arXiv:2403.18361v2 [cs.CV] 28 Mar 2024

ViTAR: Vision Transformer with Any Resolution

Qihang Fan [1 2] **Quanzeng You** [3] **Xiaotian Han** [3] **Yongfei Liu** [3] **Yunzhe Tao** [3] **Huaibo Huang** [1] **Ran He** [1 2]
Hongxia Yang [3]

Abstract

This paper tackles a significant challenge faced by Vision Transformers (ViTs): their constrained scalability across different image resolutions. Typically, ViTs experience a performance decline when processing resolutions different from those seen during training. Our work introduces two key innovations to address this issue. Firstly, we propose a novel module for dynamic resolution adjustment, designed with a single Transformer block, specifically to achieve highly efficient incremental token integration. Secondly, we introduce fuzzy positional encoding in the Vision Transformer to provide consistent positional awareness across multiple resolutions, thereby preventing overfitting to any single training resolution. Our resulting model, ViTAR (Vision Transformer with Any Resolution), demonstrates impressive adaptability, achieving 83.3% top-1 accuracy at a 1120x1120 resolution and 80.4% accuracy at a 4032x4032 resolution, all while reducing computational costs. ViTAR also shows strong performance in downstream tasks such as instance and semantic segmentation and can easily combined with self-supervised learning techniques like Masked AutoEncoder. Our work provides a cost-effective solution for enhancing the resolution scalability of ViTs, paving the way for more versatile and efficient high-resolution image processing.

1. Introduction

The tremendous success of the Transformer in the field of Natural Language Processing (NLP) has spurred considerable exploration within the Computer Vision (CV) community (Vaswani et al., 2017; Dosovitskiy et al., 2021).

[1] MAIS & CRIPAC, Institute of Automa- tion, Chinese Academy of Sciences, China. [2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. [3]ByteDance, Inc.. Correspondence to: Qihang Fan <fanqihang.159@gmail.com>.

Figure 1. Comparison with other models: When the input resolution is larger than 1792, both DeiT-B and ResFormer-B encounter Out-of-Memory (OOM) errors. Annotations represent the models' computational load in terms of FLOPS. The results demonstrate that ViTAR possesses low computational overhead and exceptionally strong resolution generalization capability.

Specifically, Vision Transformers (ViTs) segment images into non-overlapping patches, project each patch into tokens, and then apply Multi-Head Self-Attention (MHSA) to capture the dependencies among different tokens. Thanks to Transformers' impressive modeling capabilities, ViTs achieve decent results in diverse visual tasks, including image classification (Liu et al., 2021; Hassani et al., 2023), object detection (Carion et al., 2020; Wang et al., 2021; 2022), vision-language modeling (ChaoJia et al., 2021; Radford et al., 2021), and even video recognition (Xing et al., 2023).

Despite achieving success in various domains, ViTs fall short in real-world scenarios that require handling variable input resolutions. Few studies have explored how to adapting ViTs to different resolutions (Touvron et al., 2021; Chu et al., 2023). Indeed, no training can cover all resolutions, a simple and widely used approach is to directly interpolate the positional encodings before feeding them into the ViT. However, this approach results in significant performance degradation in tasks such as image classification (Touvron et al., 2021; Tian et al., 2023; Ruoss et al., 2023). To tackle this problem, ResFormer (Tian et al., 2023) incorporates multiple resolution images during training. Additionally,

improvements are made to the positional encodings used by ViT, transforming them into more flexible, convolution-based positional encodings (Chu et al., 2023; Dong et al., 2022).

However, ResFormer still faces challenges. First, it can only maintain a high performance within a relatively narrow range of resolution variations, which is shown in Fig. 1. As the resolution increases, surpassing 892 or even higher, a noticeable decrease in the performance of the model becomes evident. Additionally, due to the use of convolution-based positional encoding, it becomes challenging to integrate ResFormer into widely adopted self-supervised frameworks like the Masked AutoEncoder (MAE) (He et al., 2022).

In this study, we propose the **Vi**sion **T**ransformer with **A**ny **R**esolution (ViTAR), a ViT which processes high-resolution images with low computational burden and exhibits strong resolution generalization capability. In ViTAR, we introduce the Adaptive Token Merger (ATM) module, which iteratively processes tokens that have undergone the patch embedding. ATM scatters all tokens onto grids. The process involves initially treating the tokens within a grid as a single unit. It then progressively merges tokens within each unit, ultimately mapping all tokens onto a grid of fixed shape. This procedure yields a collection of what are referred to as "grid tokens". Subsequently, this set of grid tokens undergoes feature extraction by a sequence of multiple Multi-Head Self-Attention modules. ATM module not only enhances the model's exceptional resolution adaptability but also leads to low computational complexity when dealing with high-resolution images. As shown in Fig. 1, our model generalizes better to unseen resolutions compared with DeiT (Touvron et al., 2021) and ResFormer (Tian et al., 2023). Moreover, as the input resolution increases, the computational cost associated with ViTAR is reduced to merely one-tenth or even less than that incurred by the conventional ViT.

In order to enable the model to generalize to arbitrary resolutions, we have also devised a method called Fuzzy Positional Encoding (FPE). The FPE introduces a certain degree of positional perturbation, transforming precise position perception into a fuzzy perception with random noise. This measure prevents the model from overfitting to position at specific resolutions, thereby enhancing the model's resolution adaptability. At the same time, FPE can be understood as a form of implicit data augmentation, allowing the model to learn more robust positional information and achieve better performance.

