Noname manuscript No.
(will be inserted by the editor)

ViTAEv2: Vision Transformer Advanced by Exploring Inductive Bias for Image Recognition and Beyond

Qiming Zhang[1] · **Yufei Xu**[1] · **Jing Zhang**[1] · **Dacheng Tao**[2,1]

Received: date / Accepted: date

Abstract Vision transformers have shown great potential in various computer vision tasks owing to their strong capability to model long-range dependency using the self-attention mechanism. Nevertheless, they treat an image as a 1D sequence of visual tokens, lacking an intrinsic inductive bias (IB) in modeling local visual structures and dealing with scale variance, which is instead learned implicitly from large-scale training data with longer training schedules. In this paper, we propose a **Vi**sion **T**ransformer **A**dvanced by **E**xploring intrinsic IB from convolutions, *i.e.*, ***ViTAE***. Technically, ViTAE has several spatial pyramid reduction modules to downsample and embed the input image into tokens with rich multi-scale context using multiple convolutions with different dilation rates. In this way, it acquires an intrinsic scale invariance IB and can learn robust feature representation for objects at various scales. Moreover, in each transformer layer, ViTAE has a convolution block parallel to the multi-head self-attention module, whose features are fused and fed into the feed-forward network. Consequently, it has the intrinsic locality IB and is able to learn local features and global dependencies collaboratively. The proposed two kinds of cells are stacked in both isotropic and multi-stage manners to formulate two families of ViTAE models, i.e., the vanilla ViTAE and ViTAEv2. Experiments on the ImageNet dataset as well as downstream tasks on the MS COCO, ADE20K, and AP10K datasets validate the superiority of our models over the baseline transformer models and concurrent works. Besides, we scale up our ViTAE model to 644M parameters and obtain the state-of-the-art classification performance, i.e., 88.5% Top-1 classification accuracy on ImageNet validation set and the best 91.2% Top-1 classification accuracy on ImageNet real validation set, without using extra private data. It demonstrates that the introduced inductive bias still helps when the model size becomes large. Source code and pretrained models will be publicly available at code.

Keywords Vision transformer · Neural networks · Image classification · Object detection · Inductive bias

Qiming Zhang (qzha2506@uni.sydney.edu.au)
Yufei Xu (yuxu71166@uni.sydney.edu.au)
Jing Zhang (jing.zhang1@sydney.edu.au)
✉ Dacheng Tao (dacheng.tao@gmail.com)
[1] School of Computer Science, Faculty of Engineering, The University of Sydney, Darlington, NSW 2008, Australia.
[2] JD Explore Academy, China

1 Introduction

Transformers [74, 21] have become the popular frameworks in NLP studies owing to their strong ability in modeling long-range dependencies by the self-attention mechanism. Such success and good properties of transformers have inspired many following works that apply them in various computer vision tasks [22, 98, 76]. Among them, ViT [22] is the pioneering work that adapts a pure transformer model for vision by embedding images into a sequence of visual tokens and modeling the global dependencies among them with stacked transformer blocks. Although it achieves promising performance on image classification, it experiences a severe data-hungry issue, *i.e.*, requiring large-scale training data and a longer training schedule for better performance. One important reason is that ViT does not efficiently utilize the prior knowledge in vision tasks and lacks such inductive bias (IB) in modeling local visual clues (*e.g.*, edges and corners) and dealing with objects at various scales like convolutions. Alternatively, ViT has to learn such IB implicitly from large-scale data.

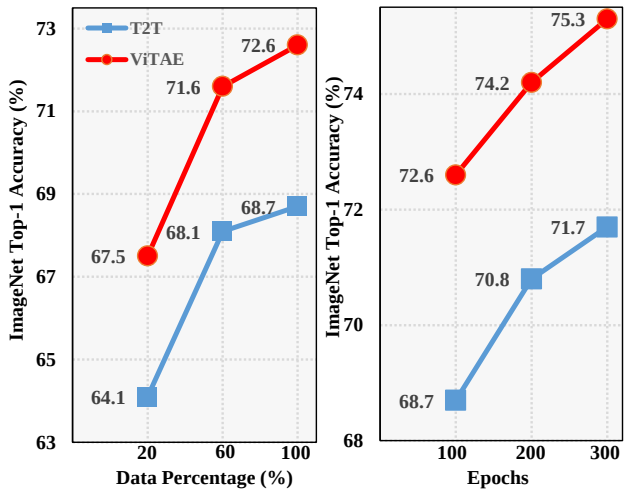

Fig. 1 Comparison of data and training efficiency of T2T-ViT-7 and ViTAE-T on ImageNet.

Unlike vision transformers, Convolution Neural Networks (CNNs) are naturally equipped with the intrinsic IBs of locality and scale-invariance and still serve as prevalent backbones in vision tasks [30,67,11,97]. The success of CNNs inspires us to explore the benefits of introducing intrinsic IBs in vision transformers. We start by analyzing the above two IBs of CNNs, *i.e.*, locality and scale-invariance. Convolution that computes local correlation among neighbor pixels is good at extracting local features such as edges and corners. Consequently, CNNs can provide great low-level features at the shallow layers [93], which are then aggregated into high-level features progressively by a bulk of sequential convolutions [34,66,68]. Moreover, CNNs have a hierarchy structure to extract multi-scale features at different layers [66,39,30]. Intra-layer convolutions can also learn features at different scales by varying their kernel sizes and dilation rates [29,67,11,45,97]. Consequently, scale-invariant feature representations can be obtained via intra- or inter-layer feature fusion. Nevertheless, CNNs are not well suited to model long-range dependencies[1], which is the key advantage of transformers. An interesting question then comes up: can we improve vision transformers by leveraging the good properties of CNNs? Recently, DeiT [72] explores the idea of distilling knowledge from CNNs to transformers to facilitate training and improve performance. However, it requires an off-the-shelf CNN model as the teacher and incurs extra training costs.

[1] Despite the projection layer in a transformer can be viewed as 1×1 convolution [12], the term of convolution here refers to those with larger kernels, *e.g.*, 3×3, which are widely used in typical CNNs to extract spatial features.

Different from DeiT, we explicitly introduce intrinsic IBs into vision transformers by re-designing the network structures in this paper. Current vision transformers always obtain tokens with single-scale context [22,92, 76,49] and learn to adapt to objects at different scales from data. For example, T2T-ViT [92] improves ViT by delicately generating tokens in a soft split manner. Specifically, it uses a series of Tokens-to-Token transformation layers to aggregate single-scale neighboring contextual information and progressively structures the image to tokens. Motivated by the success of CNNs in dealing with scale variance, we explore a similar design in transformers, *i.e.*, intra-layer convolutions with different receptive fields [67,89], to embed multi-scale context into tokens. Such a design allows tokens to carry useful features of objects at various scales, thereby naturally having the intrinsic scale-invariance IB and explicitly facilitating transformers to learn scale-invariant features more efficiently from data. On the other hand, low-level local features are fundamental elements to generate high-level discriminative features. Although transformers can also learn such features at shallow layers from data, they are not skilled as convolutions by design. Recently, [86,43,25] stack convolutions and attention layers sequentially and demonstrate that locality is a reasonable compensation of global dependency. However, this serial structure ignores the global context during locality modeling (and vice versa). To avoid such a dilemma, we follow the "divide-and-conquer" idea and propose modeling locality and long-range dependencies in parallel and fusing the features to account for both. In this way, we empower transformers to learn local and long-range features within each block more effectively.

Technically, we propose a new **Vi**sion **T**ransformers **A**dvanced by **E**xploring Intrinsic Inductive Bias (***ViTAE***), which is a combination of two types of basic cells, *i.e.*, reduction cell (RC) and normal cell (NC). RCs are used to downsample and embed the input images into tokens with rich multi-scale context, while NCs aim to jointly model locality and global dependencies in the token sequence. Moreover, these two types of cells share a simple basic structure, *i.e.*, paralleled attention module and convolutional layers followed by a feed-forward network (FFN). It is noteworthy that RC has an extra pyramid reduction module with atrous convolutions of different dilation rates to embed multi-scale context into tokens. Following the setting in [92], we stack three reduction cells to reduce the spatial resolution by 1/16 and a series of NCs to learn discriminative features from data. ViTAE outperforms representative vision transformers in terms of data efficiency and training efficiency (see Figure 1) as well as classification accuracy and generalization on downstream image classification

tasks. In addition, we further scale up ViTAE to large models and show that the inductive bias still helps to obtain better performance, e.g., ViTAE-H with 644M parameters achieves 88.5% Top-1 classification accuracy on ImageNet without using extra private data.

Beyond image classification, backbone networks should adapt well to various downstream tasks such as object detection, semantic segmentation, and pose estimation. To this end, we extend the vanilla ViTAE to the multi-stage design, *i.e.*, ViTAEv2. Specifically, a natural choice is to construct the model by re-arranging the reduction cells and normal cells according to the strategies in [76, 49] to have multi-scale feature outputs, *i.e.*, several consecutive NC cells are used following one RC module at each stage (feature resolution) rather than using a series of NCs only at the last stage. As a result, the multi-scale features from different stages can be utilized for those various downstream tasks. One remaining issue is that the vanilla attention operations in transformers have a quadratic computational complexity, requiring a large memory footprint and computation cost, especially for feature maps with a large resolution. To mitigate this issue, we further explore another inductive bias, *i.e.*, local window attention introduced in [49], in the RC and NC modules. Since the parallel convolution branch in the proposed two cells can encode position information and enable inter-window information exchange, special designs like the relative position encoding and window-shifting mechanism in [49] can be omitted. Consequently, our ViTAEv2 models outperform state-of-the-art methods for various vision tasks, including image classification, object detection, semantic segmentation, and pose estimation, while keeping a fast inference speed and reasonable memory footprint.

The contribution of this study is threefold. **First**, we explore two types of intrinsic IB in transformers, *i.e.*, scale invariance and locality, and demonstrate the effectiveness of this idea by designing a new transformer architecture named ViTAE based on two new reduction and normal cells that incorporate the above two IBs. ViTAE outperforms representative vision transformers regarding classification accuracy, data efficiency, training efficiency, and generalization on downstream vision tasks. **Second**, we scale up our ViTAE model to 644M parameters and obtain 88.5% Top-1 classification accuracy on ImageNet without using extra private data, which is better than the state-of-the-art Swin Transformer, demonstrating that the introduced inductive bias still helps when the model size becomes large. **Third**, we extend the vanilla ViTAE to the multi-stage design, *i.e.*, ViTAEv2. It learns multi-scale features at different stages efficiently while keeping a fast inference speed and reasonable memory footprint for large-size input images. Experiments on popular benchmarks demonstrate that it outperforms state-of-the-art methods for various downstream vision tasks, including image classification, object detection, semantic segmentation, and pose estimation.

The following of this paper is organized as follows. Section 2 describes the relevant works to our paper. We then detail the two basic cells, the vanilla ViTAE model, the scaling strategy for ViTAE, as well as the multi-stage design for ViTAEv2 in Section 3. Next, Section 4 presents the extensive experimental results and analysis. Finally, we conclude our paper in Section 5 and discuss the potential applications and future research directions.

