GPViT: A High Resolution Non-Hierarchical Vision Transformer with Group Propagation

Anonymous authors
Paper under double-blind review

Abstract

We present the Group Propagation Vision Transformer (GPViT): a novel non-hierarchical (i.e. non-pyramidal) transformer model designed for general visual recognition with high-resolution features. High-resolution features (or tokens) are a natural fit for tasks that involve perceiving fine-grained details such as detection and segmentation, but exchanging global information between these features is expensive in memory and computation because of the way self-attention scales. We provide a highly efficient alternative Group Propagation Block (GP Block) to exchange global information. In each GP Block, features are first grouped together by a fixed number of learnable group tokens; we then perform *Group Propagation* where global information is exchanged between the grouped features; finally, global information in the updated grouped features is returned back to the image features through a transformer decoder. We evaluate GPViT on a variety of visual recognition tasks including image classification, semantic segmentation, object detection, and instance segmentation. Our method achieves significant performance gains over previous works across all tasks, especially on tasks that require high-resolution outputs, for example, our GPViT outperforms Swin Transformer-B by 2.0 mIoU on ADE20K semantic segmentation with only half as many parameters.

1 Introduction

Vision Transformer (ViT) architectures have achieved excellent results in general visual recognition tasks, outperforming ConvNets in many instances. In the original ViT architecture, image patches are passed through transformer encoder layers, each containing self-attention and MLP blocks. The spatial resolution of the image patches is constant throughout the network. Self-attention allows for information to be exchanged between patches across the whole image i.e. globally, however it is computationally expensive and does not place an emphasis on local information exchange between nearby patches, as a convolution would. Recent work has sought to build convolutional properties back into vision transformers (Liu et al., 2021; Wu et al., 2021a; Wang et al., 2021a) through a hierarchical (pyramidal) architecture. This design reduces computational cost, and improves ViT's performance on tasks such as detection and segmentation.

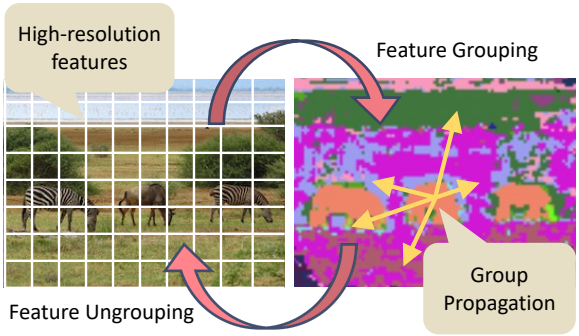


Figure 1: An illustration of our GP Block. It groups image features into a fixed-size feature set. Then, global information is efficiently propagated between the grouped features. Finally, the grouped features are queried by the image features to transfer this global information into them.

Is this design necessary for structured prediction? It incorporates additional inductive biases e.g. the assumption that nearby image tokens contains similar information, which contrasts with the motivation for ViTs in the first place. A recent study (Li et al., 2022a) demonstrates that a plain non-hierarchical ViT, a model that maintains the same feature resolution in all layers (non-pyramidal), can achieve comparable performance on object detection and segmentation tasks to a hierarchical counterpart. How do we go one step further and *surpass* these? One path would be to increase

feature resolution (i.e. the number of image tokens). A plain ViT with more tokens would maintain high-resolution features throughout the network as there is no downsampling. This would facilitate fine-grained, detailed outputs ideal for tasks such as object detection and segmentation. It also simplifies the design for downstream applications, removing the need to find a way to combine different scales of features in a hierarchical ViT. However, this brings new challenges in terms of computation. Self-attention has quadratic complexity in the number of image tokens. Doubling feature resolution (i.e. quadrupling the number of tokens) would lead to a $16\times$ increase in compute. How do we maintain global information exchange between image tokens without this huge increase in computational cost?