Our contributions can be summarized as follows:

- We propose a simple and effective multi-resolution adaptation module—Adaptive Token Merger, enabling our model to adapt to the requirements of multi-resolution inference. This module significantly improves the model's resolution generalization capability by adaptively merging input tokens and greatly reduces the computational load of the model under high-resolution inputs.
- We introduce a Fuzzy Positional Encoding that allows the model to perceive robust position information during training, rather than overfitting to a specific resolution. We transform the commonly used precise point perception of positional encoding into a Fuzzy range perception. This significantly enhances the model's adaptability to inputs of different resolutions.
- We conduct extensive experiments to validate the efficacy of our approach in multi-resolution inference. Our base model attains top-1 accuracies of 81.9, 83.4, and 80.4 under input resolutions of 224, 896, and 4032, respectively. Its robustness significantly surpasses that of existing ViT models. ViTAR also demonstrates robust performance in downstream tasks such as instance segmentation and semantic segmentation.

2. Related Works

Vision Transformers. The Vision Transformer (ViT) is a powerful visual architecture that demonstrates impressive performance on image classification, video recognition and vision-language learning (Dosovitskiy et al., 2021; Vaswani et al., 2017; Radford et al., 2021). Numerous efforts have been made to enhance ViT from the perspectives of data and computation efficiency (Yuan et al., 2022; Jiang et al., 2021; Guo et al., 2022; Fan et al., 2023; Huang et al., 2023; Ding et al., 2022). In these studies, the majority adapt the model to higher resolutions than those used during training through fine-tuning (Touvron et al., 2021; Si et al., 2022; Dong et al., 2022). There are few works attempting to directly accommodate unknown resolutions to the model without fine-tuning, which often results in a decline in performance (Liu et al., 2021; 2022). Fine-tuning on high resolutions often incur additional computational costs. Therefore, designing a visual model that can directly handle multiple resolutions becomes particularly important. However, this direction is still underexplored (Tian et al., 2023).

Multi-Resolution Inference. Investigating a single visual model that can perform inference across various resolutions remains a largely uncharted field. For the majority of visual models, if the resolution used during inference differs from the ones employed during training and direct inference is conducted without fine-tuning, a performance decline is observed (Liu et al., 2021; Touvron et al., 2021; Chu et al., 2023). As pioneering works in this field, NaViT (Dehghani et al., 2023) employs original resolution images as input to ViT, enabling ViT to better handle images with original

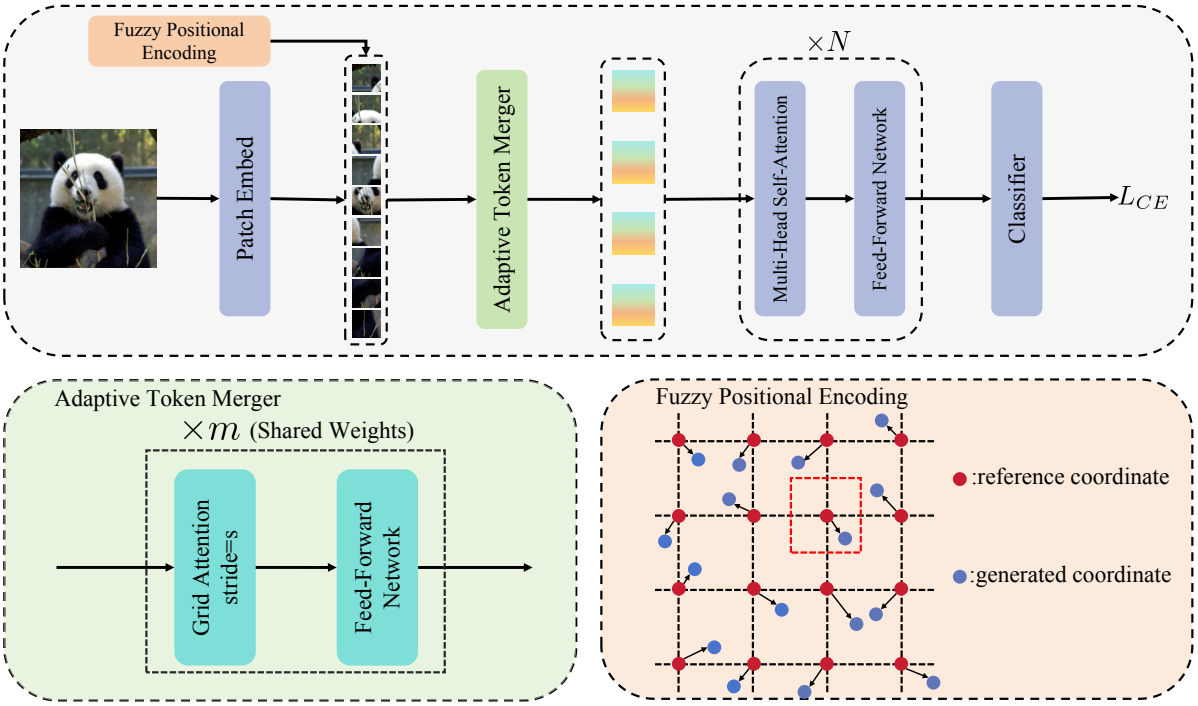

Figure 2. **Top**: Overall architecture of ViTAR consists of Adaptive Token Merger (ATM), Fuzzy Positional Encoding (FPE), and standard ViT self-attention layers. **Bottom**: *Left*: ATM with shared weights is applicable m times, depending on the input resolution. *Right*: FPE with the dashed red bounding box to denote the regions from which random points are sampled during training for positional embedding computation.

