of each stage to model local continuity. Based on the clustering-guided self-attention, the Transformer block of ClusTR can be computed as

$$\begin{aligned} z_l' &= \texttt{MHMS-ClusAtt}(\texttt{LN}(z_{l-1})) + z_{l-1} \\ z_l &= \texttt{FFN}(\texttt{LN}(z_l')) + z_l' \end{aligned} \tag{9}$$

where LN is the layer normalization, and FFN is the fully connected feedforward network. Note that the token reduction ratio λ can be defined as any value during the clustering process. To balance the efficiency and accuracy, we set λ to $\{64, 16\}$, $\{16, 4\}$, $\{4, 1\}$, $\{1\}$ from the first to the last stage, respectively. We build the tiny model, called ClusTR-T, that has a similar model size and computation complexity to PVT-Tiny/PVTv2-B1. Based on this, we scale up ClusTR-T to the small and base variants, called ClusTR-S, and ClusTR-B, which have the model size and computation complexity of about 2×, and 4× compared to the tiny version. The specific architecture details and hyper-parameters can be found in Appendix.

Table 1: Image classification performance of different backbones on the ImageNet-1K validation set. Here 'Params.' refers to the number of the model parameters, and FLOPs is calculated based on the input size of 224×224.

Methods	Resolution	Params. (M)	FLOPs (G)	Top-1 (%)	Reference
ConvNets					
RegNetY-4G (Radosavovic et al., 2020)	224	21.0	4.0	80.0	CVPR20
RegNetY-8G (Radosavovic et al., 2020)	224	39.0	8.0	81.7	CVPR20
ConvNeXt-T (Liu et al., 2022)	224	29.0	4.5	82.1	CVPR22
ConvNeXt-S (Liu et al., 2022)	224	50.0	8.7	83.1	CVPR22
MLPs					
CycleMLP-T (Chen et al., 2022b)	224	28.0	4.4	81.3	ICLR22
CycleMLP-S (Chen et al., 2022b)	224	50.0	8.5	82.9	ICLR22
AS-MLP-T (Lian et al., 2022)	224	28.0	4.4	81.3	ICLR22
AS-MLP-S (Lian et al., 2022)	224	50.0	8.5	83.1	ICLR22
Transformers					
PVT-T (Wang et al., 2021)	224	13.0	1.9	75.1	ICCV21
PVT-ACmix-T (Pan et al., 2022)	224	13.0	2.0	78.0	CVPR22
PVTv2-b1 (Wang et al., 2022b)	224	13.1	2.1	78.7	CVM22
QuadTree-B-b1 (Tang et al., 2022)	224	13.6	2.3	80.0	ICLR22
ClusTR-T	224	11.7	2.2	**80.2**	Ours
PVT-S (Wang et al., 2021)	224	24.5	3.8	79.8	ICCV21
Swin-T (Liu et al., 2021)	224	29.0	4.5	81.3	ICCV21
Twins-SVT-S (Chu et al., 2021)	224	24.0	2.9	81.7	NeurIPS21
PVTv2-b2 (Wang et al., 2022b)	224	25.4	4.0	82.0	CVM22
HRViT-b2 (Gu et al., 2022)	224	32.5	5.1	82.3	CVPR22
TCFormer (Zeng et al., 2022b)	224	25.6	5.9	82.4	CVPR22
CrossFormer-S (Wang et al., 2022c)	224	30.7	4.9	82.5	ICLR22
RegionViT-S (Chen et al., 2022a)	224	30.6	5.3	82.6	ICLR22
CSWin-T (Dong et al., 2022)	224	23.0	4.3	82.7	CVPR22
QuadTree-B-b2 (Tang et al., 2022)	224	24.2	4.5	82.7	ICLR22
ClusTR-S	224	22.7	4.8	**83.2**	Ours
PVT-L (Wang et al., 2021)	224	61.4	9.8	81.7	ICCV21
HRViT-b3 (Gu et al., 2022)	224	37.9	5.7	82.8	CVPR22
Swin-S (Liu et al., 2021)	224	50.0	8.7	83.0	ICCV21
RegionViT-M (Chen et al., 2022a)	224	41.2	7.4	83.1	ICLR22
Twins-SVT-B (Chu et al., 2021)	224	56.0	8.6	83.2	NeurIPS21
CrossFormer-B (Wang et al., 2022c)	224	52.0	9.2	83.4	ICLR22
PVTv2-b4 (Wang et al., 2022b)	224	62.6	10.1	83.6	CVM22
Quadtree-B-b3 (Tang et al., 2022)	224	46.3	7.8	83.7	ICLR22
ClusTR-B	224	40.2	7.5	**84.1**	Ours

4 Experiment

We evaluate ClusTR on four representative computer vision tasks, including image classification, semantic segmentation, object detection, and pose estimation. We also investigate the effectiveness of each part of ClusTR in the ablation section.