In this paper, we propose the **Group Propagation Vision Transformer (GPViT)**: a non-hierarchical ViT which uses high resolution features throughout, and allows for efficient global information exchange between image tokens. We design a novel Group Propagation Block (GP Block) for use in plain ViTs. Figure 1 provides a high-level illustration of how this block works. In detail, we use learnable group tokens and the cross-attention operation to group a large number of high-resolution image features into a fixed number of grouped features. Intuitively, we can view each group as a cluster of patches representing the same semantic concept. We then use an MLPMixer (Tolstikhin et al., 2021) module to update the grouped features and propagate global information among them. This process allows information exchange with a low computational cost, as the number of groups is much smaller than the number of image tokens. Finally, we ungroup the grouped features using a transformer decoder module where the updated grouped features act as key and value pairs, and are queried by the image token features. This updates the high resolution image token features with the group-propagated information. The GP Block only has a linear complexity in the number of image tokens, which allows it to scale better than ordinary self-attention. This block is the foundation of our simple non-hierarchical vision transformer architecture for general visual recognition.

We conduct experiments on multiple visual recognition tasks including image classification, object detection, instance segmentation, and semantic segmentation. We show significant improvements over previous approaches, including hierarchical vision transformers, under the same model size in all tasks. The performance gain is especially large for object detection and segmentation. For example, GPViT-L1 outperforms the Swin Transformer-S (Liu et al., 2021) by 3.3 AP^{bb} and 1.8^{mk} in COCO Mask R-CNN (He et al., 2017) object detection and instance segmentation with only 50% parameters, and GPViT-L3 outperforms Swin Transformer-B by 2.0 mIoU on UperNet (Xiao et al., 2018) ADE20K semantic segmentation also with only 50% parameters.

2 Related Work

Vision Transformers. Vision Transformers have shown great success in visual recognition. They have fewer inductive biases, e.g. translating invariance, scale-invariance, and feature locality (Xu et al., 2021b) than ConvNets and can better capture long-range relationships between image pixels. In the original ViT architecture (Dosovitskiy et al., 2021; Touvron et al., 2020), images are split into patches and are transformed into tokens that are passed through the encoder of a transformer (Vaswani et al., 2017). Based on this framework, LeViT (Graham et al., 2021) achieves a significant performance improvement over ViT by combining convolutional and transformer encoder layers. An important development in ViT architectures is the incorporation of a hierarchical feature pyramid structure, as typically seen in ConvNets (Wang et al., 2021a; Liu et al., 2021; Xu et al., 2021a; Wu et al., 2021b; Fan et al., 2021). For example, Liu et al. (2021) propose a shifted windowing scheme to efficiently propagate feature information in the hierarchical ViT. Such a pyramid architecture provides multi-scale features for a wide range of visual recognition tasks. Following this line of research, recent developments have been focusing on studying the usage of the hierarchical features in ViT (Ren et al., 2022b; Guo et al., 2022; Li et al., 2022b; Dong et al., 2022; Hatamizadeh et al., 2022; Chen et al., 2022a; d'Ascoli et al., 2021; Lee et al., 2022). For example, Ren et al. (2022b) introduce using multi-resolution features as attention keys and values to make the model learn better multi-scale information. While this is encouraging, it introduces extra complexity in the downstream model's design on how to utilize the multi-scale features effectively. Recently, Li et al. (2022a) propose to revisit the plain non-hierarchical ViT for visual recognition, which simplifies the usage of features and better decouples the pre-training and downstream model design stages. Our work extends this line of effort on how to efficiently increase the feature resolution in a non-hierarchical ViT.