2 Related Work

2.1 CNNs with intrinsic inductive bias

CNNs [39,93,30] have explored several inductive biases with specially designed operations and have led to a series of breakthroughs in vision tasks, such as image classification, object detection, and semantic segmentation. For example, following the fact that local pixels are more likely to be correlated in images [42], the convolution operations in CNNs extract features from the neighbor pixels within the receptive field determined by the kernel size [41]. By stacking convolution operations, CNNs have the inductive bias in modeling locality naturally.

In addition to the locality, another critical inductive bias in visual tasks is scale-invariance, where multi-scale features are needed to represent the objects at different scales effectively [52,88]. For example, to effectively learn features of large objects, a large receptive field is needed by either using large convolution kernels [88, 89] or a series of convolution layers in deeper architectures [30,34,66,68]. However, such operations may ignore the features of small objects. To construct multi-scale feature representation for objects at different scales effectively, various image pyramid techniques [11,1,55,6, 40,19] have been explored, where features are extracted from a pyramid of images at different resolutions respectively [44,11,53,62,35,4], either in a hand-crafted manner or learned manner. Accordingly, features from the small-scale images mainly encode the large objects, while features from the large-scale images respond more to small objects. Then, features extracted from different resolutions are fused to form the scale-invariant feature, *i.e.*, the inter-layer fusion. Another way to obtain the scale-invariant feature is to extract and aggregate multi-scale context by using multiple convolutions with different receptive fields in a parallel manner, *i.e.*,

the intra-layer fusion [97,68,67,67,69]. Either the inter-layer or intra-layer fusion empowers the CNNs with the scale-invariance inductive bias. It helps improve their performance in recognizing objects at different scales.

However, it is unclear whether these inductive biases can help the visual transformer to achieve better performance. This paper explores the possibility of introducing two types of inductive biases in the vision transformer, namely locality by introducing convolution in the vision transformer and scale-invariance by encoding a multi-scale contxt into each visual token using multiple convolutions with different dilation rates, following the convention of intra-layer fusion.

2.2 Vision transformers with inductive bias

ViT [22] is the pioneering work that applies a pure transformer to vision tasks and achieves promising results. It treats images as a 1D sequence, embeds them into several tokens, and then processes them by stacked transformer blocks to get the final prediction. However, since ViT simply treats images as 1D sequences and thus lacks inductive bias in modeling local visual structures, it indeed implicitly learns the IB from a large amount of data. Similar phenomena can also be observed in models with fewer inductive biases in their structures [71,23, 31].

To alleviate the data-hungry issue, the following works explicitly introduce inductive bias into vision transformers, *e.g.*, leveraging the IB from CNNs to facilitate the training of vision transformers with less training data or shorter training schedules. For example, DeiT [72] proposes to distill knowledge from pre-trained CNNs to transformers during training via an extra distillation token to imitate the behavior of CNNs. However, it requires an off-the-shelf CNN model as a teacher, introducing extra computation cost. Recently, some works try to introduce the intrinsic IB of CNNs into vision transformers explicitly [26,58,25,43,18,86, 80,91,9,49]. For example, [43,25,80,17] stack convolutions and attention layers sequentially, resulting in a serial structure and modeling the locality and global dependency accordingly. [76] design sequential multi-stage structures while [49] apply attention within local windows. However, these serial structures may ignore the global context during locality modeling (and vice versa). [77] establishes connection across different scales at the cost of heavy computation. To jointly model global and local context, Conformer [58] and MobileFormer [13] employ a model-parallel structure, consisting of parallel individual convolution and transformer branches and a complicated bridge connection between the two branches. Different from them, we follow the "divide-and-conquer" idea and propose to model locality and global dependencies simultaneously via a parallel structure within each transformer layer. In this way, the convolution and attention modules are designed to complement each other within the transformer block, which is more beneficial for the models to learn better features for both classification and dense prediction tasks.

2.3 Self supervised learning and model scaling

As demonstrated in previous studies, scaled-up models are naturally few-shot learners and beneficial to obtain better performance no matter in language, image, or cross-modal domains [21,94,60]. Recently, many efforts have been made to scale up vision models, *e.g.*, BiT [36] and EfficientNet [70] scale up the CNN models to hundreds of millions of parameters by employing wider and deeper networks, and obtain superior performance on many vision tasks. However, they need to train the scaled-up models with a much larger scale of private data, *i.e.*, JFT300M [36]. Similar phenomena can be observed when training the scaled-up vision transformer models for better performance [22,94].

However, it is not easy to gather such large amounts of labeled data to train the scaled-up models. On the other hand, self-supervised learning can help train scaled-up models using data without labels. For example, CLIP [60] adopts paired text and image data captured from the Internet and exploits the consistency between text and images to train a big transformer model, which obtains good performance on image and text generation tasks. [47] adopt masked language modeling (MLM) as pretext tasks and generate supervisory signals from the input data. Specifically, they take masked sentences with several words overrode with mask and predicted the masked words with the words from the sentence before masking as supervision. In this way, these models do not require additional labels for the training data and achieve superior performance on translation, sentiment analysis, *etc.* Inspired by the superior performance of MLM tasks in language, masked image modeling (MIM) tasks have been explored in vision tasks recently. For example, BEiT [3] tokenizes the images into visual tokens and randomly masks some tokens using a block-wise manner. The vision transformer model must predict the original tokens for those masked tokens. In this way, BEiT obtains superior classification and dense prediction performance using publicly available ImageNet-22K dataset [20]. MAE [27] simplifies the requirement of tokenizers and simply treats the image pixels as the targets for reconstruction. Using only ImageNet-1K training data, MAE obtains impressive performance. It is under-explored whether the vision transformers with

introduced inductive bias can be scaled up, *e.g.*, in a self-supervised setting. Besides, whether inductive bias can still help these scaled-up models achieve better performance remains unclear. In this paper, we make an attempt to answer this question by scaling up the ViTAE model and training it in a self-supervised manner. Experimental results confirm the value of introducing inductive bias in scaled-up vision transformers.

2.4 Comparison to the conference version

A preliminary version of this work was presented in [85]. This paper extends the previous study by introducing three major improvements.

1. We scale up the ViTAE model to different model sizes, including ViTAE-B, ViTAE-L, and ViTAE-H. With the help of inductive bias, the proposed ViTAE-H model with 644M parameters obtains the state-of-the-art classification performance, i.e., 88.5% Top-1 classification accuracy on ImageNet validation set and the best 91.2% Top-1 classification accuracy on ImageNet real validation set, without using extra private data. It demonstrates that the introduced inductive bias still helps when the model size becomes large. We also show the excellent few-shot learning ability of the scaled-up ViTAE models.
2. We extend the vanilla ViTAE to the multi-stage design and devise ViTAEv2. The efficiency of the RC and NC modules is also improved by exploring another inductive bias from local window attention. ViTAEv2 outperforms state-of-the-art models for image classification tasks as well as downstream vision tasks, including object detection, semantic segmentation, and pose estimation.
3. We also present more ablation studies and experiment analysis regarding module design, inference speed, memory footprint, and comparisons with the latest works.

3 Methodology

3.1 Revisit vision transformer

We first give a brief review of the vision transformer in this part. To adapt transformers to vision tasks, ViT [22] first splits an image $x \in R^{H\times W\times C}$ into several non-overlapping patches with the patch size p, and embeds them into visual tokens (*i.e.*, $x_t \in R^{N\times D}$) in a patch-to-token manner, where H, W, C denote the height, width, and channel dimensions of the input image respectively, N and D denote the token number and token dimension, respectively, and $N = (H \times W)/p^2$. Then, an extra learnable embedding with the same dimension D, considered as a class token, is concatenated to the visual tokens before adding position embeddings to all the tokens in an element-wise manner. In the following part of this paper, we use x_t to represent all tokens, and N is the total number of tokens after concatenation for simplicity unless specified. These tokens are fed into several sequential transformer layers for the final prediction. Each transformer layer is composed of two parts, *i.e.*, a multi-head self-attention module (MHSA) and a feed-forward network (FFN).

MHSA extends single-head self-attention (SHSA) by using different projection matrices for each head. In other words, MHSA is obtained after repeating SHSA for h times, where h is the number of heads. Specifically, for SHSA, the input tokens x_t are first projected to queries (Q), keys (K) and values (V) using three different projection matrices, *i.e.*, $Q, K, V = x_tW_Q, x_tQ_K$, x_tQ_V, where $W_{Q/K/V} \in R^{D\times \frac{D}{h}}$ denotes the projection matrix for query/key/value, respectively. Then, the self-attention operation is calculated as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D}})V, \tag{1}$$

where the output of each head is of size $R^{N\times \frac{D}{h}}$. Then the features of all the h heads are concatenated along the channel dimension and formulate the MHSA module's output.

FFN is placed on top of the MHSA module and applied to each token identically and separately. It consists of two linear transformations with an activation function in between. Besides, a layer normalization [2] and a shortcut are added before and aside from the MHSA and FFN, respectively.

3.2 The isotropic design of ViTAE

ViTAE aims to introduce the intrinsic IB in CNNs to vision transformers. As shown in Figure 2, ViTAE is composed of two types of cells, *i.e.*, RCs and NCs. RCs are responsible for downsampling while embedding multi-scale context and local information into tokens, and NCs are used to further model the locality and long-range dependencies in the tokens. Taken an image $x \in R^{H\times W\times C}$ as input, three RCs are used to gradually downsample x with a total of 16× ratio by 4×, 2×, and 2×, respectively. Thereby, the output tokens of the RCs after downsampling are of size $[H/16, W/16, D]$ where D is the token dimension (64 in our experiments). The output tokens of RCs are then flattened as $R^{(HW/256)\times D}$, concatenated with the class token (red in the figure),