4.1 Classification on ImageNet-1K

Dataset: We conduct image classification experiments on the ImageNet-1K dataset (Deng et al., 2009), which includes 1.28 million training images and 50K validation images from 1,000 categories. **Setting:** We randomly crop 224×224 regions as the input. Following (Wang et al., 2022b), we apply a rich set of data augmentations to diversify the training set, including random cropping, random flipping, random erasing, label-smoothing regularization, CutMix, and Mixup. We adopt the AdamW optimizer (Loshchilov & Hutter, 2018) with a cosine decaying learning rate (Loshchilov & Hutter, 2017), a momentum of 0.9, and a weight decay of 0.05, to train our ClusTR model. We set the initial learning rate to 0.001, batch size to 1024, and epochs to 300, which are popular for ImageNet training. During the inference time, we take a 224×224 center crop as the input and adapt the Top-1 accuracy as the evaluation metric.

Table 2: Semantic segmentation performance of different backbones on the ADE-20K validation set. Here '*' indicates that the numbers are cited from the reproduced results of Twins.

	Semantic FPN 80k		UperNet 160K	
Methods	Params. (M)	mIOU (%)	Params. (M)	mIOU (%)
ResNet-50 (He et al., 2016)	28.5	36.7	-	-
PVT-S (Wang et al., 2021)	28.2	39.8	-	-
Swin-T* (Liu et al., 2021)	31.9	41.5	59.9	44.5
CycleMLP-b2 (Chen et al., 2022b)	30.6	43.4	-	-
ConvNeXt-T (Liu et al., 2022)	-	-	60.0	46.0
Twins-SVT-S (Chu et al., 2021)	28.3	43.2	54.4	46.2
RegionViT-S+ (Chen et al., 2022a)	35.7	45.3	-	-
CrossFormer-S (Wang et al., 2022c)	34.3	46.0	62.3	47.6
MPViT-S (Lee et al., 2022)	-	-	52.0	48.3
ClusTR-S (Ours)	26.4	**48.0**	52.5	**49.6**

Results: In Table 1, we compare ClusTR to other advanced backbones based on ConvNets, MLPs, and Transformers. Compared with the advanced Transformer-based methods, ClusTR outperforms the Transformer-based architectures with comparable or fewer parameters and computation budgets, surpassing 1.9% than Swin Transformer (ClusTR-S 83.2 vs. Swin-T 81.3), and 1.5% than PVTv2 (ClusTR-T 80.2 vs. PVTv2-b1 78.7). Compared to the ConvNet-based methods, ClusTR is superior to keep a balance between accuracy and complexity. With a similar complexity budget, ClusTR achieves 1.1% performance gain over ConvNets (ClusTR-S 83.2 vs. ConvNeXt-T 82.1). With a comparable accuracy (ClusTR 83.2 vs. ConvNeXt-S 83.1), ClusTR reduces the model complexity of ConvNexts by half (ClusTR-S 22.7M/4.8G vs. ConvNeXt-S 50M/8.7G). Such an advantageous accuracy-complexity trade-off still remains when compared to MLP-based methods.

4.2 Semantic Segmentation on ADE20K

Dataset: We conduct semantic segmentation experiments on the ADE20K dataset (Zhou et al., 2017), which includes 20,210 training images and 2,000 validation images from 150 fine-grained semantic categories. **Settings:** We randomly resize and crop 512×512 image patches as the input and set the batch size to 16. We empoy the ClusTR-S, pre-trained on ImageNet, as the backbone, and evaluate it with two segmentation architectures, *i.e.*, Semantic FPN (Kirillov et al., 2019) and UperNet (Xiao et al., 2018b). The segmentation training process follows the default settings in (Wang et al., 2022b) and (Liu et al., 2021). When training the Semantic FPN, we adopt the AdamW optimizer (Loshchilov & Hutter, 2018) with an initial learning rate of 0.0001 and a weight decay of 0.0001, and set the number of iterations to 80K. As for UperNet, we adopt the AdamW optimizer with an initial learning rate of 0.00006 and a weight decay of 0.01, and set the number of iterations to 160K. We also warm up the model linearly for the first 1500 iterations. During the test, we re-scale the shorter side of the input image to 512 pixels and adapt the mIOU metric for evaluation.

Results: As shown in Table 2, we can see that ClusTR outperforms these advanced and popular backbones, including ConvNets-based and Transformer-based, in both semantic FPN and UpperNet modes. Compared to the ConvNet-based backbones, the proposed ClusTR achieves better segmentation performance (ClusTR 48.0 vs. ResNet 36.7 with Semantic FPN; ClusTR 49.6 vs. ConvNeXt 46.0 with UpperNet) while using fewer parameters. Compared with the Transformer-based methods, ClusTR outperforms other counterparts in both semantic FPN and UpperNet modes with comparable or even fewer parameters, surpassing CrossFormer (Wang et al., 2022c) by 2.0%, and MPViT (Lee et al., 2022) by 1.3%.