resolutions without the need for operations such as interpolation that may degrade the image quality. FlexiViT (Beyer et al., 2023), on the other hand, uses patches of multiple sizes to train the model, allowing the model to adapt to input resolutions by varying patch sizes. ResFormer (Tian et al., 2023) employs a method involving multi-resolution training, enabling the model to adapt to input images of various resolutions. It also incorporates several unique positional encodings, enhancing the model's ability to adapt to different resolutions. However, the positional encoding utilized by ResFormer is based on convolutional neural networks (Chu et al., 2023; Dong et al., 2022), and this configuration is challenging to be used in self-supervised learning frameworks like MAE (He et al., 2022). Additionally, ResFormer itself is based on the original ViT architecture, and when the input resolution increases, it incurs significant computational overhead. To enable the model to adapt to a broader range of resolutions and be applicable to commonly used self-supervised learning frameworks, further model optimization is necessary.

Positional Encodings. Positional encoding is crucial for ViT, often providing it with positional awareness and performance improvements. Earlier versions of ViT used sin-cos encoding to convey positional information, and some studies have demonstrated the limited resolution robustness of this positional encoding method (Touvron et al., 2021; Chu et al., 2023; Dosovitskiy et al., 2021). In contrast, convolution-based positional encoding demonstrates stronger resolution robustness. Models using convolutional positional encoding may even achieve performance improvements when faced with unseen resolutions (Chu et al., 2023; Tian et al., 2023; Dong et al., 2022). Unfortunately, convolutional positional encoding hinders the model to be used in self-supervised learning frameworks such as MAE (He et al., 2022). This makes it challenging for the model to be employed in the training of large-scale unlabeled datasets (ChaoJia et al., 2021).

3. Methods

3.1. Overall Architecture

The overall framework of our model is illustrated in Fig. 2, mainly comprising the Adaptive Token Merger (ATM), Fuzzy Positional Encodings (FPE), and the conventional ViT architecture. We do not employ a hierarchical structure; instead, we utilize a straight-through architecture similar to ResFormer (Tian et al., 2023) and DeiT (Touvron et al., 2021).

3.2. Adaptive Token Merger (ATM)

The ATM module receives tokens that have been processed through patch embedding as its input. We preset $G_h \times G_w$ to be the number of tokens we ultimately aim to obtain. ATM partitions tokens with the shape of $H \times W$ into a grid of size $G_{th} \times G_{tw}$. For ease of representation, we assume H is divisible by G_{th}, and W is divisible by G_{tw}. Thus, the number of tokens contained in each grid is $\frac{H}{G_{th}} \times \frac{W}{G_{tw}}$. In practical use, $\frac{H}{G_{th}}$ is typically set to 1 or 2, and the same applies to $\frac{W}{G_{tw}}$. Specifically, when $H \geq 2G_h$, we set $G_{th} = \frac{H}{2}$, resulting in $\frac{H}{G_{th}} = 2$. When $2G_h > H > G_h$, we pad H to $2G_h$ and set $G_{th} = G_h$, still maintaining $\frac{H}{G_{th}} = 2$. The padding method can be found in Appendix. When $H = G_h$, tokens on the edge of H are no longer fused, resulting in $\frac{H}{G_{th}} = 1$. For a specific grid, suppose its tokens are denoted as $\{x_{ij}\}$, where $0 \leq i < \frac{H}{G_{th}}$ and $0 \leq j < \frac{W}{G_{tw}}$. We perform average pooling on all $\{x_{ij}\}$ to obtain a mean token. Subsequently, using the mean token as the query, and all $\{x_{ij}\}$ as key and value, we employ cross-attention to merge all tokens within a grid into a single token. We name this process as GridAttention, which can be briefly described using Fig. 3.

Similar to the standard multi-head self-attention, our GridAttention also incorporates residual connections. To align the shapes of tokens, we employ residual connections with average pooling. The complete GridAttention is shown in Eq. 1.

$$\begin{aligned} x_{avg} &= \text{AvgPool}(\{x_{ij}\}) \\ \text{GridAttn}(\{x_{ij}\}) &= x_{avg} + \text{Attn}(x_{avg}, \{x_{ij}\}, \{x_{ij}\}) \end{aligned} \tag{1}$$

After passing through GridAttention, the fused token is fed into a standard Feed-Forward Network to complete channel fusion, thereby completing one iteration of merging token. GridAttention and FFN undergo multiple iterations and all iterations share the same weights. During these iterations, we gradually decrease the value of (G_{th}, G_{tw}), until $G_{th} = G_h$ and $G_{tw} = G_w$. Similar to the standard ViT, we typically set $G_h = G_w = 14$. This progressive token fusion approach significantly enhances our model's resolution adaptability and reduces the model's computational burden when the input resolution is large.

3.3. Fuzzy Positional Encoding

Many studies indicate that commonly used learnable positional encoding and sin-cos positional encoding are highly sensitive to changes in input resolution, and they fail to provide effective resolution adaptability (Chu et al., 2023; Tian et al., 2023). While convolution-based positional encoding exhibits better resolution robustness, its perception of adjacent tokens prevents its application in self-supervised learning frameworks like MAE (Chu et al., 2023; He et al., 2022).