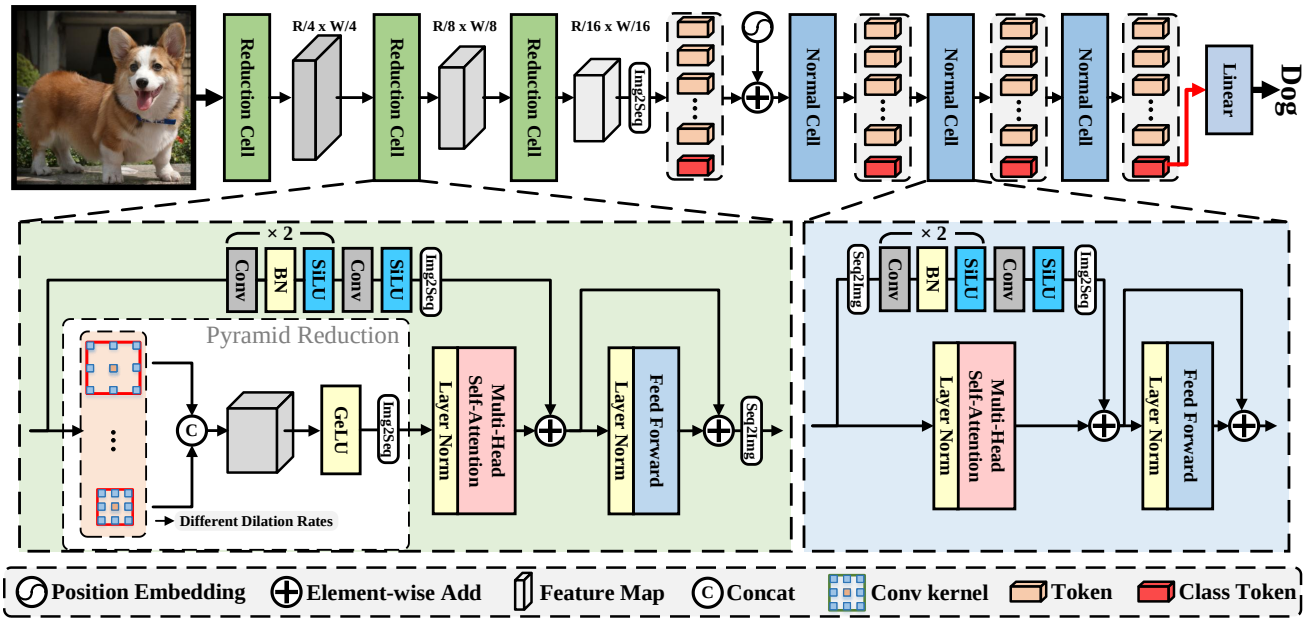

Fig. 2 The structure of the proposed ViTAE. It is constructed by stacking three RCs and several NCs. Both types of cells share a simple basic structure, *i.e.*, an MHSA module and a parallel convolutional module followed by an FFN. In particular, RC has an extra pyramid reduction module using atrous convolutions with different dilation rates to embed multi-scale context.

and added by the sinusoid position encoding. Next, the tokens are fed into the following NCs, which keep the length of the tokens. Finally, the prediction probability is obtained using a linear classification layer on the class token from the last NC.

3.2.1 Reduction cell

Instead of directly splitting and flattening images into visual tokens based on a linear image patch embedding layer, we devise the reduction cell to embed multi-scale context and local information into visual tokens, introducing the intrinsic scale-invariance and locality IBs from convolutions into ViTAE. Technically, RC has two parallel branches responsible for modeling locality and long-range dependency, followed by an FFN for feature transformation. We denote the input feature of the i_{th} RC as $f_i \in R^{H_i \times W_i \times D_i}$. The input of the first RC is the image x. In the global dependency branch, f_i is firstly fed into a Pyramid Reduction Module (PRM) to extract multi-scale context, *i.e.*,

$$\begin{aligned} f_i^{ms} &\triangleq PRM_i(f_i) \\ &= Cat([Conv_{ij}(f_i; s_{ij}, r_i) | s_{ij} \in \mathcal{S}_i, r_i \in \mathcal{R}]), \end{aligned} \tag{2}$$

where $Conv_{ij}(\cdot)$ indicates the j_{th} convolutional layer in the i_{th} PRM (i.e., $PRM_i(\cdot)$). It uses a dilation rate s_{ij} from the predefined dilation rate set $\mathcal{S}_i$ corresponding to the ith RC. Note that we use stride convolution to reduce the spatial dimension of features by a ratio r_i from the predefined reduction ratio set $\mathcal{R}$. The features after convolution are concatenated along the channel dimension, *i.e.*, $f_i^{ms} \in R^{(W_i/r_i)\times(H_i/r_i)\times(|\mathcal{S}_i|D_i)}$, where $|\mathcal{S}_i|$ denotes the number of dilation rates in the set $\mathcal{S}_i$. f_i^{ms} is then processed by an MHSA module to model long-range dependencies, *i.e.*,

$$f_i^g = MHSA_i(Img2Seq(f_i^{ms})), \tag{3}$$

where $Img2Seq(\cdot)$ is a simple reshape operation to flatten the feature map to a 1D sequence. In this way, f_i^g embeds the multi-scale context in each token. Note that the traditional MHSA individually attends each token at the same scale and thus lacks the ability to model the relationship between tokens at different scales. By contrast, the introduced multi-scale convolutions in reduction cells can (1) mitigate the information loss when merging tokens by looking at a larger field and (2) embed multi-scale information into tokens to aid the following MHSA to model the better global dependencies based on features at different scales.

In addition, we use a Parallel Convolutional Module (PCM) to embed local context within the tokens, which are fused with f_i^g as follows:

$$f_i^{lg} = f_i^g + PCM_i(f_i). \tag{4}$$

Here, $PCM_i(\cdot)$ represents the PCM of the i_{th} RC, which is composed of an $Img2Seq(\cdot)$ operation and three stacked convolution layers with BN layers and activation layers in between. It is noteworthy that the parallel convolution branch has the same spatial downsampling ratio as the PRM by using stride convolutions. In this

way, the token features can carry both local and multi-scale context, implying that RC acquires the locality IB and scale-invariance IB by design. The fused tokens are then processed by the FFN and reshaped back to feature maps, *i.e.*,

$$f_{i+1} = Seq2Img(FFN_i(f_i^{lg}) + f_i^{lg}), \tag{5}$$

where the $Seq2Img(\cdot)$ is a simple reshape operation to reshape a token sequence back to feature maps. $FFN_i(\cdot)$ represents the FFN in the i_{th} RC. In our ViTAE, three RCs are stacked sequentially to gradually reduce the input image's spatial dimension by 4×, 2×, and 2×, respectively. As the first RC handles images with high resolution, we adopt Performer [14] to reduce the computational burden and memory cost.

3.2.2 Normal cell

As shown in the bottom right part of Figure 2, NCs share a similar structure with the RC except for the absence of the PRM, which can provide rich spatial information via aggregating multi-scale information and compensate the spatial information loss caused by downsampling in RC. Given the features containing the multi-scale information, NCs are expected to focus on modeling long- and short-range dependency among the features. Besides, omitting the PRM module in NCs also helps to reduce the computational cost due to the large number of NCs in the stacked models. Therefore, we do not use PRM in NC. Specifically, given f_3 from the third RC, we first concatenate it with the class token t_{cls}, and then add it to the positional encodings to get the input tokens t for the following NCs. We ignore the subscript for clarity since all NCs have an identical architecture but different learnable weights. t_{cls} is randomly initialized at the start of training and fixed during the inference. Similar to the RC, the tokens are fed into the MHSA module, *i.e.*, $t_g = MHSA(t)$. Meanwhile, they are reshaped to 2D feature maps and fed into the PCM, *i.e.*, $t_l = Img2Seq(PCM(Seq2Img(t)))$. Note that the class token is discarded in PCM because it has no spatial connections with other visual tokens. To further reduce the parameters in NCs, we use group convolutions in PCM. The features from MHSA and PCM are then fused via element-wise sum, *i.e.*, $t_{lg} = t_g + t_l$. Finally, t_{lg} are fed into the FFN to get the output features of NC, *i.e.*, $t_{nc} = FFN(t_{lg}) + t_{lg}$. Similar to ViT [22], we apply layer normalization to the class token generated by the last NC and feed it to the classification head to get the final classification result.

3.3 Scaling up ViTAE via self-supervised learning

Except stacking the proposed RCs and NCs to construct the isotropic ViTAE models with 4M, 6M, 13M, and 24M parameters, we also scale up ViTAE to evaluate the benefit of introducing the inductive bias in vision transformers with large model sizes. Specifically, we follow the setting in ViT [22] to scale up the proposed ViTAE model, *i.e.*, we embed the image into visual tokens and process them using stacked NCs to extract features. The stacking strategy is exactly the same as the strategy adopted in ViT [22], where we use 12 NCs with 768 embedding dimensions to construct the ViTAE base model (*i.e.*, ViTAE-B with 89M parameters), 24 NCs with 1,024 embedding dimensions to construct the ViTAE large model (*i.e.*, ViTAE-L with 311M parameters), and 36 normal cells with 1,248 embedding dimension to construct the ViTAE huge model (*i.e.*, ViTAE-H with 644M parameters). The normal cells are stacked sequentially. However, the scaled-up models are easy to overfit if only trained using the ImageNet-1K dataset under a fully supervised training setting. Self-supervised learning [27], on the contrary, can eliminate this issue and facilitate the training of scaled-up models. In this paper, we adopt MAE [27] to train the scaled-up ViTAE model due to its simplicity and efficiency. Specifically, we first embed the input images into tokens and then randomly remove 75% of the tokens. The removed tokens are filled with randomly initialized mask tokens. After that, the remained visual tokens are processed by the ViTAE model for feature extraction. The extracted features and the mask tokens are then concatenated and fed into the decoder network to predict the values of the pixels belonging to the masked regions. The mean squared errors between the prediction and the masked pixels are minimized during the training.

However, as the encoder only processes the visual tokens, *i.e.*, the remained tokens after removing, the built-in locality property of images has been broken among the visual tokens. To adapt the proposed ViTAE model to the self-supervised task, we simply use convolution with kernel size 1×1 instead of 3×3 to formulate the ViTAE model for pretraining. This simple modification helps us to preserve a similar architecture between the network's pretraining and finetuning stage and helps the convolution branch to learn a meaningful initialization, as demonstrated in [95]. After the pretraining stage, we convert the kernels of the convolutions from 1×1 to 3×3 by zero-padding to recover the complete ViTAE models, which are further finetuned on the ImageNet-1k training data for 50 epochs. Inspired by [3], we use layer-wise learning rate decay during the finetuning to adapt the pre-trained models for specific vision tasks.

3.4 The multi-stage design for ViTAE

Apart from classification, other downstream tasks, including object detection, semantic segmentation, and pose estimation, are also very important that a general backbone should adapt to. These downstream tasks usually need to extract multi-level features from the backbone to deal with those objects at different scales. To this end, we extend the vanilla ViTAE model to the multi-stage design, *i.e.*, ViTAE-v2. A natural choice for the design of ViTAE-v2 can be re-constructing the model by re-organizing RCs and NCs. As shown in Figure 3, ViTAE-v2 has four stages where four corresponding RCs are used to gradually downsample the features by 4×, 2×, 2×, and 2×, respectively. At each stage, a number of N_i normal cells are sequentially stacked following the i_{th} RC. Note that a series of NCs are used only at the most coarse stage in the isotropic design. The number of normal cells, *i.e.*, N_i, controls the model depth and size. By doing so, ViTAE-v2 can extract a feature pyramid from different stages which can be used by the decoders specifically designed for various downstream tasks.