4.3 Object detection on COCO

Dataset: We perform object detection and instance segmentation experiments on the COCO2017 dataset (Lin et al., 2014), which includes 118,287 training images and 5,000 validation images from 80 categories. **Settings:** We use the ClusTR-S pre-trained on ImageNet as the backbone of two mainstream detectors, *i.e.*, RetinaNet (Lin et al., 2017) and Mask R-CNN (He et al., 2017). We follow the default settings of PVTv2 (Wang et al., 2022b) and mmdetection (Chen et al., 2019). We adopt the AdamW optimizer with a batch size of 16, and perform the 1× training schedule with 12

Table 3: Detection and instance segmentation performance of Mask-RCNN with different backbones on the COCO validation set.

Methods	Params. (M)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet-50 (He et al., 2016)	44.2	38.0	58.6	41.4	34.4	55.1	36.7
PVT-S (Wang et al., 2021)	44.1	40.4	62.9	43.8	37.8	60.1	40.3
Swin-T (Liu et al., 2021)	47.8	42.2	64.6	46.2	39.1	61.6	42.0
Twins-SVT-S (Chu et al., 2021)	44.0	43.4	66.0	47.3	40.3	63.2	43.4
CrossFormer-S (Wang et al., 2022c)	50.2	45.4	68.0	49.7	41.4	64.8	44.6
ClusTR-S (Ours)	42.3	**47.0**	**68.7**	**51.6**	**42.5**	**65.9**	**45.9**

Table 4: Detection performance of RetinaNet with different backbones on the COCO validation set.

Methods	Params. (M)	AP^b	AP^b_{50}	AP^b_{75}	AP_S	AP_M	AP_L
ResNet-50 (He et al., 2016)	37.7	36.3	55.3	38.6	19.3	40.0	48.8
PVT-S (Wang et al., 2021)	34.2	40.4	61.3	43.0	25.0	42.9	55.7
CycleMLP-b2 (Chen et al., 2022b)	36.5	40.6	61.4	43.2	22.9	44.4	54.5
Swin-T (Liu et al., 2021)	38.5	41.5	62.1	44.2	25.1	44.9	55.5
Twins-SVT-S (Chu et al., 2021)	34.3	43.0	64.2	46.3	28.0	46.4	57.5
RegionViT-B (Chen et al., 2022a)	83.4	43.3	65.2	46.4	29.2	46.4	57.0
CrossFormer-S (Wang et al., 2022c)	40.8	44.4	65.8	47.4	28.2	48.4	59.4
Shunted-S (Ren et al., 2022)	32.1	45.4	65.9	49.2	28.7	49.3	60.0
ClusTR-S (Ours)	32.4	**45.8**	**66.4**	**49.5**	**30.4**	**49.5**	**61.2**

epochs. During training, we re-scale the shorter side of the input image to 800 pixels while keeping the longer side no more than $1,333$ pixels. During test, the shorter side of input images is resized to 800 pixels, and the bbox mAP (AP^b) and mask mAP (AP^m) are used as evaluation metrics.

Results: As shown in Table 3, with comparable/fewer parameters, our ClusTR model surpasses both ConvNet- and Transformer-based competitors when using Mask-RCNN for object detection and instance segmentation. Compared to ConvNet backbones, our model outperforms ResNet (He et al., 2016) by 9.0 points for box AP, and 8.1 points for mask AP. Compared to Transformer backbones, our model achieves 6.6 box AP/4.7 mask AP over PVT, and 4.8 box AP/3.4 mask AP over Swin. Besides, Table 4 reports the detection performance of different backbones when using RetinaNet as a detector. Our model achieves the 45.8 box AP with only 32.4M parameters, outperforming other competitors especially in detecting small objects. We clarify that these results are expected, since the proposed clustering-guided self-attention is able to pay equal attention to diverse objects, insensitive to the object size, which is particularly beneficial for small objects.

4.4 2D Whole-body Pose Estimation on COCO.

Dataset: We perform pose estimation experiments on the COCOWholeBody V1.0 dataset (Jin et al., 2020), which contains 133 keypoints, including 17 for the body, 6 for the feet, 68 for the face, and 42 for the hands. **Settings:** We follow the same settings in (Zeng et al., 2022a), and adopt the AdamW optimizer with an initial learning rate of 0.0005 (Loshchilov & Hutter, 2017), a momentum of 0.9, and a weight decay of 0.01. We set the batch size to 512, and the number of epochs to 210. The OKS-based Average Precision (AP) and Average Recall (AR) are used as evaluation metrics.

Results: In Table 5, we compare ClusTR with other advanced models on COCOWholeBody V1.0 dataset. Our model achieves the new state-of-the-art performance on the pose estimation (59.4% AP and 69.7% AR), outperforming the best ConvNet-based HRNet by 4.1 AP and 7.1 AR, and surpassing the best Transformer-based TCFormer by 2.2 AP and 1.9 AR.