Attention Mechanisms in ViTs. One bottleneck in using high resolution features in ViT is the quadratic complexity of the self-attention layer's computation. To tackle this challenge, a local

attention mechanism is proposed (Liu et al., 2021; Huang et al., 2019; Dong et al., 2022; Xu et al., 2021a; Zhang et al., 2022; Han et al., 2021) to allow each image token to attend to local regions instead of the whole image. However, using only local attention will hinder a model's ability to exchange information globally. To counter this problem, RegionViT (Chen et al., 2022a) and GCViT (Hatamizadeh et al., 2022) first down-sample their feature maps and exchange global information between the down-sampled features, then they use self-attention to transfer information between the original image features and the down-sampled features. This is similar in spirit to our GP Block. However, unlike RegionViT and GCViT, in a GP Block the grouped features are not necessarily constraint to a particular rectangular region, but can correspond to any shape or even entirely disconnected image parts, which could extract precise features of objects. There is recent work using transformer decoder layers with cross-attention between visual tokens and learnable tokens (Carion et al., 2020; Cheng et al., 2022; Jaegle et al., 2021; Hudson & Zitnick, 2021). However, there are three fundamental differences between these and ours: (i) Each of our GP blocks operates as an 'encoder-decoder' architecture with two rounds of cross-attention between visual tokens and group tokens: the first round groups the visual tokens for group propagation, and the second round ungroups the updated groups back into visual tokens; (ii) The underlying functionality is different: GP blocks facilitate more efficient global information propagation throughout the ViT, while previous work applies the decoder to obtain the final results for inference (e.g bounding boxes, or masks in (Carion et al., 2020; Cheng et al., 2022)); (iii) The GP block is a general module that can be insert into any layer of the ViT, while previous work utilizes the decoder only in the end of the network.

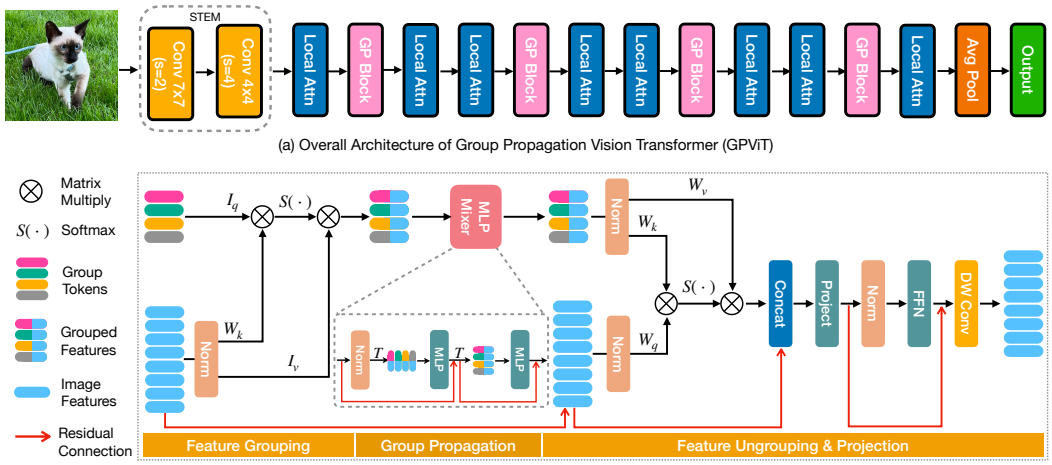
High-Resolution Visual Recognition. Previous work (Wang et al., 2020; Cheng et al., 2020) has shown that high-resolution images and features are beneficial to visual recognition tasks, especially to those requiring the perception of fine-grained image details, for example, semantic segmentation (Wang et al., 2020), pose-estimation (Sun et al., 2019), and small object detection (Yang et al., 2022). For example, HRNet (Wang et al., 2020) introduces a high-resolution ConvNet backbone. It maintains a high-resolution branch and exchanges information between different resolutions of features with interpolation and strided convolutions. Inspired by this work, HRFormer (Yuan et al., 2021) and HRViT (Gu et al., 2021) replace the convolutions in HRNet with self-attention blocks. GPViT is even simpler: it maintains single-scale and high-resolution feature maps without requiring any cross-resolution information to be maintained.