One remaining issue is that the vanilla attention operations in transformers have a quadratic computational complexity, therefore requiring a large memory footprint and computation cost, especially for feature maps with a large resolution. In contrast to the fast resolution reduction in the vanilla ViTAE design, we adopt a slow resolution reduction strategy in the multi-stage design, *e.g.*, the resolution of the feature maps at the first stage is only 1/4 of the original image size, thereby incurring more computational cost especially when the images in downstream tasks have high resolutions. To mitigate this issue, we further explore another inductive bias, *i.e.*, local window attention introduced in [49], in the RC and NC modules. Specifically, the window attention split the whole feature map into several non-overlap local windows and conducts the multi-head self-attention within each window, *i.e.*, each *query* token within the same window shares the same *key* and *value* sets. Since the parallel convolution branch in the proposed two cells can encode position information and enable inter-window information exchange, special designs like the relative position encoding and window-shifting mechanism in [49] can be omitted. We empirically find that replacing the full attention with local window attention at early stages can achieve a good trade-off between computational cost and performance. Therefore, we only use local window attention in the RC and NC modules at the first two stages. Consequently, our ViTAEv2 models can deliver superior performance for various vision tasks, including image classification, object detection, semantic segmentation, and pose estimation, while keeping a fast inference speed and reasonable memory footprint.

3.5 Model details

In this paper, we propose ViTAE and further extend it to the multi-stage version ViTAEv2 as described above. We devise several ViTAE and ViTAEv2 variants in our experiments to be compared with other models with similar model sizes. The details of them are summarized in Table 1. The 'dilation' column determines the dilation rate sets $\mathcal{S}$ in each RC. The two rows in the 'RC' and 'NC' columns denote the specific configurations of RCs and NCs, respectively, where 'P', 'W', 'F' refers to Performer [14], local window attention, and the vanilla full attention, respectively, and the number in the second rows denotes the number of heads in the corresponding attention module. The 'arrangement' column denotes the number of NC at each stage, while the 'embedding' denotes the token embedding size at each stage. Specifically, the default convolution kernel size in the first RC is 7×7 with a stride of 4 and dilation rates from $\mathcal{S}_1 = [1, 2, 3, 4]$. In the following two RCs (or three RCs for ViTAEv2), the convolution kernel size is 3×3 with a stride of 2 and dilation rates from $\mathcal{S}_2 = [1, 2, 3]$ and $\mathcal{S}_3 = [1, 2]$ (and $\mathcal{S}_4 = [1, 2]$ for ViTAEv2), respectively. Since the number of tokens decreases at later stages, there is no need to use large kernels and dilation rates at later stages. PCM in both RCs and NCs comprises three convolutional layers with a kernel size of 3×3.

4 Experiments

4.1 Implementation details

Unless explicitly stated, we train and test the proposed ViTAE and ViVTAEv2 model on the ImageNet-1k [39] dataset, which contains about 1.3 million images from 1k classes. The image size during training is set to 224×224. We use the AdamW [51] optimizer with the cosine learning rate scheduler and use the data augmentation strategy exactly the same as T2T [92] for a fair comparison regarding the training strategies and the size of models. We use a batch size of 512 for training ViTAE and 1024 for ViTAEv2. The learning rate is set to be proportion to 512 batch size with a base value 5e-4. The results of our models can be found in Table 2, where all the models are trained for 300 epochs. The models are built on PyTorch [57] and TIMM [79].

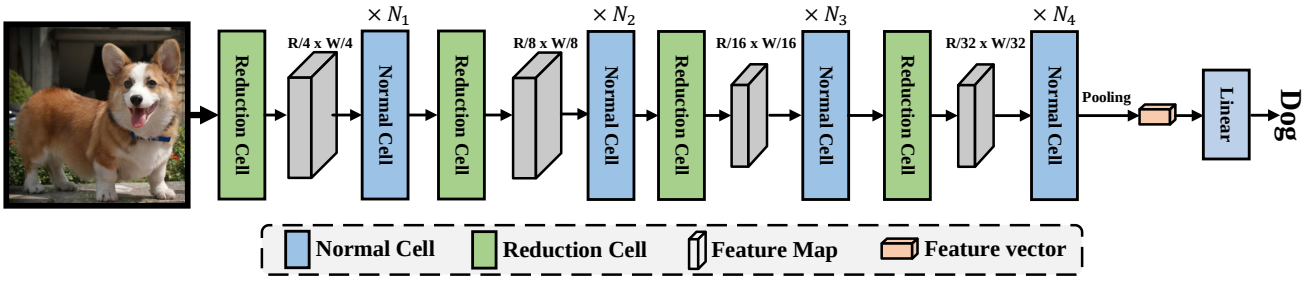

Fig. 3 The structure of the proposed multi-stage design ViTAEv2. The RCs and NCs are re-arranged in a stage-wise manner. At each stage, a number of NCs are sequentially stacked following each RC, which gradually downsamples the features by a certain ratio, *i.e.*, 4×, 2×, 2×, and 2×, respectively.

Table 1 Model details of ViTAE and ViTAEv2 variants. The two rows in the 'RC' and 'NC' columns denote the specific configurations of RCs and NCs, *i.e.*, the attention type and number of heads, respectively. '-' denotes there is no RC at the corresponding stage. 'arrangement' and 'embedding' denote the number of NCs and the token embedding size at each stage. 'P', 'T', 'W' represents performer, vanilla transformer, and local window attention, respectively.

Model	dilation	RC	NC	arrangement	embedding	Params (M)	Flops (G)
ViTAE-T	[1,2,3,4]	P, P, P 1, 1, 1	F 4	0, 0, 7	64, 64, 256	4.8	1.5
ViTAE-6M	[1,2,3,4]	P, P, P 1, 1, 1	F 4	0, 0, 10	64, 64, 256	6.5	2.0
ViTAE-13M	[1,2,3,4]	P, P, P 1, 1, 1	F 4	0, 0, 11	64, 64, 320	13.2	3.3
ViTAE-S	[1,2,3,4]	P, P, P 1, 1, 1	F 4	0, 0, 14	96, 192, 384	23.6	5.6
ViTAE-B	-	- -	F 12	0, 0, 12	768	89.3	36.9
ViTAE-L	-	- -	F 16	0, 0, 24	1024	311.7	125.8
ViTAE-H	-	- -	F 16	0, 0, 32	1280	644.3	335.7
ViTAEv2-S	[1,2,3,4] ↓	W, W, F, F 1, 1, 2, 4	W, W, F, F 1, 2, 4, 8	2, 2, 8, 2	64, 128, 256, 512	19.3	5.7
ViTAEv2-48M	[1,2,3,4] ↓	W, W, F, F 1, 1, 2, 4	W, W, F, F 1, 2, 4, 8	2, 2, 11, 2	96, 192, 384, 768	48.7	13.3
ViTAEv2-B	[1,2,3,4] ↓	W, W, F, F 1, 1, 2, 4	W, W, F, F 1, 2, 4, 8	2, 2, 12, 2	128, 256, 512, 1024	89.7	24.3

4.2 Comparison with the state-of-the-art

We compare our ViTAE and ViTAEv2 with both CNN models and vision transformers with similar model sizes in Table 2 and Table 3. Both Top-1/5 accuracy and real Top-1 accuracy [5] on the ImageNet validation set are reported. We categorize the methods into CNN models, vision transformers with learned IB, and vision transformers with introduced intrinsic IB. Compared with CNN models, our ViTAE-T achieves a 75.3% Top-1 accuracy, which is better than ResNet-18 with more parameters. The real Top-1 accuracy of the ViTAE model is 82.9%, which is comparable to ResNet-50 that has four more times of parameters than ours. Similar phenomena can also be observed when comparing ViTAE-T with MobileNetV1 [33] and MobileNetV2 [64], where ViTAE obtains better performance with fewer parameters. ViTAE-S achieves 82.0% Top-1 accuracy with half of the parameters of ResNet-101 and ResNet-152, showing the superiority of learning both local and long-range features from specific structures with corresponding intrinsic IBs by design. When adopting the multi-stage design, ViTAEv2-S further improves the Top-1 accuracy to 82.6% significantly. When finetuning the model using images of a larger resolution, *e.g.*, using 384×384 images as input, ViTAE-S's performance is further improved significantly by 1.2% absolute Top-1 accuracy. ViTAEv2-48M in Table 3 also benefits from it, and the performance increases from 83.8% to 84.7%, which further shows the potential of vision transformers with intrinsic IBs for large resolution images that are common in downstream dense prediction tasks. When the model size increases to 88M, ViTAEv2-B reaches 84.6% Top-1 accuracy, significantly outperforming other

Table 2 Comparison with SOTA methods. ↑ 384 denotes finetuning the model using images of 384×384 resolution.

Type	Model	Params (M)	FLOPs (G)	Input Size	ImageNet Top-1	ImageNet Top-5	Real Top-1
CNN	ResNet-18 [30]	11.7	1.8	224	70.3	86.7	77.3
	ResNet-50 [30]	25.6	3.8	224	76.7	93.3	82.5
	ResNet-101 [30]	44.5	7.6	224	78.3	94.1	83.7
	ResNet-152 [30]	60.2	11.3	224	78.9	94.4	84.1
	EfficientNet-B0 [70]	5.3	0.4	224	77.1	93.3	83.5
	EfficientNet-B4 [70]	19.3	4.2	380	82.9	96.4	88.0
	RegNetY-600M [61]	6.1	0.6	224	75.5	-	-
	RegNetY-4GF [61]	20.6	4.0	224	80.0	-	86.4
	RegNetY-8GF [61]	39.2	8.0	224	81.7	-	87.4
Transformer	DeiT-T [72]	5.7	1.3	224	72.2	91.1	80.6
	DeiT-T⚗ [72]	5.7	1.3	224	74.5	91.9	82.1
	PiT-Ti [32]	4.9	0.7	224	73.0	-	-
	T2T-ViT-7 [92]	4.3	1.2	224	71.7	90.9	79.7
	ViTAE-T	4.8	1.5	224	75.3	92.7	82.9
	CeiT-T [91]	6.4	1.2	224	76.4	93.4	83.6
	ConViT-Ti [18]	6.0	1.0	224	73.1	-	-
	CrossViT-Ti [9]	6.9	1.6	224	73.4	-	-
	ViTAE-6M	6.5	2.0	224	77.9	94.1	84.9
	PVT-T [76]	13.2	1.9	224	75.1	-	-
	ConViT-Ti+ [18]	10.0	2.0	224	76.7	-	-
	PiT-XS [32]	10.6	1.4	224	78.1	-	-
	ConT-M [86]	19.2	3.1	224	80.2	-	-
	ViTAE-13M	13.2	3.3	224	81.0	95.4	86.8
	DeiT-S [72]	22.1	4.6	224	79.9	95.0	85.7
	DeiT-S⚗ [72]	22.1	4.6	224	81.2	95.4	86.8
	PVT-S [76]	24.5	3.8	224	79.8	-	
	Conformer-Ti [58]	23.5	5.2	224	81.3	-	-
	Swin-T [49]	29.0	4.5	224	81.3	-	-
	CeiT-S [91]	24.2	4.5	224	82.0	95.9	87.3
	CvT-13 [80]	20.0	4.5	224	81.6	-	86.7
	ConViT-S [18]	27.0	5.4	224	81.3	-	-
	CrossViT-S [9]	26.7	5.6	224	81.0	-	-
	PiT-S [32]	23.5	4.8	224	80.9	-	-
	TNT-S [26]	23.8	5.2	224	81.3	95.6	-
	Twins-PCPVT-S[15]	24.1	3.8	224	81.2	-	-
	Twins-SVT-S [15]	24.0	2.9	224	81.7	-	-
	T2T-ViT-14 [92]	21.5	5.2	224	81.5	95.7	86.8
	ViTAE-S	23.6	5.6	224	82.0	95.9	87.0
	ViTAE-S ↑ 384	23.6	20.2	384	83.0	96.2	87.5
	ViTAEv2-S	19.2	5.7	224	82.6	96.2	87.6
	ViTAEv2-S ↑ 384	19.2	17.8	384	83.8	96.7	88.3

transformer models including Swin-B [49], Focal-B [87], and CrossFormer-B [77]. When finetuning using images of larger resolution or pretraining the model with ImageNet-22k, ViTAEv2-B's performance increases to 85.3% and 86.1% Top-1 accuracy, respectively, confirming the scalability of using IBs for large models and trained on large-scale datasets.