4.5 Ablations

We perform the following ablation experiments to further verify the effectiveness of ClusTR. All classification experiments are conducted based on ClusTR-T and the number of training epochs is set to 100. The segmentation experiments are conducted based on the pre-trained ClusTR-T and the Semantic FPN segmentation architecture.

Table 5: Pose estimation performance of different backbones on the COCOWholeBody V1.0 dataset. Here '*' indicates that the numbers are cited from the reproduced results of TCFormer.

Methods	Resolution	body		foot		face		hand		whole	
		AP	AR	AP	AR	AP	AR	AP	AR	AP	AR
ZoomNet* (Jin et al., 2020)	384×288	74.3	80.2	79.8	86.9	62.3	70.1	40.1	49.8	54.1	65.8
SBL-Res152* (Xiao et al., 2018a)	256×192	68.2	76.4	66.2	78.8	62.4	72.8	48.2	60.6	54.8	66.1
HRNet-w32* (Sun et al., 2019)	256×192	70.0	74.6	56.7	64.5	63.7	68.8	47.3	54.6	55.3	62.6
PVTv2-b2 (Wang et al., 2022b)	256×192	69.6	77.3	69.0	80.3	64.9	74.8	54.5	65.9	57.5	68.0
TCFormer (Zeng et al., 2022a)	256×192	69.1	77.0	69.8	81.3	64.9	74.6	53.5	65.0	57.2	67.8
ClusTR-S (Ours)	256×192	**71.4**	**78.8**	**73.3**	**83.8**	**66.5**	**75.7**	**55.9**	**67.1**	**59.4**	**69.7**

Table 6: Comparison of different sparse attentions on the ImageNet-1K dataset.

Methods	Token Aggregation	Params. (M)	FLOPs (G)	Top1 (%)
SRA (Wang et al., 2021)	Grid-based	13.2	2.1	76.7
SRA+kNN attention (Wang et al., 2022a)	Grid-based	13.2	2.1	76.8
ClusTR (Ours)	Clustering	10.8	2.0	77.2

Table 7: Comparison of single-scale and multi-scale attention on ImageNet-1K and ADE-20K.

	Reduction ratios				Params. (M)	FLOPs (G)	Top1	mIOU
	Stage1	Stage2	Stage3	Stage4				
Single-scale	64	16	4	1	10.8	2.0	77.2	41.2
	16	4	1	1	10.8	2.1	77.4	41.8
Multi-scale	{64, 16}	{16, 4}	{4, 1}	1	11.7	2.2	77.9	42.6

Grid-based vs. clustering-guided token aggregation: Token aggregation is an important operation in the self-attention process that dramatically reduces the computation complexity. We compare the clustering-guided token aggregation to the convolution-based grid token aggregation. Following the spatial reduction attention (SRA) in (Wang et al., 2021), we utilize the convolution with large strides to achieve the grid token aggregation. Note that the other settings are the same for a fair comparison. Table 6 reveals that our clustering-guided method not only reduce the parameters and FLOPs, but also improve 0.5 points of Top1 accuracy (grid-based 76.7 *vs.* clustering-guided 77.2).

Compared to different sparse attentions: We also compare the clustering-guided attention to the sparse attention method, kNN attention (Wang et al., 2022a), which is embedded in SRA to speed up self-attention learning. As compared in Table 6, the kNN attention achieves a slight performance gain (0.1 points) over SRA without increasing parameters and FLOPs. It is noteworthy that our ClusTR not only outperforms kNN attention by 0.4% but also reduces about 18% parameters. It demonstrates that the proposed ClusTR is superior to modeling abundant semantic dependencies, thus leading to better performance.

Single-scale vs. multi-scale attention: In Table 7, we compare the single-scale attention with two reduction ratios and multi-scale attention. For the single-scale, the smaller reduction ratio keeps more detailed information, thus contributing to better accuracy, especially for dense prediction tasks (+0.6 points for segmentation). By contrast, the multi-scale attention outperforms the single-scale attention by at least 0.5 points on ImageNet and at least 0.8 points on segmentation, though it suffers from a slight increase of parameters (+0.9M) and FLOPs (+0.1G).

5 Conclusion

The dense self-attention in Transformers suffers from the high computation complexity when processing vision tasks, especially on dense prediction scenarios or high-resolution images. In this work, we propose the content-based sparse attention that clusters vision tokens and aggregates them in the same cluster. The clustering-guided self-attention not only reduces the computation complexity but also invests the explicit and intensive semantics to each aggregated token, thus contributing to better performance. Moreover, we extend it from single-scale to multi-scale self-attention, benefiting the dense prediction tasks. Based on the proposed self-attention method, we build a versatile Transformer model, called ClusTR. We conduct extensive experiments to demonstrate the effectiveness of ClusTR, and achieve state-of-the-art performance on various vision tasks, including image recognition, semantic segmentation, object detection, and pose estimation.