Object-Centric Representation. Our idea of performing information propagation among grouped regions is related to object-centric representation learning (Wang & Gupta, 2018; Kipf & Welling, 2016; Watters et al., 2017; Qi et al., 2020; Locatello et al., 2020; Kipf et al., 2021; Elsayed et al., 2022; Xu et al., 2022). For example, Locatello et al. (2020) proposes slot-attention, which allows automatic discovery of object segments via a self-supervised reconstruction objective. Instead of using reconstruction, Xu et al. (2022) utilizes language as an alternative signal for object segmentation discovery and shows it can be directly transferred to semantic segmentation in a zero-shot transfer manner. All these works extract object-centric features for downstream applications, while our work inserts this object-centric information propagation mechanism as a building block inside ViT to compute high-resolution representations more efficiently and improve high-resolution features. In this aspect, our work is more related to (Li & Gupta, 2018) where the graph convolution operations are inserted into ConvNets for better spatial reasoning.

3 METHOD

We present the overall architecture of our Group Propagation Vision Transformer (GPViT) in Figure 2 (a). GPViT is designed for general high-resolution visual recognition. For stable training, we first feed the input image into a down-sampling convolutional stem to generate image features (also known as image tokens), as in Dosovitskiy et al. (2021); Liu et al. (2021). In GPViT we downsample by a factor of 8 by default. The features are therefore higher resolution than in the original ViT where the factor is 16. Unlike most recently proposed methods (Liu et al., 2021; Li et al., 2022b) that adopt a pyramid structure to generate features in multiple resolutions, we keep the features at a high resolution without any down-sampling.

After combining the initial image features with positional embeddings (Vaswani et al., 2017), we feed them into the core GPViT architecture. We replace the original self-attention block in ViT with local attention e.g. window attention Liu et al. (2021) to avoid the quadratic complexity of self-attention. However, stacking local attention blocks alone does not allow for long-range information exchange between patches and therefore is harmful to performance. To counter this problem, we propose



(a) Overall Architecture of Group Propagation Vision Transformer (GPViT)

(b) Architecture of Group Propagation Block (GP Block)

Figure 2: (a). GPViT architecture: The input image is first fed to a convolutional stem that downsamples by a factor of 8. Each pixel of this downsampled image is treated as a high resolution image token or *feature*, and positional embeddings are added (not show in figure). These features are fed into our transformer, which consists of 12 encoder layers. 8 of these use local attention, and 4 use our proposed GP Block to propagate global information between image features. (b). GP Block: Image features are grouped using a fixed number of learnable group tokens. An MLPMixer module is then used to exchange global information and update the grouped features. Next, the grouped features are queried by, and concatenated with the image features to transfer global information to every image feature. Finally the updated image features are transformed by a feed-forward network to produce the output.

the Group Propagation Block (GP Block)—which we describe in full in Section 3.1—to efficiently propagate global information across the whole image. In our implementation, we evenly combine the GP Block with the local attention layers to form our GPViT and keep the overall depth unchanged. Lastly, we average the final features to get the model's output.

3.1 Group Propagation Block

Our key technical contribution is the GP block, which efficiently exchanges global information between each image patch with a linear complexity. We visualize the structure of the GP block in Figure 2 (b). It has a bottleneck structure and composes of three stages, namely, *Feature Grouping*, *Group Propagation*, and *Feature Ungrouping*. In the first stage the image features are grouped, then in the second stage global information is propagated between the grouped features, and in the last stage, this global information is transferred back to the image features.

Feature Grouping. The input to a GP Block is a matrix of image features $X \in \mathbb{R}^{N \times C}$ (The blue tokens in Figure 2 (b)) where N is the total number of image features (or image tokens) and C is the dimensionality of each feature vector. We use M learnable group tokens stored in a matrix $G \in \mathbb{R}^{M \times C}$ (the multi-colored tokens in Figure 2 (b)) where the group number M is a model hyper-parameter. Grouping is performed using a simplified multi-head attention operation (Vaswani et al., 2017), which gives us grouped features $Y \in \mathbb{R}^{M \times C}$ (the half-and-half tokens in Figure 2 (b)):

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

$$Y = \text{Concat}_{\{h\}}\big(\text{Attention}(W_h^Q G_h, W_h^K X_h, W_h^V X_h)\big), \tag{2}$$

where d is the channel number, h is the head index, and $W_h^{\{Q,K,V\}}$ are projection matrices for the query, key, and values, respectively in the attention operation. We remove the feature projection layers after the concatenation operation and set W_h^Q and W_h^V to be identity matrix. Therefore, the grouped features are simply the weighted sum of image features at each head where the weights are computed by the attention operation.