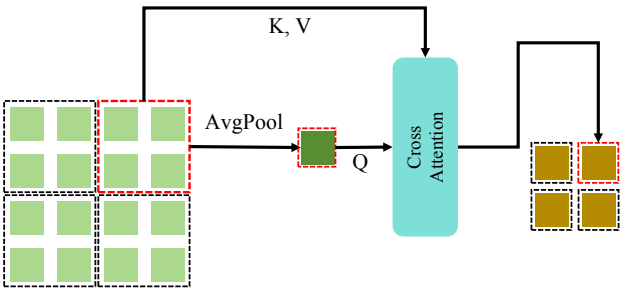

Figure 3. Illustration of GridAttention.

Our FPE differs from the aforementioned methods. While enhancing the model's resolution robustness, it does not introduce a specific spatial structure like convolution does. Therefore, it can be applied to self-supervised learning frameworks. This property enables ViTAR to be applied to large-scale, unlabeled training sets for training, aiming to obtain a more powerful vision foundation model.

In FPE, following ViT, we randomly initialize a set of learnable positional embeddings to serve as the model's positional encoding (Dosovitskiy et al., 2021). During the model training, typical positional encodings often provide precise location information to the model (Dosovitskiy et al., 2021; Touvron et al., 2021; Chu et al., 2023). In contrast, FPE only supplies the model with fuzzy positional information, as illustrated in Fig 2. The positional information provided by FPE undergos shifts within a certain range. Assuming the precise coordinates of the target token is (i, j), the positional coordinates provided by FPE during training are $(i + s_1, j + s_2)$, where $-0.5 \leq s_1, s_2 \leq 0.5$. Both s_1 and s_2 follow a uniform distribution. During the training process, we add randomly generated coordinate offsets to the reference coordinates. Based on the newly generated coordinates, we perform a grid sample on the learnable positional embeddings, resulting in the fuzzy positional encoding.

During inference, we no longer use the Fuzzy Positional Encoding but opt for the precise positional encoding. When there is a change in the input image resolution, we perform interpolation on the learnable positional embeddings. Due to the use of fuzzy positional encoding during the training phase, for any interpolated positional encoding, the model may have seen and used it somehow. Thus, the model gains robust positional resilience. As a result, during inference, when faced with inputs of unseen resolutions, the model continues to exhibit robust performance.

3.4. Multi-Resolution Training

Similar to ResFormer, when training ViTAR, we also employ a multi-resolution training approach. Our model, in

contrast to ResFormer (Tian et al., 2023), handles high-resolution images with significantly lower computational demands, enabling us to utilize a wider spectrum of resolutions during training. Unlike ResFormer, which processes batches containing inputs of various resolutions and employs KL loss for inter-resolution supervision, ViTAR processes each batch with inputs of a consistent resolution, relying solely on basic cross-entropy loss for supervision. Based on the multi-resolution training strategy, our model can be applied to a very broad range of resolutions and achieves favorable results in image classification tasks. Meanwhile, our model achieves similar performance to existing models in high-resolution input tasks (instance segmentation, semantic segmentation) with a much smaller computational cost. Specifically, in instance segmentation and semantic segmentation tasks that require high-resolution inputs, our model achieves results similar to ResFormer and DeiT with 50% of the FLOPs.

4. Experiments

We conduct extensive experiments on multiple vision tasks, such as image classification on ImageNet-1K (Deng et al., 2009), instance segmentation on COCO (Lin et al., 2014), and semantic segmentation on ADE20K (Zhou et al., 2017). We also train the model on the self-supervised framework MAE to verify the compatibility between ViTAR and MAE. After them, we make ablation studies to validate the importance of each component in ViTAR.

4.1. Image Classification

Settings. We train our models on ImageNet-1K (Deng et al., 2009) from scratch. We follow the training strategy in DeiT (Touvron et al., 2021) but without distillation loss for additional supervision. The only supervision for our model is classification loss. We use the AdamW optimizer with a cosine decay learning rate scheduler to train all of our models. We set the initial learning rate, weight decay, and batch size to 0.001, 0.05, and 1024, respectively. Like DeiT (Touvron et al., 2021), we adopt the strong data augmentation and regularization. The settings are RandAugment (Cubuk et al., 2020) (randm9-mstd0.5-inc1), Mixup (Zhang et al., 2018) (prob=0.8), CutMix (Yun et al., 2019) (prob=1.0), Random Erasing (Zhong et al., 2020) (prob=0.25). To stabilize the training process, we also use the layer decay (Dosovitskiy et al., 2021; He et al., 2022).

Results. The results achieved by the model on ImageNet are shown in Tab. 1. ViTAR demonstrates excellent classification accuracy across a considerable range of resolutions. For models of "S" size, when the resolution of input images exceeds 2240, traditional ViT architectures (DeiT and ResFormer) cannot perform inference due to computational resource limitations. In contrast, ViTAR is capable of inference with lower computational costs, maintaining accuracy at a high level (**2800: 52G, 81.3%**, **3360: 72G, 80.0%**). For models of size "B", ViTAR exhibits a similar pattern and adapts well to changes in the input image resolution. It demonstrates the best classification accuracy of **83.3%** when the input size is 1120 but only requires 1/20 FLOPs of ViT. Simultaneously, based on experimental results, our model exhibits improved resolution generalization capability as the size increases (at the resolution of 4032, the ViTAR-B achieves a 1.8% performance improvement compared to the ViTAR-S).

4.2. Object Detection

Settings. We adopt MMDetection (Chen et al., 2019) to implement Mask-RCNN (He et al., 2017) to validate our model's performance on object detection. The dataset we use is COCO (Lin et al., 2014). Our experimental settings follow the ResFormer (Tian et al., 2023) and ViTDet (Li et al., 2022). We use the commonly used $3\times$ training schedule to train the model. We apply AdamW optimizer with the initial learning rate of $1e-4$ and set the weight decay to $5e-2$. Following previous works (Li et al., 2022; Liu et al., 2021; Wang et al., 2021), we use the batch size of 16.