4.3 Analysis of the isotropic design of ViTAE

4.3.1 Data efficiency and training efficiency

To validate the effectiveness of introducing intrinsic IBs in improving data efficiency and training efficiency, we compare our ViTAE-T model with the baseline model T2T-ViT-7 at different training settings: (a) training them using 20%, 60%, and 100% ImageNet training set for equivalent 100 epochs regarding the full ImageNet training set, *e.g.*, we employ 5 times epochs when using 20% data for training compared with using 100% data; and (b) training them using the full ImageNet training set for 100, 200, and 300 epochs, respectively. The results are shown in Figure 1. As can be seen, ViTAE-T consistently outperforms the T2T-ViT-7 baseline by a large margin in terms of both data efficiency and training efficiency. For example, ViTAE-T using only 20% training data achieves comparable performance with T2T-ViT-7 using all data. When 60% training data are used, ViTAE-T significantly outperforms T2T-ViT-7 using all data by about an absolute 3% accuracy. It is also

Table 3 Comparison with SOTA methods (Table 2 continued). ↑ 384 denotes finetuning the model using images of 384×384 resolution, while * denotes the model pretrained using ImageNet-22k.

Type	Model	Params (M)	FLOPs (G)	Input Size	ImageNet Top-1	ImageNet Top-5	Real Top-1
Transformer	ViT-B/16 [22]	86.5	35.1	384	77.9	-	-
	ViT-L/16 [22]	304.3	122.9	384	76.5	-	-
	DeiT-B [72]	86.6	17.5	224	81.8	95.6	86.7
	PVT-M [76]	44.2	13.2	224	81.2	-	-
	PVT-L [76]	61.4	9.8	224	81.7	-	-
	Conformer-S [58]	37.7	10.6	224	83.4	-	-
	Swin-S [49]	50.0	8.7	224	83.0		
	ConT-B [86]	39.6	6.4	224	81.8	-	-
	CvT-21 [80]	32.0	7.2	224	82.5	-	87.2
	ConViT-S+ [18]	48.0	10.0	224	82.2	-	-
	ConViT-B [18]	86.0	17.0	224	82.4	-	-
	ConViT-B+ [18]	152.0	30.0	224	82.5	-	-
	PiT-B [32]	73.8	12.5	224	82.0	-	-
	TNT-B [26]	65.6	14.1	224	82.8	96.3	-
	T2T-ViT-19 [92]	39.2	8.9	224	81.9	95.7	86.9
	ViL-Base [96]	55.7	13.4	224	83.2	-	-
	ViTAEv2-48M	48.5	13.3	224	83.8	96.6	88.4
	ViTAEv2-48M ↑ 384	48.5	41.1	384	84.7	97.0	88.8
	Swin-B [49]	88.0	15.4	224	83.3	-	-
	Twins-SVT-L [15]	99.2	14.8	224	83.7	-	-
	PVTv2-B5 [75]	82.0	11.8	224	83.8	-	-
	Focal-B [87]	89.8	16.0	224	83.8	-	-
	DAT-B [81]	88.0	15.4	224	84.0	-	-
	CrossFormer [77]	92.0	16.1	224	84.0	-	-
	Conformer-B [58]	83.3	23.3	224	84.1	-	-
	ViTAEv2-B	89.7	24.3	224	84.6	96.9	88.7
	ViTAEv2-B ↑ 384	89.7	74.4	384	85.3	97.1	89.2
	ViTAEv2-B*	89.7	24.3	224	86.1	97.9	89.9

noteworthy that ViTAE-T trained for only 100 epochs has outperformed T2T-ViT-7 trained for 300 epochs. After training ViTAE-T for 300 epochs, its performance is significantly boosted to 75.3% Top-1 accuracy. With the proposed RCs and NCs, the transformer layers in our ViTAE only need to focus on modeling long-range dependencies, leaving the locality and multi-scale context modeling to its convolution counterparts, *i.e.*, PCM and PRM. Such a "divide-and-conquer" strategy facilitates ViTAE's training, making learning more efficient with less training data and fewer training epochs.

4.3.2 Generalization on downstream classification tasks

We further investigate the generalization of the proposed ViTAE models pre-trained on ImageNet-1k for downstream image classification tasks by fine-tuning them further on the training sets of several fine-grained classification tasks, including Flowers [54], Cars [37], Pets [56], and iNaturalist19. We also fine-tune the proposed ViTAE models pre-trained on ImageNet-1k further on Cifar10 [38] and Cifar100 [38]. The results are shown in Table 4. It can be seen that ViTAE achieves SOTA performance on most of the datasets using comparable or fewer parameters. These results demonstrate the good generalization ability of our ViTAE models.

4.3.3 Ablation study of the design of RC and NC

We use T2T-ViT [92] as our baseline model in the following ablation study of our ViTAE. As shown in Table 5, we investigate the hyper-parameter settings of RC and NC in the ViTAE-T model by isolating them separately. All the models are trained for 100 epochs on ImageNet-1k, following the same training setting and data augmentation strategy as described in Section 4.1.

We use ✓ and × to denote whether or not the corresponding module is enabled during the experiments. If all columns under the RC and NC are marked × as shown in the first row, the model becomes the standard T2T-ViT-7 model. "Pre" denotes the early fusion strategy that fuses output features of PCM and MHSA before FFN, while "Post" denotes a late fusion strategy alternatively. The ✓in "BN" denotes PCM uses BN. "×3" in the first column denotes that the dilation rate set is the same in the three RCs. "$[1, 2, 3, 4]\downarrow$" denotes using smaller dilation rates in deeper RCs, *i.e.*, $\mathcal{S}_1 = [1, 2, 3, 4]$, $\mathcal{S}_2 = [1, 2, 3]$, $\mathcal{S}_3 = [1, 2]$.

Table 4 Generalization of ViTAE and SOTA methods on different downstream image classification tasks.

Model	Params (M)	Cifar10	Cifar100	iNat19	Cars	Flowers	Pets
Grafit ResNet-50 [73]	25.6	-	-	75.9	92.5	98.2	-
EfficientNet-B5 [70]	30	98.1	91.1	-	-	98.5	-
ViT-B/16 [22]	86.5	98.1	87.1	-	-	89.5	93.8
ViT-L/16 [22]	304.3	97.9	86.4	-	-	89.7	93.6
DeiT-B [72]	86.6	99.1	90.8	77.7	92.1	98.4	-
T2T-ViT-14 [92]	21.5	98.3	88.4	-	-	-	-
ViTAE-T	4.8	97.3	86.0	73.3	89.5	97.5	92.6
ViTAE-S	23.6	98.8	90.8	76.0	91.4	97.8	94.2

Table 5 Ablation Study of RC and NC in ViTAE-T. "Pre" denotes the early fusion strategy that fuses output features of PCM and MHSA before FFN while "Post" denotes a late fusion strategy alternatively. The ✓in "BN" denotes PCM uses BN. "×3" in the first column denotes that the dilation rate set is the same in the three RCs. "$[1,2,3,4]\downarrow$" denotes using smaller dilation rates in deeper RCs, *i.e.*, $\mathcal{S}_1 = [1,2,3,4]$, $\mathcal{S}_2 = [1,2,3]$, $\mathcal{S}_3 = [1,2]$.

Reduction Cell		Normal Cell			Top-1
Dilation ($\mathcal{S}_1 \sim \mathcal{S}_3$)	PCM	Pre	Post	BN	
×	×	×	×	×	68.7
×	×	✓	×	×	69.1
×	×	×	✓	×	69.0
×	×	×	✓	✓	68.8
×	×	✓	×	✓	69.9
$[1,2]\times 3$	×	×	×	×	69.5
$[1,2,3]\times 3$	×	×	×	×	69.9
$[1,2,3,4]\times 3$	×	×	×	×	69.2
$[1,2,3,4,5]\times 3$	×	×	×	×	68.9
$[1,2,3,4]\downarrow$	×	×	×	×	69.8
$[1,2,3,4]\downarrow$	✓	×	×	×	71.7
$[1,2,3,4]\downarrow$	✓	✓	×	✓	72.6

As can be seen, using an early fusion strategy and BN in NC achieves the best 69.9% Top-1 accuracy among other settings. It is noteworthy that all the variants of NC outperform the vanilla T2T-ViT, implying the effectiveness of PCM, which introduces the intrinsic locality IB in transformers. It can also be observed that BN plays an important role in improving the model's performance as it can help to alleviate the scale deviation between convolution's and attention's features. For RC, we first investigate the influence of using different dilation rates in the PRM, as shown in the first column. As can be seen, using larger dilation rates (*e.g.*, 4 or 5) does not deliver better performance. We suspect that larger dilation rates may lead to plain features in the deeper RCs due to the smaller resolution of feature maps. To validate the hypothesis, we use smaller dilation rates in deeper RCs as denoted by $[1,2,3,4]\downarrow$. As can be seen, it achieves comparable performance as $[1,2,3]\times$. However, compared with $[1,2,3,4]\downarrow$, $[1,2,3]\times$ increases the amount of parameters from 4.35M to 4.6M. Therefore, we select $[1,2,3,4]\downarrow$ as the default setting. In addition, using PCM in the RC introduces the intrinsic locality IB and the performance increases to 71.7% Top-1 accuracy. Finally, the combination of RCs and NCs achieves the best accuracy at 72.6%, demonstrating their complementarity.

4.3.4 Visual inspection of ViTAE

To further analyze the property of our ViTAE, we apply Grad-CAM [65] on the MHSA's output in the last NC to qualitatively inspect ViTAE. The visualization results are provided in Figure 4. Compared with the baseline T2T-ViT, our ViTAE covers the single or multiple targets in the images more precisely and attends less to the background. Moreover, ViTAE can better handle the scale variance issue as shown in Figure 4(b). That is, it covers birds accurately whether they are small, medium, or large in size. Such observations demonstrate that introducing the intrinsic IBs of locality and scale-invariance from convolutions to transformers helps ViTAE learn more discriminate features than the pure transformers.