Group Propagation. After acquiring the grouped features, we can update and propagate global information between them. We use an MLPMixer (Tolstikhin et al., 2021) (Equation 3; the red box in Figure 2 (b)) to achieve this, as MLPMixer provides a good trade-off between model parameters, FLOPs, and model accuracy. MLPMixer requires a fixed-sized input, which is compatible with our

fixed number of groups. Specifically, our MLPMixer contains two consecutive MLPs. Recall that $Y \in \mathbb{R}^{M \times C}$ contains the grouped features from the first *Feature Grouping* stage. We can update these features to $\tilde{Y} \in \mathbb{R}^{M \times C}$ with the MLPMixer by computing:

$$Y' = Y + \text{MLP}_1(\text{LayerNorm}(Y)^T))^T, \quad \tilde{Y} = Y' + \text{MLP}_2(\text{LayerNorm}(Y'))), \tag{3}$$

where the first MLP is used for mixing information between each group, and the second is used to mix channel-wise information.

Feature Ungrouping. After updating the grouped features, we can return global information to the image features through a *Feature Ungrouping* process. Specifically, the features are ungrouped using a transformer decoder layer where grouped features are queried by the image features.

$$U = \text{Concat}_{\{h\}}\big(\text{Attention}(\tilde{W}_h^Q X_h, \tilde{W}_h^K \tilde{Y}_h, \tilde{W}_h^V \tilde{Y}_h)\big), \tag{4}$$

$$Z' = W_{proj} * \text{Concat}(U, X), \quad Z'' = Z' + \text{FFN}(Z'), \quad Z = \text{DWConv}(Z''), \tag{5}$$

where $\tilde{W}_h^{\{Q,K,V\}}$ are the projection matrices in the attention operation, W_{proj} is a linear matrix that projects concatenated features Z' to the same dimension as image features X, FFN is a feed-forward network, and DWConv is a depth-wise convolution layer. We modify the original transformer decoder layer by replacing the first residual connection with a concatenation operation (Equation 4; the blue box in Figure 2 (b)), and move the feature projection layer after this to transform the feature to the original dimension. We find this modification benefits the downstream tasks in different sizes of models. In addition, inspired by recent works (Ren et al., 2022b), we add a depth-wise convolution at the end of GP Block to improve locality property of the features (Equation 5; the yellow box in Figure 2 (b)). Finally, a GP Block outputs Z as its final output.

3.2 Computational Costs and Architecture Variants of GPViT

On comparing computational costs between hierarchical and non-hierarchical ViTs. We visualize both the non-hierarchical and hierarchical ViT in Figure 3 (a), where the non-hierarchical ViT simply stacks attention blocks and the hierarchical ViT divides the network into several stages and down-samples the feature map at each stage. Naturally, with the same resolution input, the non-hierarchical ViT will have a higher computation cost. The cost is divided by two parts as shown in Figure 3 (b), which include the self-attention module and the FFN module. Our GPViT largely reduces the computation of information propagation by using our GP Block instead of the self-attention. However, the cost on FFN is still inevitable for high resolution image features. Thus we will expect higher FLOPs from GPViT compared to hierarchical ViT given similar model parameters. However, we believe non-hierarchical ViT is still a direction worthy of exploration given its simplicity in extracting high-resolution features and in removing the requirement on studying how to design effective downstream models to utilize multi-scale features in a hierarchical ViT. This helps to maintain the independence of the model pre-training and fine-tuning designs as mentioned in Li et al. (2022a). In our experiments, we will show that our GPViT can achieve better accuracy in detection and segmentation compared to state-of-the-art hierarchical ViTs with similar FLOPs.