Results. In the tasks of object detection and instance segmentation, we do not utilize the multi-resolution training strategy. Thus, the ATM iterates only once.The $\frac{H}{G_{th}}$ and $\frac{W}{G_{tw}}$ in ATM are fixed to 1. The results, as shown in Tab. 2, indicate that our model achieves excellent performance in both object detection and instance segmentation. Specifically, the AP^b of ViTAR-B exceeds that of ResFormer-B by 0.7, and ViTAR-B outperforms ResFormer in the remaining metrics. Afterward, to reduce the computational burden of ViTAR, we set $\frac{H}{G_{th}}$ and $\frac{W}{G_{tw}}$ to 2, and compared ViTAR with the standard ViT, as shown in Tab. 3. Our ATM module reduces approximately 50% of the computational cost while maintaining high precision in dense predictions, demonstrating its effectiveness. These outcomes convincingly show that ATM has acquired the capability to adapt to resolutions of various sizes.

4.3. Semantic Segmentation

Settings. Follow the ResFormer, we implement the UperNet (Xiao et al., 2018) with the MMSegmentation (Contributors, 2020) to validate the performance of Our ViTAR. The dataset we use is ADE20K (Zhou et al., 2017). To train the UperNet, we follow the default settings in the Swin (Liu et al., 2021). We take AdamW as the optimizer to train the models for 80k/160k iterations.

Table 1. Comparison on the size "S" and size "B". Compared to DeiT and ResFormer, ViTAR can handle high-resolution input images with extremely low computational cost and exhibits strong resolution generalization capability.

Model	Resolution	FLOPs(G)	Top1-acc(%)	Model	Resolution	FLOPs(G)	Top1-acc(%)
DeiT-S	224	5	79.8	DeiT-B	224	18	81.8
ResFormer-S	224	5	**82.2**	ResFormer-B	224	18	**82.7**
ViTAR-S	224	5	80.3	ViTAR-B	224	19	81.9
DeiT-S	448	23	75.9	DeiT-B	448	79	79.8
ResFormer-S	448	23	82.5	ResFormer-B	448	79	82.9
ViTAR-S	448	6	**82.6**	ViTAR-B	448	23	**83.2**
DeiT-S	512	32	72.6	DeiT-B	512	107	78.2
ResFormer-S	512	32	82.0	ResFormer-B	512	107	82.6
ViTAR-S	512	7	**82.7**	ViTAR-B	512	26	83.2
DeiT-S	640	58	63.9	DeiT-B	640	184	74.2
ResFormer-S	640	58	80.7	ResFormer-B	640	184	81.7
ViTAR-S	640	7	**82.7**	ViTAR-B	640	29	**83.0**
DeiT-S	896	159	41.6	DeiT-B	896	450	64.2
ResFormer-S	896	159	70.8	ResFormer-B	896	450	75.5
ViTAR-S	896	9	**83.0**	ViTAR-B	896	37	**83.4**
DeiT-S	1120	327	16.4	DeiT-B	1120	863	50.1
ResFormer-S	1120	327	62.7	ResFormer-B	1120	863	68.6
ViTAR-S	1120	13	**83.1**	ViTAR-B	1120	49	**83.3**
DeiT-S	1792	1722	3.5	DeiT-B	1792	3976	29.6
ResFormer-S	1792	1722	30.8	ResFormer-B	1792	3976	45.5
ViTAR-S	1792	24	**82.8**	ViTAR-B	1792	93	**83.2**
DeiT-S	2240	3965	1.5	DeiT-B	2240	OOM	–
ResFormer-S	2240	3965	16.1	ResFormer-B	2240	OOM	–
ViTAR-S	2240	35	**82.3**	ViTAR-B	2240	137	**82.8**
DeiT-S	2800	OOM	–	DeiT-B	2800	OOM	–
ResFormer-S	2800	OOM	–	ResFormer-B	2800	OOM	–
ViTAR-S	2800	52	**81.3**	ViTAR-B	2800	203	**81.9**
DeiT-S	3360	OOM	–	DeiT-B	3360	OOM	–
ResFormer-S	3360	OOM	–	ResFormer-B	3360	OOM	–
ViTAR-S	3360	72	**80.0**	ViTAR-B	3360	282	**81.3**
DeiT-S	4032	OOM	–	DeiT-B	4032	OOM	–
ResFormer-S	4032	OOM	–	ResFormer-B	4032	OOM	–
ViTAR-S	4032	102	**78.6**	ViTAR-B	4032	399	**80.4**

Results. We report the results for segementation in Tab. 4 and Tab. 5. Like what we do in object detection, we first set $\frac{H}{G_{th}}$ and $\frac{W}{G_{tw}}$ of ATM to 1 to compare ViTAR with standard ViT models. As shown in Tab. 4, ViTAR shows better performance than ResFormer and other variants. Specifically, both our Small and Base models outperform ResFormer by 0.6 mIoU. After that, we set the ATM's $\frac{H}{G_{th}}$ and $\frac{W}{G_{tw}}$ to 2, the results are shown in Tab. 5. Our ATM reduces approximately 40% of the computational cost while still maintaining accuracy similar to the original model. This strongly demonstrates the effectiveness of the ATM module. The results of instance segmentation and semantic segmentation both indicate that our model can handle high-resolution images with relatively small computational cost, almost without affecting model performance.