Besides, we calculate the average attention distance of each layer in ViTAE-T and the baseline T2T-ViT-7 on the ImageNet validation set, respectively. The results are shown in Figure 5. It can be observed that with the usage of PCM, which focuses on modeling locality, the transformer layers in the proposed NCs can better focus on modeling long-range dependencies, especially in shallow layers. In the deep layers, the average attention distances of ViTAE-T and T2T-ViT-7 are almost the same, where modeling long-range dependencies is much more important. It implies that the PCM does not affect the transformer's behavior in deep layers. These results confirm the effectiveness of the adopted "divide-and-conquer" idea in ViTAE, *i.e.*, introducing the intrinsic locality IB from convolutions into vision transformers makes it possible that transformer layers only need to be responsible to long-range dependencies since convolutions can well model locality in PCM.

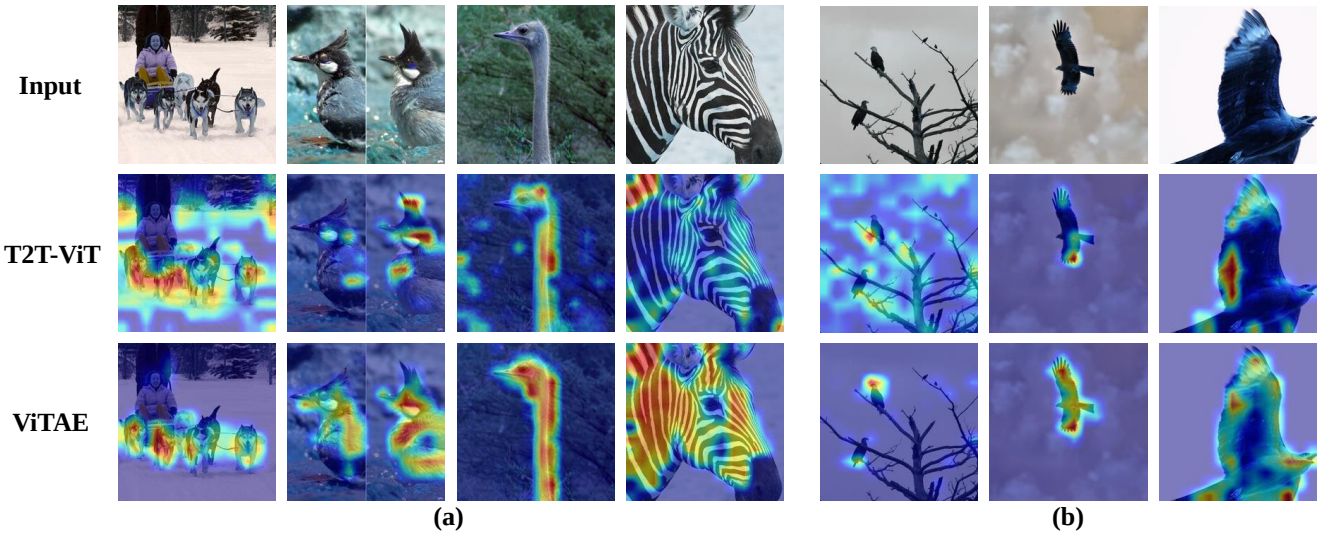

Fig. 4 Visual inspection of T2T-ViT-7 and ViTAE-T using Grad-CAM [65]. (a) Images containing multiple or single objects and the heatmaps obtained by T2T-ViT-7 and ViTAE-T. (b) Images containing the same class of objects at different scales and the heatmaps obtained by T2T-ViT-7 and ViTAE-T. Best viewed in color.

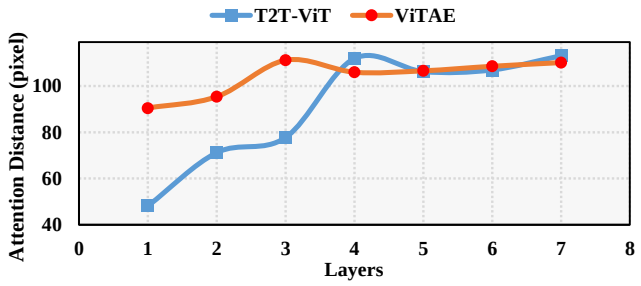

Fig. 5 The average per-layer attention distance of T2T-ViT-7 and our ViTAE-T on the ImageNet validation set.

4.4 Analysis of the scaled up ViTAE models

4.4.1 Image classification performance

We evaluate the performance of the scaled-up models on the ImageNet dataset. The scaled-up models are pre-trained for 1600 epochs using MAE [27], taking images from the ImageNet-1K training set. Then, the models are finetuned for 100 (the base model) or 50 (the large and huge model) epochs using the labeled data from the ImageNet-1K training set. It should be noted that the original MAE is trained on the TPU machines with Tensorflow, while our implementation adopts PyTorch as the framework and uses NVIDIA GPU for the training. This implementation difference may cause a slight performance difference in the models' classification accuracy. The results are summarized in Table 6. It demonstrates that the proposed ViTAE-B model with the introduced inductive bias outperforms the baseline ViT-B model by 0.4 Top-1 classification accuracy. For the ViTAE-L model with 300M parameters, the inductive bias still brings about 0.3 performance gains. After using ImageNet-22K labeled data for finetuning, the classification accuracy on the ImageNet-1K validation set further increases by about 1%. These results show that the benefit of introducing inductive bias in vision transformers is scalable to large models and

Table 6 The performance of scaled up ViTAE models on the ImageNet1K dataset. * denotes the results that we re-implement on GPU with PyTorch framework. † indicates that ImageNet22K are used to further finetune the models with 224×224 resolution for 90 epochs.

	#Params	Test size	Method	ImageNet Top-1	Real Top-1
T2TViT-24	65 M	224	Supervised [92]	82.3	87.2
ViT-B*	88 M	224	MAE [27]	83.4	89.1
ViTAE-B	89 M	224	MAE [27]	83.8	89.4
ViTAE-B†	89 M	224	MAE [27]	84.8	89.9
Swin-L†	197 M	384	Supervised [49]	87.3	90.0
SwinV2-L†	197 M	384	Supervised [48]	87.7	-
CoAtNet-4†	275 M	384	Supervised [17]	87.9	-
CvT-W24†	277 M	384	Supervised [80]	87.7	-
ViT-L*	304 M	224	MAE [27]	85.5	90.1
ViT-L	304 M	224	MaskFeat [78]	85.7	-
ViTAE-L	311 M	224	MAE [27]	86.0	90.3
ViTAE-L†	311 M	224	MAE [27]	87.5	90.8
ViTAE-L†	311 M	384	MAE [27]	88.3	91.1
SwinV2-H	658 M	224	SimMIM [84]	85.7	-
SwinV2-H	658 M	512	SimMIM [84]	87.1	-
ViTAE-H	644 M	224	MAE [27]	86.9	90.6
ViTAE-H	644 M	512	MAE [27]	87.8	91.2
ViTAE-H†	644 M	224	MAE [27]	88.0	90.7
ViTAE-H†	644 M	448	MAE [27]	88.5	90.8

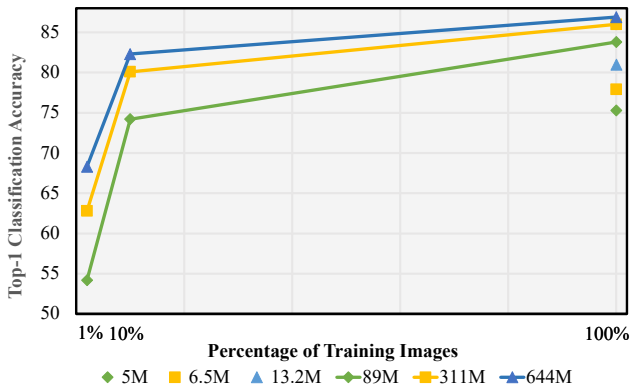

Fig. 6 The few-shot image classification results of scaled up ViTAE models. With only 10% data for finetuning, the scaled up ViTAE models obtain comparable or even better performance compared with the small ViTAE models ($<$ 20M) with 100% data for training.

datasets. Notably, our ViTAE-H, trained with only the ImageNet-1K dataset, obtains a classification accuracy of 91.2 on the ImageNet Real dataset [5], which is the highest accuracy we are aware of. It outperforms other methods trained with additional private data, such as EfficientNet [59] and ViT-G [94], where the former obtains 91.1 accuracy using the JFT300M dataset and the latter obtains 90.8 accuracy using the JFT3B dataset.

4.4.2 Few-shot learning performance

We further evaluate the data efficiency of scaled-up models by using different percentages of data to finetune the pre-trained models. We use 1%, 10%, and 100% data from the ImageNet-1k training set to finetune the self-supervised pre-trained ViTAE models with different amounts of parameters. We ensure that each model sees the same amount of the training images under different data settings, *i.e.*, we train the ViTAE-B model for 10,000 epochs using 1% training data, for 1,000 epochs using 10% training data, and for 100 epochs using 1% training data. Similarly, the ViTAE-L and ViTAE-H models are trained for 5,000 epochs using 1% training data and 500 epochs using 10% training data. The smaller models, *i.e.*, with less than 20M parameters, are trained from scratch using 100% ImageNet-1k training data for 300 epochs. As shown in Figure 6, the models with more parameters are more data-efficient than those with fewer parameters. For example, the ViTAE-H model with 644M parameters trained with 10% data outperforms the small model with 13.2 parameters trained with 100% data, *i.e.*, 82.4 v.s. 81.0. Such observations mirror the findings in previous studies, including both image classification and language modeling.

4.4.3 Ablation study of the convolutional kernel size in the scaled up ViTAE models

Table 7 The influence of the convolutional kernel size in the scaled up ViTAE models during pretraining.

	Kernel Size	#Params	Accuracy
ViT-L	0	304 M	84.1
ViTAE-L	0	311 M	84.1
ViTAE-L	3	311 M	84.4
ViTAE-L	1	311 M	84.6

We conduct experiments to investigate the influence of the convolutional kernel size in the scaled-up ViTAE models during pretraining. The results are presented in Table 7. The kernel size 0 represents that we do not use the convolution branch in ViTAE-L during pretraining, which degenerates to the original ViT-L model. Then, we add the convolution branches during finetuning and initialize the convolutional kernel weight as 0. If we use 1×1 kernels in ViTAE-L during pretraining, we pad them to 3×3 with zero padding during finetuning. We pretrain the models for 400 epochs and further finetune them for 50 epochs on the ImageNet-1K training set. As can be seen, using no convolution branch during pretraining leads to no improvement over the baseline ViT-L since those convolutional kernel weights in ViTAE-L during finetuning are zero-initialized. Directly pretraining and finetuning ViTAE-L with 3×3 convolutional kernels in the convolution branches leads to slightly better performance over the baseline ViT-L, but it is inferior to the proposed setting, *i.e.*, using 1×1 convolutional kernels in the convolution branches during pretraining ViTAE-L while zero-padding them to 3×3 during finetuning. We argue the reason is that most tokens (75%) during pretraining are randomly removed, and the remaining ones have lost spatial information. Therefore, using 3×3 kernels may lead to overfitting while 1×1 convolutions pay little attention to spatial structures and could learn better feature representation, which is in line with the observations in [95].