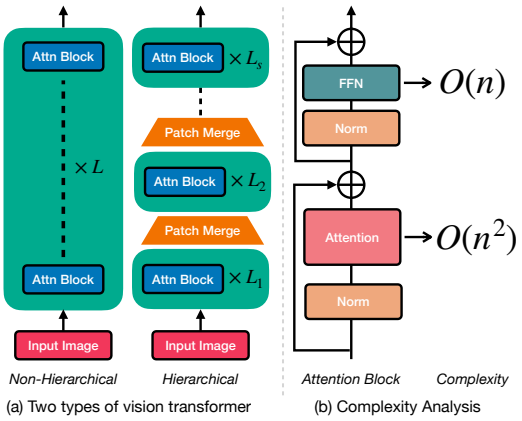


Figure 3: Structure comparison between non-hierarchical and hierarchical ViT. and the computation complexity of an attention block.

Variants. We study 3 variants of GPViT, as presented in Table 1. These variants are mainly differentiated by their feature channels. We use LePE attention Dong et al. (2022) as local attention by default. The FLOPs are counted using 224×224 inputs. Please refer to our Appendix for detailed architectures and training schedules for these variants.

Table 1: GPViT architecture variants.

Model	Channels	Param (M)	FLOPs (G)
GPViT-L1	216	9.3	5.8
GPViT-L2	348	23.6	15.0
GPViT-L3	432	36.2	22.9
GPViT-L4	624	75.4	48.2

4 EXPERIMENTS

4.1 IMAGENET-1K CLASSIFICATION

Setting: To ensure a fair comparison with previous works, we largely follow the training recipe of Swin Transformer (Liu et al., 2021). Specifically, we build models using the MMClassification (Contributors, 2020a) toolkit; the models are trained for 300 epochs with a batch size of 2048; AdamW optimizer was used with a weight decay of 0.05 and a peak learning rate of 0.002; the cosine learning rate schedule is used to gradually decrease the learning rate; and data augmentations are also inherited from (Liu et al., 2021) that include Mixup (Zhang et al., 2017), Cutmix (Yun et al., 2019), Random erasing (Zhong et al., 2020) and Rand augment (Cubuk et al., 2020).

Table 2: Comparison between GPViT and the recent proposed models on ImageNet-1K.

Model	Params (M)	FLOPs (G)	Top-1 Acc
Hierarchical			
PVT-T (Wang et al., 2021a)	13.2	1.9	75.1
PVT-S (Wang et al., 2021a)	24.5	3.8	79.8
CvT-13 (Wu et al., 2021a)	20.0	4.5	81.6
Focal-Tiny (Yang et al., 2021)	29.1	4.9	82.2
Focal-Base (Yang et al., 2021)	89.8	16.0	83.8
Swin-T (Liu et al., 2021)	29.0	4.5	81.3
Swin-B (Liu et al., 2021)	88	15.4	83.5
ShiftViT-T (Wang et al., 2022)	29	4.5	81.3
ShiftViT-B (Wang et al., 2022)	89	15.6	83.3
CSwin-T (Dong et al., 2022)	23	4.3	82.7
RegionViT-Ti (Chen et al., 2022a)	13.8	2.4	80.4
RegionViT-S (Chen et al., 2022a)	30.6	5.3	82.6
RegionViT-B (Chen et al., 2022a)	72.7	13.0	83.2
MViTv2-T (Li et al., 2022b)	24.0	4.7	82.3
GC ViT-XXT (Hatamizadeh et al., 2022)	12.0	2.1	79.8
GC ViT-XT (Hatamizadeh et al., 2022)	28.0	4.7	83.4
DaViT-Tiny (Ding et al., 2022)	28.3	4.5	82.8
MPViT-S (Lee et al., 2022)	22.8	4.7	83.0
Shunted-T(Ren et al., 2022b)	11.5	2.1	79.8
Shunted-S(Ren et al., 2022b)	22.4	4.9	82.9
DW-T (Ren et al., 2022a)	30.0	5.2	82.0
DW-B (Ren et al., 2022a)	91.0	17.0	83.8
DAT-T (Xia et al., 2022)	29.0	4.6	82.0
DAT-B (Xia et al., 2022)	88.0	15.8	84.0
Non-hierarchical			
DeiT-S (Touvron et al., 2020)	22.1	4.6	79.9
CaiT-XS24 (Touvron et al., 2021b)	26.6	5.4	81.8
CaiT-S48 (Touvron et al., 2021b)	89.5	18.6	83.5
LocalViT-S (Li et al., 2021b)	22.4	4.6	80.8
Visformer-S (Chen et al., 2021)	40.2	4.9	82.3
ConViT-S (d'Ascoli et al., 2021)	27.0	5.4	81.3
ConViT-B (d'Ascoli et al., 2021)	86.0	17.0	82.4
GPViT L1	9.3	5.8	80.5
GPViT L2	23.8	15.0	83.4
GPViT L3	36.2	22.9	84.1
GPViT L4	75.4	48.2	84.3