4.4. Compatibility with Self-Supervised Learning

Settings. ResFormer employs convolution for positional encoding, making it difficult to be compatible with self-supervised learning frameworks like Mask AutoEncoder (MAE) (He et al., 2022), which disrupt the spatial structure of images. Since our model does not introduce spatial structures related to convolution, and our proposed Fuzzy Positional Encoding (FPE) does not require additional spatial information, it can be more conveniently integrated into the MAE. Unlike the standard MAE, we still employ a multi-resolution input strategy during training. We pretrain ViTAR-B for 300 epochs and fine-tune for an additional 100

Table 2. Results and comparisons of different backbones on the COCO2017 with the Mask R-CNN and 3× training schedule. The performances are evaluated by AP^b and AP^m.

Model	Params (M)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
PVT-Small	44	43.0	65.3	46.9	39.9	62.5	42.8
XCiT-S12/16	44	45.3	67.0	49.5	40.8	64.0	43.8
ViT-S	44	44.0	66.9	47.8	39.9	63.4	42.2
ViTDet-S	46	44.5	66.9	48.4	40.1	63.6	42.5
ResFormer-S	46	46.4	68.5	50.4	40.7	64.7	43.4
ViTAR-S	51	**46.9**	**69.0**	**50.7**	**41.2**	**64.9**	**43.8**
PVT-Large	81	44.5	66.0	48.3	40.7	63.4	43.7
XCiT-M24/16	101	46.7	68.2	51.1	42.0	65.6	44.9
ViT-B	114	45.8	68.2	50.1	41.3	65.1	44.4
ViTDet-B	121	46.3	68.6	50.5	41.6	65.3	44.5
ResFormer-B	115	47.6	69.0	52.0	41.9	65.9	44.4
ViTAR-B	123	**48.3**	**69.6**	**52.7**	**42.5**	**66.7**	**44.9**

Table 3. Results and comparison of FLOPs for different backbones. "*" indicates that we set $\frac{H}{G_{th}}$ and $\frac{W}{G_{tw}}$ of ATM to 2. FLOPs are measured with the input resolution of 800×1280.

Model	FLOPs (G)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ViTDet-B	812	46.3	68.6	50.5	41.6	65.3	44.5
ResFormer-B	823	47.6	69.0	52.0	41.9	65.9	44.4
ViTAR-B	836	**48.3**	**69.6**	**52.7**	**42.5**	**66.7**	**44.9**
ViTAR-B*	**421**	**47.9**	**69.2**	**52.2**	**42.1**	**66.2**	**44.5**

Table 4. Results and comparisons of different backbones on ADE20K. All backbones are pretrained on ImageNet-1k.

Model	Params (M)	Lr sched	mIoU_{ss}	mIoU_{ms}
DeiT-S	52	80k	43.0	43.8
Swin-T	60	160k	44.5	46.1
XCiT-S12/16	52	160	45.9	46.7
ResFormer-S	52	80k	46.3	47.5
ViTAR-S	57	80k	**47.0**	**48.1**
DeiT-B	121	160k	45.4	47.2
XCiT-S24/16	109	160k	47.7	48.6
ViT-B+MAE	177	160k	48.1	48.7
Swin-B	121	160k	48.1	49.7
ResFormer-B	120	160k	48.3	49.3
ViTAR-B	128	160k	**49.0**	**49.9**

epochs.

Results. We report the experimental results in Tab. 6. ViTAR, pre-trained for just 300 epochs, demonstrates a distinct advantage over a ViT model pre-trained for 1600 epochs. When the input resolution is increased, ViT+MAE exhibits a significant drop in performance. On the other hand, ViTAR+MAE demonstrates strong resolution robustness. Even with an input resolution exceeding 4000, the model still maintains high performance. The findings suggest that our method holds considerable promise for use in self-supervised learning frameworks, as exemplified by MAE (He et al., 2022). ViTAR's performance advantage over MAE may stem from two aspects. The first is that ATM enables the model to learn higher-quality tokens, providing the model with a portion of information gain. The second is that FPE, as a form of implicit data augmentation, allows the model to learn more robust positional information. As demonstrated in Droppos (Wang et al., 2023), the model's positional information is crucial for its learning process.

Table 5. Results and comparison of FLOPs for different backbones. "*" indicates that we set $\frac{H}{G_{th}}$ and $\frac{W}{G_{tw}}$ of ATM to 2. FLOPs are measured with the input resolution of 512×2048.

Model	FLOPs (G)	Lr sched	mIoU_{ss}	mIoU_{ms}
ResFormer-S	1099	80k	46.3	47.5
ViTAR-S	1156	80k	**47.0**	**48.1**
ViTAR-S*	**579**	80k	**46.4**	**47.5**
ViT-B+MAE	1283	160k	48.1	48.7
ResFormer-B	1286	160k	48.3	49.3
ViTAR-B	1296	160k	**49.0**	**49.9**
ViTAR-B*	**687**	160k	**48.6**	**49.4**

4.5. Ablation Study

Adaptive Token Merger. Adaptive Token Merging is a crucial module in our model that enables it to handle multi-resolution inputs. It plays a crucial role in the process of token fusion, effectively reducing the computational burden. Moreover, it enables the model to seamlessly adapt to a range of input resolutions. We compare two token fusion methods: ATM (Adaptive Token Merger) and AvgPool (adaptive avgpooling). For AvgPool, we partition all tokens into 14×14 grids, and each token within a grid undergoes avgpooling to merge the tokens in that grid. The number of tokens obtained after this fusion is the same as the number of tokens after processing with ATM. The results of the comparison are shown in Tab. 7, we use the ViTAR-S to conduct the ablation study. The results in Tab. 7 demonstrate that our ATM significantly enhances the performance and resolution adaptability of the model. Especially in high-resolution scenarios, the advantage of ATM becomes increasingly evident. Specifically, at a resolution of 4032, our proposed ATM achieves a 7.6% increase in accuracy compared with the baseline. At the resolution of 224, ATM also exhibits a

Table 6. Results with the MAE framework. The training resolution we used are (224, 448, 672, 896, 1120).