4.5 Analysis of the multi-stage design ViTAEv2

4.5.1 Ablation study of multi-stage design

In this paper, we extend ViTAE to a multi-stage design and propose ViTAEv2 accordingly. To achieve a good trade-off between classification performance and computational cost, we study the design choice of the attention type at each stage. The results are summarized in Table 8, where 'P', 'W', 'F' refer to the Performer

Table 8 Ablation study of the stage-wise ViTAE design. 'P', 'W', and 'F' represent using the Performer attention, window attention, and vanilla attention for each stage respectively. bs is the short name for batch size. We report the memory footprint and images throughout during training for comparison.

Block type		P, F, F, F	W, F, F, F	W, W, F, F	W, W, W, F	W, W, W, W
Params (M)		19.4	19.4	19.4	19.4	19.4
Top-1 Accuracy (224 img size)		82.2	82.7	82.6	82.2	82.2
img size 224, bs 128 per GPU	Memory (GB)	32.2	31.3	28.1	27.1	27.1
	Training speed (img/s)	1328.8	1300.8	1377.6	1333.6	1335.2
img size 448, bs 32 per GPU	Memory (GB)	40.5	40.5	32.3	26.8	26.7
	Training speed (img/s)	297.6	295.2	338.4	342.4	344.8
img size 672, bs 8 per GPU	Memory (GB)	38.7	38.8	23.7	16.3	16.0
	Training speed (img/s)	97.4	97.6	116.8	125.2	127.3
img size 896, bs 2 per GPU	Memory (GB)	34.4	34.3	15.0	9.5	9.2
	Training speed (img/s)	37.0	36.5	44.5	45.0	45.7

Table 9 Ablation study of the window shifting mechanism and relative position encoding in the local window attention in the ViTAEv2-S model.

shift	Y	Y	N	N
relative position enc.	Y	N	Y	N
Top-1 Accuracy	82.7	82.4	82.5	82.6
Memory (GB)	28.8	28.7	28.8	28.7
Training time (s/epoch)	486	477	481	472

attention, local window attention, and vanilla attention, respectively. They only differ in the implementation of attention calculation while having the same number of parameters. We list the Top-1 classification accuracy of different model variants trained from scratch using 224×224 images from the ImageNet-1k training set. We gradually increase the image resolution to compare the memory footprint and training speed of different models considering that the backbone models should well adapt to downstream vision tasks where large resolution images are common. Specifically, we set the batch size to 128 for all models for the 224×224 resolution and reduce it for larger resolutions to fit the A100 GPU memory.

We start from a baseline multi-stage design where the performer attention is used at the first stage while the vanilla full attention is used at the following three stages, denoting as 'P,F,F,F'. Then, we gradually introduce inductive bias into the model by replacing the performer and full attention with local window attention. As can be seen, all the models with introduced inductive bias outperform or are at least comparable to the baseline in terms of both classification performance and training cost. Specifically, using local window attention at the first two stages (*i.e.*, 'W,W,F,F') leads to the best trade-off between classification performance and computational cost for different image resolutions. Compared with the model with the best performance (*i.e.*, 'W,F,F,F'), its classification accuracy only drops by 0.1 while the memory footprint is significantly reduced by 56.4% at the setting 896×896 image resolution. Moreover, it outperforms the other two designs (*i.e.*, 'W,W,W,F' and 'W,W,W,W') by an absolute 0.4% Top-1 accuracy while having about the same training speed. Therefore, we choose the design of 'W, W, F, F' and devise the ViTAEv2 models at different model sizes accordingly.

One following interesting question is whether we still need the window-shifting mechanism to enable the inter-window information exchange and the relative position encoding (RPE) in the original implementation of local window attention proposed in [49] since the convolutional layers in PRM and PCM can enable inter-window information exchange and encode position information. We carry out an ablation study by isolating them one by one in our ViTAEv2 model to answer this question. We choose ViTAEv2-S as the base model, and all model variants are trained using 224×224 images. The results are summarized in Table 9. As can be seen, the above two components only contribute marginally in our ViTAE model, *i.e.*, about 0.1% accuracy. Therefore, we do not include them in our default design to make the model simple and easy to implement.

Table 10 Inference speed comparison of ViTAEv2.

	bs 128, size 224 (img/s)	bs 64, size 448 (img/s)	bs 16, size 896 (img/s)	Acc.
ViT-Small [72]	1459	318	48	79.9
ViT-Base [72]	803	167	25	81.8
T2T-ViT-14 [92]	996	220	33	81.2
T2T-ViT-24 [92]	575	118	17	82.3
Swin-T [49]	815	246	60	81.3
ViTAEv2-S	722	205	46	82.6

We also compare ViTAEv2 and some representative transformer models in terms of inference speed in Table 10. All the experiments are conducted on the same A100 GPU, and TensorRT is adopted to accelerate all models. As can be seen, our ViTAEv2-S model outperforms ViT-Small by 2.7% Top-1 accuracy while keeping a fast inference speed, especially for large size

Table 11 Object detection results on the MS COCO [46] validation set regarding different backbones. 'Schd' is short for training schedules.

Decoder	Schd	backbone	Venue	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	mAP^m_{75}	params (M)
Mask RCNN [28]	1x	ResNet-50 [30]	CVPR'16	38.0	58.6	41.4	34.4	55.1	36.7	44
		PVT-S [76]	ICCV'21	40.4	62.9	43.8	37.8	60.1	40.3	44
		Swin-T [49]	ICCV'21	43.7	66.6	47.7	39.8	63.3	42.7	47
		Focal-T [87]	NeurIPS'21	44.8	67.7	49.2	41.0	64.7	44.2	49
		PVTv2-B2 [75]	Arxiv'21	45.3	67.1	49.6	41.2	64.2	44.4	45
		RegionViT-B [10]	Arxiv'21	43.5	66.7	47.4	40.1	63.4	43.0	92
		Conformer-S/32 [58]	ICCV'21	43.6	-	-	39.7	-	-	58
		DAT-T [81]	Arxiv'22	44.4	67.6	48.5	40.4	64.2	43.1	48
		CrossFormer-S [77]	ICLR'21	45.4	68.0	49.7	41.4	64.8	44.6	50
		ViTAEv2-S	-	46.3	68.8	51.0	41.8	65.6	44.9	37
	3x	ResNet-50 [30]	CVPR'16	41.0	61.7	44.9	37.1	58.4	40.1	44
		PVT-S [76]	ICCV'21	43.0	65.3	46.9	39.9	62.5	42.8	44
		Swin-T [49]	ICCV'21	46.0	68.2	50.2	41.6	65.1	44.8	48
		MViT-T [24]	ICCV'21	45.9	68.7	50.5	42.1	66.0	45.4	46
		DAT-T [81]	Arxiv'22	47.1	69.2	51.6	42.4	66.1	45.5	48
		ViTAEv2-S	-	47.8	69.4	52.2	42.6	66.6	45.8	37
Cascade Mask RCNN [8]	1x	ResNet-50 [30]	CVPR'16	41.2	59.4	45.0	35.9	56.6	38.4	82
		Swin-T [49]	ICCV'21	48.1	67.1	52.2	41.7	64.4	45.0	86
		DAT-T [81]	Arxiv'22	49.1	68.2	52.9	42.5	65.4	45.8	86
		ViTAEv2-S	-	50.6	69.9	54.9	43.6	66.9	47.2	75
	3x	ResNet-50 [30]	CVPR'16	46.3	64.3	50.5	40.1	61.7	43.4	82
		Swin-T [49]	ICCV'21	50.4	69.2	54.7	43.7	66.6	47.3	86
		PVTv2-B2 [75]	Arxiv'21	51.1	69.8	55.3	-	-	-	83
		DAT-T [81]	Arxiv'22	51.3	70.1	55.8	44.5	67.5	48.1	86
		ViTAEv2-S	-	51.4	70.4	55.6	44.5	67.8	48.2	75

images, *e.g.*, 896×896. Compared with the state-of-the-art Swin transformer, the inference speed of ViTAEv2-S is slightly slower, *i.e.*, about 10%∼20%, but its classification performance is significantly improved by an absolute 1.3% Top-1 accuracy. ViTAEv2-S also outperforms T2T-ViT-24 in terms of both performance and inference speed.

We further evaluate the proposed ViTAEv2 models on representative downstream vision tasks, including object detection, semantic segmentation, and pose estimation. The results are detailed below.

4.5.2 The performance on object detection and instance segmentation

Settings To evaluate ViTAEv2's performance on object detection and instance segmentation tasks, we adopt Mask RCNN [28] and Cascade RCNN [7] as the detection framework and finetune the models on COCO 2017 dataset, which contains 118K training images, 5K validation images, and 20K test-dev images. We adopt exactly the same training setting used in Swin [49], *i.e.*, multi-scale training, AdamW optimizer [50] and the mmdetection code base. The models are trained for 12 (the 1x setting) and 36 epochs (the 3x setting), respectively. We compare the performance of ViTAEv2-S and other backbones, including the classic CNNs, *i.e.*, ResNet [30], and current transformer models.

Results The results are summarized in Table 11 and ViTAEv2-S achieves the best performance with the least number of parameters. Thanks to the introduced inductive bias like locality and scale invariance, the proposed ViTAEv2 model obtains 2.6 AP^b and 2.0 AP^m performance gains over Swin when using Mask RCNN as the decoder for the 1× setting. It also significantly outperforms other backbones like Conformer and CrossFormer, owning to our model's efficient divide-and-conquer structure design. When we extend the training schedule to the 3× setting (36 epochs in total), ViTAEv2 reaches 50.6 AP^b and 42.6 AP^m, significantly better than the other models. It is noteworthy that ViTAEv2 trained for 12 epochs has outperformed Swin-T trained for 36 epochs, validating the data efficiency of our model by introducing the inductive bias. The superiority of ViTAEv2 retains when using Cascade RCNN as the decoder, obtaining 50.6 AP^b and 43.6 AP^m when training 12 epochs and 51.4 AP^b and 44.5 AP^m when training 36 epochs. It can be concluded that introducing inductive bias into transformers helps our model better utilize the data and deliver the best performance for both object detection and instance segmentation.

4.5.3 The performance on semantic segmentation

Settings We evaluate the ViTAEv2's performance on the semantic segmentation task on the ADE20K [99,

Table 12 Semantic segmentation results on the ADE20k [99] validation set regarding different backbones. MS denotes that multi-scale inputs are used during testing.

backbone	Venue	mIoU	mIoU (MS)	params (M)
ResNet-50 [30]	CVPR'16	42.1	42.9	67
Swin-T [49]	ICCV'21	44.5	45.8	60
DAT-T [81]	Arxiv'22	45.5	46.4	60
ViTAEv2-S	-	45.0	48.0	49

100] dataset. The ADE20K dataset covers 150 semantic categories with 20K images for training and 2K for validation. We adopt UperNet [83] as the segmentation framework and train the UperNet with ViTAEv2-S as backbone with default setting used in mmsegmentation [16], *i.e.*, using the AdamW [50] optimizer and fixed image size 512×512. The models are trained for 160K iterations with a polynomial learning rate decay scheduler.