REFERENCES

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *International Conference on Learning Representations*, 2022a.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021a.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12299–12310, 2021b.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Shoufa Chen, Enze Xie, GE Chongjian, Runjian Chen, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. In *International Conference on Learning Representations*, 2022b.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12124–12134, 2022.

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 *International Conference on Learning Representations*, 2021.

Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pp. 2286–2296. PMLR, 2021.

Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, Yu-Hsin Chen, Liangzhen Lai, Vikas Chandra, and David Z Pan. Multi-scale high-resolution vision transformer for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12094–12103, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11936–11945, 2021.

Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34, 2021.

Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *European Conference on Computer Vision*, pp. 196–214. Springer, 2020.

A. Kirillov, R. Girshick, K. He, and P. Dollar. Panoptic feature pyramid networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7287–7296, June 2022.

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.

Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. In *International Conference on Learning Representations*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018.

Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 815–825, 2022.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Sucheng Ren, Daquan Zhou, Shengfeng He, Jiashi Feng, and Xinchao Wang. Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10853–10862, 2022.

Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *science*, 344(6191):1492–1496, 2014.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5693–5703, 2019.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *International Conference on Learning Representations*, 2022.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Wen Xie, Hao Li, and Rong Jin. Kvt: k-nn attention for boosting vision transformers. In *European conference on computer vision*, 2022a.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 568–578, October 2021.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022b.

Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *International Conference on Learning Representations*, 2022c.

Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 466–481, 2018a.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018b.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 558–567, 2021.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11101–11111, 2022a.

Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11101–11111, 2022b.

Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2998–3008, 2021.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6881–6890, 2021.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Appendix of “ClusTR: Exploring Efficient Self-attention via Clustering for Vision Transformers”

Anonymous authors
Paper under double-blind review

A1 Overview

In this material, we provide more experimental details and results to supplement the main submission. We first provide the limitations and future Work (Section A2). We then provide the architecture settings of ClusTR variants (Section A3), and the training strategies used in classification, segmentation, and detection tasks respectively (Section A4). To further verify the effectiveness of the proposed ClusTR, we also compare the curves of training loss and validation performance on ImageNet-1K (Section A5). Finally, we visualize qualitatively the inference results from different tasks (Section A6).

A2 Limitations and Future Work

As for the design of ClusTR, we manually set the same token reduction ratio for all samples, resulting in the number of clusters for each sample being only related to its resolution. In contrast, the image content may play a more important role in deciding how many clusters should be produced. Thus, a method of adaptively selecting the number of clusters according to the content is necessary, which would be beneficial to further improve the accuracy and efficiency of ClusTR, and we leave it for future work. Besides, our proposed backbone ClusTR is general and can be applied to a broader range of applications, *e.g.*, medical image analysis and vision-language tasks, which would also be explored in future work.

A3 Architecture details of ClusTR

In Table A1, we provide the architecture hyper-parameters of three ClusTR variants, including Transformer layers/channels/heads and multi-scale token reduction ratios at four stages.

Table A1: Architectures of ClusTR variants. Here ‘L, C, H’ represents the number of Transformer layers, channels, and heads, respectively. λ is the token reduction ratio.

	Output_size	ClusTR-T				ClusTR-S				ClusTR-B			
		L	C	H	λ	L	C	H	λ	L	C	H	λ
Stage1	W/4 * H/4	1	64	1	{64,16}	3	64	1	{64,16}	3	64	1	{64,16}
Stage2	W/8 * H/8	2	128	2	{16,4}	5	128	2	{16,4}	5	128	2	{16,4}
Stage3	W/16 * H/16	6	256	4	{4,1}	13	256	4	{4,1}	18	320	5	{4,1}
Stage4	W/32 * H/32	1	512	8	1	2	512	8	1	3	512	8	1

A4 Implementation Settings

A4.1 Classification on ImageNet-1K

In Table A2, we provide the hyper-parameter settings on ImageNet-1K for our ClusTR, which mainly follow Wang et al. (2022b).

Table A2: ImageNet-1K training settings for our ClusTR.

Classification on ImageNet-1K		
Training Configs		ClusTR-T/S/B
Input size		224 × 224
Data augmentation	Rand augment	(9, 0.5)
	Rand cropping	Yes
	Rand flipping	Yes
	Rand erasing	0.25
	Label-smoothing	0.1
	CutMix	1
	Mixup	0.8
Dropout		0.1/0.1/0.3
Clip grad		None/None/1.0
Optimizer		adamw
Optimizer momentum		0.9
Learning rate		$0.0005 \times \frac{batchsize}{512}$
Learning rate schedule		Cosine decay
Weight decay		0.05
Batch size		1024
Epochs		300
Warmup epochs		5

A4.2 Semantic Segmentation on ADE20K

In Table A3, we provide the training settings of Semantic FPN (Kirillov et al., 2019) and UperNet (Xiao et al., 2018) on ADE20K, which follows Wang et al. (2022b) and Liu et al. (2021). As done in the abovementioned papers, we used the pre-trained ClusTR model (on ImageNet-1K) as the segmentation backbone.