Model	Pre-Epochs	Testing resolution								
		224	384	448	640	1120	1792	2240	3360	4032
ViT-B+MAE	1600	**83.6**	81.0	77.5	66.1	38.6	13.4	OOM	OOM	OOM
ViTAR-B+MAE	300	82.5	**83.2**	**83.3**	**83.6**	**83.5**	**83.6**	**83.3**	**82.6**	**82.0**

performance gain of 0.5% compared with AvgPool.

Table 7. Ablation study of ATM. All experiments are conducted based on ViTAR-S.

Method	Resolution					
	224	448	896	1120	2240	4032
Avg	79.8	80.1	79.6	79.7	76.3	71.0
ATM	**80.3**	**82.6**	**83.0**	**83.1**	**82.3**	**78.6**

Fuzzy Positional Encoding. We compare the impact of different positional encodings on the model's resolution generalization ability. This includes commonly used sin-cos absolute position encoding (APE), conditional position encoding (CPE), global-local positional encoding (GLPE) in ResFormer, Relative Positional Bias (RPB) in Swin (Liu et al., 2021), and our proposed FPE. It is worth noting that only APE and FPE are compatible with the MAE framework. The other two positional encodings, due to the inherent spatial positional structure of convolution, are challenging to integrate into the MAE learning framework. We use the ViTAR-S to conduct the experiments for the models without MAE, and ViTAR-B for the models with MAE. The results of different positional encodings at various test resolutions are shown in Tab 8. It can be observed that our proposed FPE exhibits a significantly pronounced advantage in resolution generalization capability. Additionally, under the MAE self-supervised learning framework, FPE also demonstrates superior performance relative to APE, proving the potential applicability of FPE to a broader range of domains. Specifically, at the 4032 input resolution, FPE achieves a top-1 accuracy surpassing GLPE by 4.5%. In the MAE framework, FPE outperforms APE by 4.6%.

Training Resolutions. Unlike ResFormer, which only utilizes lower resolutions (128, 160, 224) during training, our model, due to its computational efficiency, can handle the inputs with very high resolutions. Additionally, employing a wider range of resolutions enhances the generalization capabilities of our model. In the previous experiments, we use resolutions of (224, 448, 672, 896, 1120) to train all models. In this section, we attempt to reduce the resolutions used during training to examine the model's resolution generalization capability. The experimental results, as shown in Tab. 9, reveal that within the range of resolutions used

Table 8. Comparison among different methods for positional encodings. Only APE and our FPE can be intergrated into the framework of MAE.

PE	Testing Resolution				
	224	448	1120	2240	4032
APE	80.0	81.5	81.2	76.1	70.6
CPE	80.4	81.8	81.6	78.2	72.0
RPB	80.3	81.6	81.2	75.8	69.8
GLPE	**80.8**	82.5	82.8	80.0	74.1
FPE	80.3	**82.6**	**83.1**	**82.3**	**78.6**
APE+MAE	82.0	82.5	82.5	80.6	77.4
FPE+MAE	**82.5**	**83.3**	**83.5**	**83.3**	**82.0**

in the experiments, the model's resolution generalization capability increases with the increment of resolutions used during training. Specifically, when our model is trained using these five resolutions (224, 448, 672, 896, 1120), the model exhibits the strongest resolution generalization capability. It achieves a 4.9% increase in accuracy at high resolution (4032) compared to training with only (224, 448). This strongly proves the effectiveness of multi-resolution training.

Table 9. Ablation of training resolutions. Using more resolutions during training significantly enhances the model's resolution generalization capability. All experiments are conducted based on the ViTAR-S.

Training Resolutions	Testing Resolutions				
	224	448	1120	2240	4032
(224, 448, 672, 896, 1120)	80.3	82.6	83.1	82.3	78.6
(224, 448, 672, 896)	80.4	82.6	83.1	82.1	78.0
(224, 448, 672)	80.6	82.6	83.3	81.6	76.8
(224, 448)	80.7	82.8	82.2	80.2	73.7

5. Conclusions

In this work, we propose a novel architecture: Vision Transformer with Any Resolution (ViTAR). The Adaptive Token Merger in ViTAR enables the model to adaptively handle variable-resolution image inputs, progressively merging tokens to a fixed size, greatly enhancing the model's resolution generalization capability, and reducing computational costs

when dealing with high-resolution inputs. Additionally, ViTAR incorporates Fuzzy Positional Encoding, allowing the model to learn robust positional information and handle high-resolution inputs not encountered during training. Our ViTAR is also compatible with existing MAE-based self-supervised learning frameworks, indicating its potential applicability to large-scale unlabeled datasets. In tasks such as instance segmentation and semantic segmentation that require high-resolution inputs, ViTAR significantly reduces computational costs with minimal performance loss. We hope this study can inspire subsequent research on the processing of high-resolution or variable-resolution images.

References

Beyer, L., Izmailov, P., Kolesnikov, A., Caron, M., Kornblith, S., Zhai, X., Minderer, M., Tschannen, M., Alabdulmohsin, I., and Pavetic, F. Flexivit: One model for all patch sizes. In *CVPR*, 2023.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *ECCV*, 2020.