Results The results can be found in Table 12. With 10M fewer parameters, the segmentation model with ViTAEv2-S as the backbone obtains 45.0 mIoU and outperforms the counterparts using either ResNet or Swin transformer significantly. Besides, when tested with the multi-scale input, the segmentation model with ViTAEv2-S as the backbone obtains much better performance than others, *i.e.*, obtaining 48.0 mIoU. It implies that the ViTAE model can better extract the multi-scale feature owing to the introduced scale-invariance inductive bias. Therefore, it can be benefited more from the multi-scale input.

4.5.4 The performance on pose estimation

Table 13 Pose estimation results on the AP-10K [90] test set.

	ResNet [30]	Swin-T [49]	ViTAEv2-S
#Params	34.1 M	32.8 M	23.1 M
AP	0.681	0.689	0.718
AP_{50}	0.923	0.931	0.939
AP_{75}	0.718	0.751	0.786
AR	0.718	0.727	0.751
AR_{50}	0.933	0.939	0.947
AR_{75}	0.776	0.790	0.814

Settings We evaluate the models' performance on the animal pose estimation task on the AP10K [90] dataset. The AP-10K dataset contains 50 different animal species with animal keypoint annotations. Compared with human pose estimation tasks, animal pose estimation is more challenging due to the diverse species, less labeled data for each species, and significant appearance variance. Therefore, it is more suitable to evaluate the model's generalization ability on this task. Following the setting in AP-10K, we adopt SimpleBaseline [82] as the pose estimation framework and train the models with various backbones for 210 epochs using Adam optimizer and images of size 256×256. A step-wise learning rate decay scheduler is employed, and the learning rate is reduced by a factor of 10 after 170 and 200 epochs.

Results The results are summarized in Table 13. As can be seen, the proposed ViTAEv2-S model has fewer parameters yet brings an absolute 3% AP performance gain over the ResNet-50 backbone. Besides, it also outperforms the Swin-T [49] backbone, especially in the more strict evaluation metric, *i.e.*, AP_{75}. These results further demonstrate the superiority of the proposed ViTAEv2 model, which can better handle the tasks with limited data but rich categories, owing to the introduced inductive bias.

5 Discussions

This paper explores different types of IBs and incorporates them into transformers through the proposed reduction and normal cells. With the collaboration of these two cells, our ViTAE model achieves impressive performance on the ImageNet with fast convergence and high data efficiency. According to the attention distance analysis shown in Figure 5, the ensemble nature enables the transformer and convolution layers to focus on what they are good at, *i.e.*, modeling long-range dependencies and locality, respectively. As illustrated in Figure 2, our ViTAE model can be viewed as an intra-cell ensemble of complementary transformer layers and convolution layers owing to the skip connection and parallel structure. Moreover, the inductive bias also benefits the transformer models at larger model sizes, *i.e.*, ViTAE-H, or on larger datasets, *i.e.*, ImageNet-22K. Besides, we explore the multi-stage design of ViTAE models and propose ViTAEv2 accordingly, which obtains SOTA performance on image classification and downstream vision tasks, including object detection, semantic segmentation, and pose estimation. More kinds of IBs such as constituting viewpoint invariance [63] can be explored in the future study. On the other hand, although the proposed parallel structure obtains comparable inference speed with better performance, it may also slow down the training depending on the deep learning framework, *e.g.*, dynamic computation graph frameworks like PyTorch need to compute the parallel branches sequentially. Alternatively, static computation graph frameworks like TensorFlow can be adopted to mitigate this issue.

6 Conclusion

In this paper, we re-design the transformer block by proposing two basic cells (reduction cells and normal cells) to incorporate two types of intrinsic inductive bias (IB) into transformers, *i.e.*, locality and scale-invariance, resulting in a simple yet effective vision transformer architecture in both isotropic and multi-stage manner. Extensive experiments show that ViTAE outperforms representative vision transformers in various respects, including classification accuracy, data efficiency, and generalization ability on downstream tasks. When scaling to large-scale models, the inductive bias still helps in improving vision transformers' performance. In future work, we can explore other kinds of IBs to improve their performance further. We hope that this study will provide valuable insights to the following studies of introducing intrinsic IB into vision transformers and understanding the impact of intrinsic and learned IBs.

References

1. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA engineer **29**(6), 33–41 (1984)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Bao, H., Dong, L., Wei, F.: BEiT: BERT pre-training of image transformers (2021)
4. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision, pp. 404–417. Springer (2006)
5. Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., Oord, A.v.d.: Are we done with imagenet? arXiv preprint arXiv:2006.07159 (2020)
6. Burt, P.J., Adelson, E.H.: The laplacian pyramid as a compact image code. In: Readings in computer vision, pp. 671–679. Elsevier (1987)
7. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6154–6162 (2018)
8. Cai, Z., Vasconcelos, N.: Cascade r-cnn: High quality object detection and instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
9. Chen, C.F., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. arXiv preprint arXiv:2103.14899 (2021)
10. Chen, C.F., Panda, R., Fan, Q.: Regionvit: Regional-to-local attention for vision transformers. arXiv preprint arXiv:2106.02689 (2021)
11. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
12. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021)
13. Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., Liu, Z.: Mobile-former: Bridging mobilenet and transformer. arXiv preprint arXiv:2108.05895 (2021)
14. Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al.: Rethinking attention with performers. arXiv preprint arXiv:2009.14794 (2020)
15. 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)
16. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation` (2020)
17. Dai, Z., Liu, H., Le, Q.V., Tan, M.: Coatnet: Marrying convolution and attention for all data sizes. arXiv preprint arXiv:2106.04803 (2021)
18. 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)
19. Demirel, H., Anbarjafari, G.: Image resolution enhancement by using discrete and stationary wavelet decomposition. IEEE transactions on image processing **20**(5), 1458–1460 (2010)
20. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. Ieee (2009)
21. 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)
22. 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)
23. 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)
24. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. arXiv preprint arXiv:2104.11227 (2021)
25. 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)
26. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. arXiv preprint arXiv:2103.00112 (2021)
27. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)
28. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969 (2017)
29. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence **37**(9), 1904–1916 (2015)
30. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
31. He, L., Dong, Y., Wang, Y., Tao, D., Lin, Z.: Gauge equivariant transformer. In: Thirty-Fifth Conference on Neural Information Processing Systems (2021). URL `https://openreview.net/forum?id=fyL9HD-kImm`

32. 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)
33. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
34. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708 (2017)
35. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: Proceedings of the IEEE conference on computer vision and pattern recognition, vol. 2, pp. II–II. IEEE (2004)
36. Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N.: Big transfer (bit): General visual representation learning. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16, pp. 491–507. Springer (2020)
37. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013)
38. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
39. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)
40. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 624–632 (2017)
41. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
42. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks **3361**(10), 1995 (1995)
43. Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021)
44. Lin, G., Shen, C., Van Den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3194–3203 (2016)
45. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125 (2017)
46. 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)
47. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
48. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. arXiv preprint arXiv:2111.09883 (2021)
49. 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)
50. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
51. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
52. Luo, W., Li, Y., Urtasun, R., Zemel, R.S.: Understanding the effective receptive field in deep convolutional neural networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, vol. 29, pp. 4898–4906 (2016)
53. Ng, P.C., Henikoff, S.: Sift: Predicting amino acid changes that affect protein function. Nucleic acids research **31**(13), 3812–3814 (2003)
54. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conference on Computer Vision, Graphics and Image Processing (2008)
55. Olkkonen, H., Pesola, P.: Gaussian pyramid wavelet transform for multiresolution analysis of images. Graphical Models and Image Processing **58**(4), 394–398 (1996)
56. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
57. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32, pp. 8026–8037 (2019)
58. Peng, Z., Huang, W., Gu, S., Xie, L., Wang, Y., Jiao, J., Ye, Q.: Conformer: Local features coupling global representations for visual recognition. arXiv preprint arXiv:2105.03889 (2021)
59. Pham, H., Dai, Z., Xie, Q., Le, Q.V.: Meta pseudo labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11557–11568 (2021)
60. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021)
61. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing network design spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 10428–10436 (2020)
62. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Proceedings of the IEEE international conference on computer vision, pp. 2564–2571. Ieee (2011)
63. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. arXiv preprint arXiv:1710.09829 (2017)
64. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520 (2018)
65. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In:

Proceedings of the IEEE international conference on computer vision, pp. 618–626 (2017)
66. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
67. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
68. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
69. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826 (2016)
70. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114. PMLR (2019)
71. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., Dosovitskiy, A.: Mlp-mixer: An all-mlp architecture for vision. arXiv preprint arXiv:2105.01601 (2021)
72. 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)
73. Touvron, H., Sablayrolles, A., Douze, M., Cord, M., Jégou, H.: Grafit: Learning fine-grained image representations with coarse labels. arXiv preprint arXiv:2011.12982 (2020)
74. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, vol. 30, pp. 5998–6008 (2017)
75. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pvtv2: Improved baselines with pyramid vision transformer (2021)
76. 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)
77. Wang, W., Yao, L., Chen, L., Lin, B., Cai, D., He, X., Liu, W.: Crossformer: A versatile vision transformer hinging on cross-scale attention. arXiv preprint arXiv:2108.00154 (2021)
78. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. arXiv preprint arXiv:2112.09133 (2021)
79. Wightman, R.: Pytorch image models. `https://github.com/rwightman/pytorch-image-models` (2019). DOI 10.5281/zenodo.4414861
80. 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)
81. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention (2022)
82. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
83. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 418–434 (2018)
84. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. arXiv preprint arXiv:2111.09886 (2021)
85. Xu, Y., Zhang, Q., Zhang, J., Tao, D.: Vitae: Vision transformer advanced by exploring intrinsic inductive bias. arXiv preprint arXiv:2106.03348 (2021)
86. Yan, H., Li, Z., Li, W., Wang, C., Wu, M., Zhang, C.: Contnet: Why not use convolution and transformer at the same time? arXiv preprint arXiv:2104.13497 (2021)
87. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers (2021)
88. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR 2016 : International Conference on Learning Representations 2016 (2016)
89. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 472–480 (2017)
90. Yu, H., Xu, Y., Zhang, J., Zhao, W., Guan, Z., Tao, D.: Ap-10k: A benchmark for animal pose estimation in the wild. arXiv preprint arXiv:2108.12617 (2021)
91. Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W.: Incorporating convolution designs into visual transformers. arXiv preprint arXiv:2103.11816 (2021)
92. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. arXiv preprint arXiv:2101.11986 (2021)
93. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision, pp. 818–833. Springer (2014)
94. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers. arXiv preprint arXiv:2106.04560 (2021)
95. Zhang, J., Cao, Y., Wang, Y., Wen, C., Chen, C.W.: Fully point-wise convolutional neural network for modeling statistical regularities in natural images. In: Proceedings of the 26th ACM international conference on Multimedia, pp. 984–992 (2018)
96. 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)
97. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2881–2890 (2017)
98. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H.S., Zhang, L.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. arXiv preprint arXiv:2012.15840 (2020)
99. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 633–641 (2017)
100. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**(3), 302–321 (2019)