Table A3: ADE20K training settings for our ClusTR.

Semantic segmentation on ADE20K			
Training Configs		Semantic FPN	UperNet
Pre-trained weights		ClusTR-S on ImageNet-1K	
Input size		512 × 512	
Data augmentation	Rand scaling	img scale=(2048, 512), ratio range=(0.5, 2.0)	
	Rand cropping	crop size=(512, 512), cat max ratio=0.75	
	Rand flipping	0.5	
Dropout		0.1	
Optimizer		adamw	
Learning rate		0.0001	0.00006
Learning rate schedule		poly, power=0.9	poly, power=1.0
Weight decay		0.0001	0.01
Batch size		16	
Interations		80, 000	160, 000
Warmup interations		No	1500

A4.3 Object detection on COCO

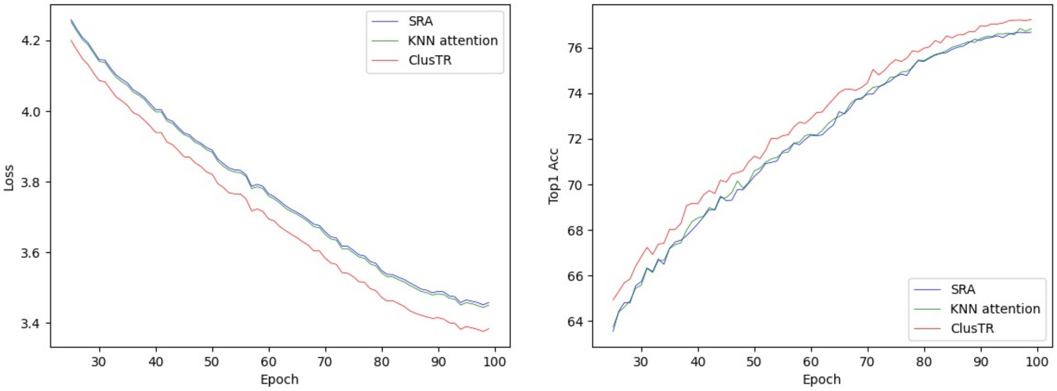
In Table A4, we provide the hyper-parameter details of object detection on COCO. Similar to the segmentation task, we use the ClusTR-S pre-trained on ImageNet-1K as the backbone of two mainstream detectors, *i.e.*, RetinaNet (Lin et al., 2017) and Mask R-CNN (He et al., 2017). We follow the default settings of PVTv2 (Wang et al., 2022b) and mmdetection (Chen et al., 2019).

A5 Training Loss

We plot the curves of the training loss and validation performance obtained by different methods on the image classification task in Figure A1. Compared with the spatial-reduction attention

Table A4: COCO object detection training settings for our ClusTR.

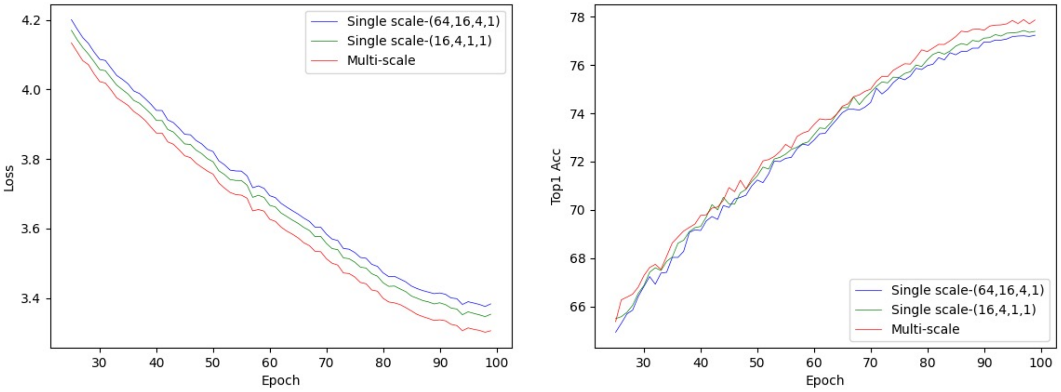
Object detection on COCO			
Training Configs		Mask-RCNN	RetinaNet
Pre-trained weights		ClusTR-S on ImageNet-1K	
Data augmentation	Rand scaling	img scale=(1333, 800), keep ratio=True	
	Rand flipping	0.5	
Dropout		0.1	
Optimizer		adamw	
Learning rate		0.0002	0.0001
Learning rate schedule		step, drops 10× at 8th epoch and 11th epoch	
Weight decay		0.0001	
Batch size		16	
Epochs		12	
Warmup interations		500	



(a) Training Loss

(b) Validation Top1 accuracy

Figure A1: Training loss and validation Top 1 accuracy of image classification for different sparse attentions.