ChaoJia, YinfeiYang, YeXia, Yi-TingChen, ZaranaParekh, Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

Chen, K., Wang, J., Pang, J., et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Chu, X., Tian, Z., Zhang, B., Wang, X., and Shen, C. Conditional positional encodings for vision transformers. In *ICLR*, 2023.

Contributors, M. Mmsegmentation, an open source semantic segmentation toolbox, 2020.

Cubuk, E. D., Zoph, B., Shlens, J., et al. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020.

Dehghani, M., Mustafa, B., Djolonga, J., et al. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. In *NeurIPS*, 2023.

Deng, J., Dong, W., Socher, R., et al. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Ding, M., Xiao, B., Codella, N., et al. Davit: Dual attention vision transformers. In *ECCV*, 2022.

Dong, X., Bao, J., Chen, D., et al. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Fan, Q., Huang, H., Zhou, X., and He, R. Lightweight vision transformer with bidirectional interaction. In *NeurIPS*, 2023.

Guo, J., Han, K., Wu, H., Xu, C., Tang, Y., Xu, C., and Wang, Y. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, 2022.

Hassani, A., Walton, S., Li, J., Li, S., and Shi, H. Neighborhood attention transformer. In *CVPR*, 2023.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. Mask r-cnn. In *ICCV*, 2017.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Huang, H., Zhou, X., Cao, J., He, R., and Tan, T. Vision transformer with super token sampling. In *CVPR*, 2023.

Jiang, Z.-H., Hou, Q., Yuan, L., Zhou, D., Shi, Y., Jin, X., Wang, A., and Feng, J. All tokens matter: Token labeling for training better vision transformers. In *NeurIPS*, 2021.

Li, Y., Mao, H., Girshick, R., and He, K. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022.

Lin, T.-Y., Maire, M., Belongie, S., et al. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., and Guo, B. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Ruoss, A., Delétang, G., Genewein, T., Grau-Moya, J., Csordás, R., Bennani, M., Legg, S., and Veness, J. Randomized positional encodings boost length generalization of transformers. In *ACL*, 2023.

Si, C., Yu, W., Zhou, P., Zhou, Y., Wang, X., and YAN, S. Inception transformer. In *NeurIPS*, 2022.

Tian, R., Wu, Z., Dai, Q., Hu, H., Qiao, Y., and Jiang, Y.-G. Resformer: Scaling vits with multi-resolution training. In *CVPR*, 2023.

Touvron, H., Cord, M., Douze, M., et al. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

Vaswani, A., Shazeer, N., Parmar, N., et al. Attention is all you need. In *NeurIPS*, 2017.

Wang, H., Fan, J., Wang, Y., Song, K., Wang, T., and Zhang, Z. Droppos: Pre-training vision transformers by reconstructing dropped positions. In *NeurIPS*, 2023.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., and Sun, J. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

Xing, Z., Dai, Q., Hu, H., Chen, J., Wu, Z., and Jiang, Y.-G. Svformer: Semi-supervised video transformer for action recognition. In *CVPR*, 2023.

Yuan, L., Hou, Q., Jiang, Z., Feng, J., and Yan, S. Volo: Vision outlooker for visual recognition. *TPAMI*, 2022.

Yun, S., Han, D., Oh, S. J., et al. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., et al. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Zhong, Z., Zheng, L., Kang, G., et al. Random erasing data augmentation. In *AAAI*, 2020.

Zhou, B., Zhao, H., Puig, X., et al. Scene parsing through ade20k dataset. In *CVPR*, 2017.

A. Grid Padding.

The commonly used padding approach involves surrounding the original tokens with an additional layer of padding tokens. However, in the context of ATM, employing this method might result in the outer grids containing only padding tokens, rendering the grid tokens ineffective. Therefore, as depicted in Fig. 4, we employ the grid padding method. Specifically, we pad with additional tokens on the right and bottom sides of each grid, ensuring a uniform distribution of padding tokens within each grid. This prevents the occurrence of ineffective grids.

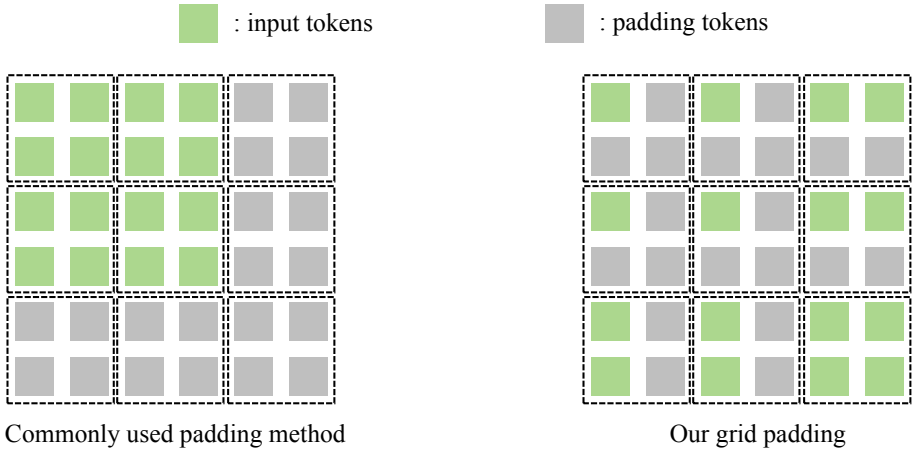

Figure 4. Illustration of grid padding.