(a) Training Loss

(b) Validation Top1 accuracy

Figure A2: Training loss and validation Top 1 accuracy of image classification for single scale and multi-scale attentions.

(SRA) (Wang et al., 2021) and kNN based sparse attention (Wang et al., 2022a), the loss of our clustering-guided attention is consistently lower and the top 1 accuracy is higher.

We also compare the training loss and validation performance of the single-scale attention and multi-scale attention in Figure A2. We can see that multi-scale attention outperforms single-scale attention by a lower training loss and higher performance. In addition, for the single-scale, the smaller reduction ratio achieves lower training loss and better performance.

A6 Visualized Results

Figure A3 shows some examples of the clustered vision tokens obtained from the third stage of ClusTR-S. It reveals that ClusTR is able to locate the tokens with similar semantics and then group them into a cluster systematically. Besides, we also observe that the scale of each cluster, i.e., the number of tokens, can be adaptively adjusted by the model. For instance, in the first image, a large number of tokens from the curtain background are grouped into one cluster, while more detailed information, like the cakes, flowers on cakes, and candles are identified as different clusters, which provide the fine-grained and semantic-rich information for the clustered tokens. We also provide some visualization results of different downstream tasks, including semantic segmentation (see Figure A4), object detection (see Figure A5), and whole-body pose estimation (see Figure A6).

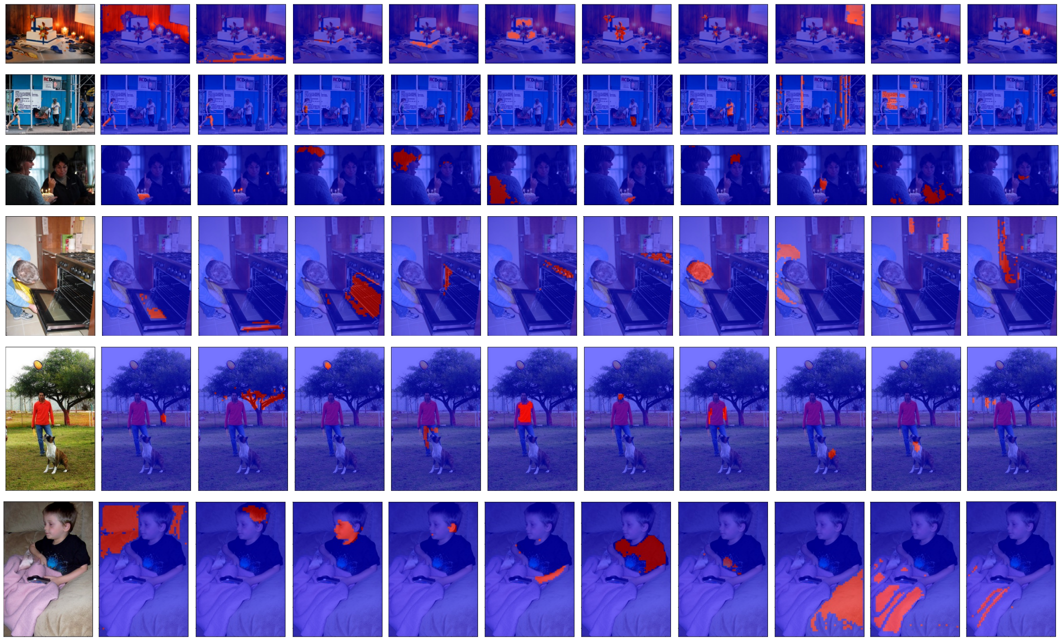


Figure A3: Visualization of tokens in the different clusters obtained from the third stage of ClusTR-S. The red area indicates the tokens belonging to the same cluster.

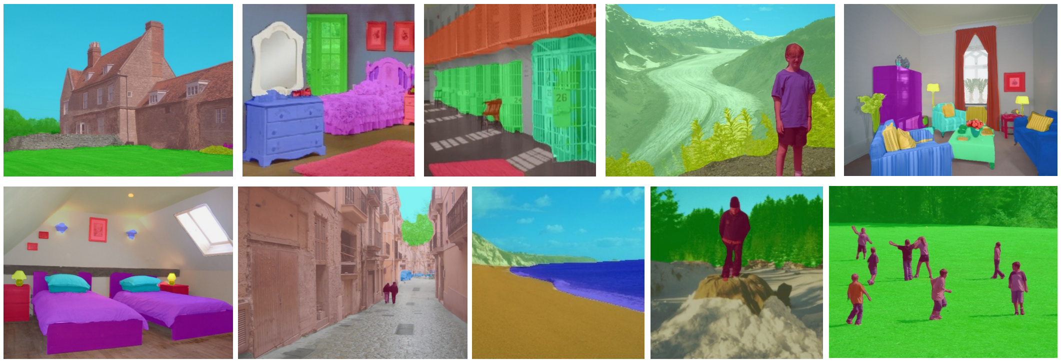


Figure A4: Semantic segmentation results of our ClusTR-S backbone on the ADE-20K set.

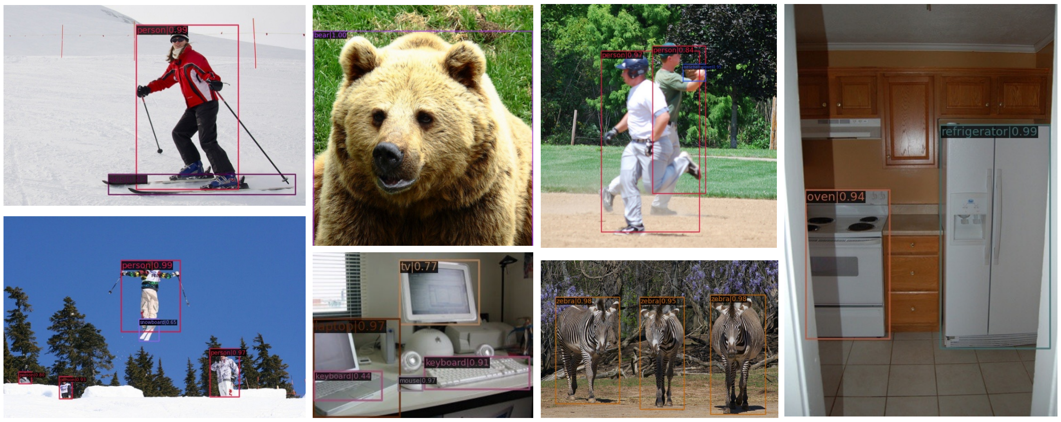


Figure A5: Object detection results of our ClusTR-S backbone on the COCO dataset.

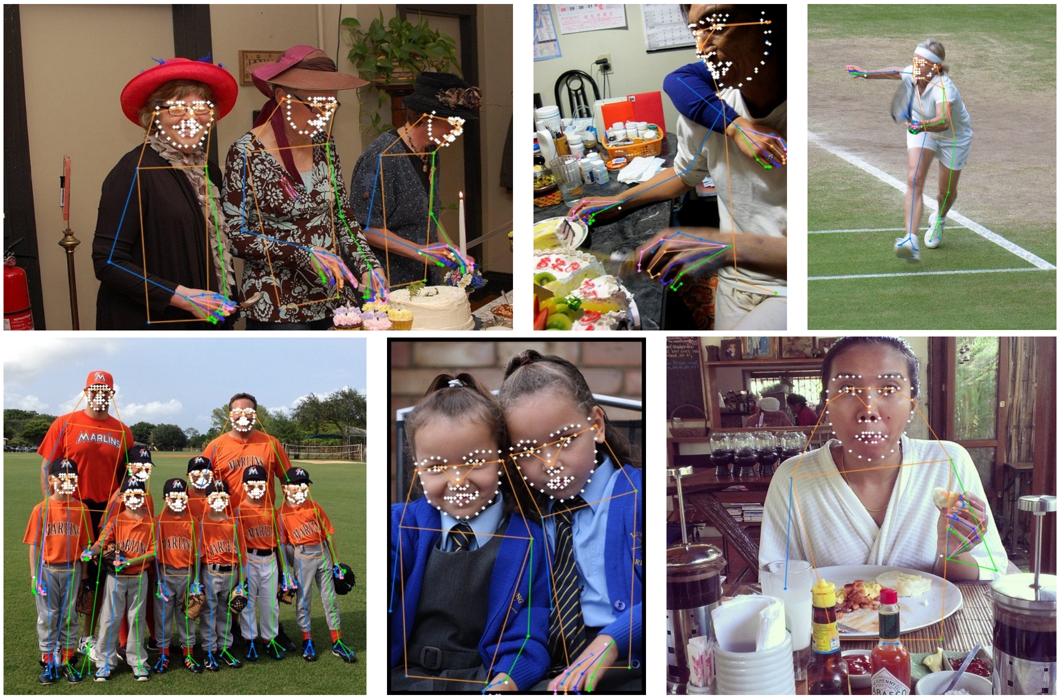


Figure A6: Pose estimation results of our ClusTR-S backbone on the COCOWholeBody V1.0 dataset.

REFERENCES

Kai Chen, Jiaqi Wang, Jiangmiao Pang, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

A. Kirillov, R. Girshick, K. He, and P. Dollar. Panoptic feature pyramid networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Wen Xie, Hao Li, and Rong Jin. Kvt: k-nn attention for boosting vision transformers. In *European conference on computer vision*, 2022a.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 568–578, October 2021.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022b.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018.