Results: We compare GPViT with the latest state-of-the-art visual recognition models on the ImageNet-1K dataset and report the results in Table 2. As shown in the table, because of its high-resolution design and effective global information propagation via the grouping mechanism, our method outperforms a number of recently proposed advanced vision transformer architectures. For example, GPViT-L2 achieves 83.4 top-1 accuracy, surpassing the recently published DAT (Xia et al., 2022), DWViT (Ren et al., 2022a) and MPViT (Lee et al., 2022) under a similar number of parameters. Notably, our GPViT follows the simple non-pyramid architecture of ViT (Li et al., 2022a) instead of the recent mainstream hierarchical design. The strong performance of our method suggests that the hierarchical design inherited from the CNN era is not necessary for the high-performing vision transformers. This also corroborates with some very recent studies (Li et al., 2022a). On the other hand, because of its high-resolution design, we observe that the FLOPs of our method are higher than most methods with a similar number of parameters as explained in Section 3.2. However, when we align the models in terms of a similar number of FLOPs, we observe that GPViT can achieve a comparable top-1 accuracy, but with much fewer parameters. For example, GPViT-L2 (15.0 G) has similar FLOPs to the Swin Transformer-B (15.4 G) and ShiftViT-B (15.6 G), but it achieves a similar accuracy with much fewer parameters (23.8 M v.s. 88 M and 89 M).

4.2 COCO OBJECT DETECTION AND INSTANCE SEGMENTATION

Setting: We follow Chen et al. (2022b) to use Mask R-CNN and RetinaNet models for the COCO object detection and instance segmentation tasks. We use ViTAdapter (Chen et al., 2022b) to generate multi-scale features as FPN inputs and evaluate the model for both 1× and 3× training schedules.

Table 3: RetinaNet object detection performance on MS COCO val2017 1× and 3× +MS schedule.

Backbone	Params (M)	FLOPs (G)	1× AP^{bb}	3× AP^{bb}
Hierarchical				
ResNet101 (He et al., 2016)	56	315	38.5	40.9
CMT-S(Guo et al., 2022)	44	231	44.3	-
DAT-S(Xia et al., 2022)	60	359	45.7	47.9
Shunted-S(Ren et al., 2022b)	42	-	45.4	46.4
PVT-L (Wang et al., 2021a)	71	345	42.6	-
PVTv2-B5 (Wang et al., 2021b)	91	335	46.1	-
Swin-B (Liu et al., 2021)	98	477	44.7	-
RegionViT-B (Chen et al., 2022a)	83	308	43.3	46.1
MPViT-B (Lee et al., 2022)	95	503	-	48.3
DaViT-Base (Ding et al., 2022)	103	471	46.7	48.7
Non-hierarchical				
GPViT-L1	21	317	45.8	48.1
GPViT-L2	37	542	48.0	49.0
GPViT-L3	52	731	48.3	49.4
GPViT-L4	96.1	1319	48.7	49.8

Results: Like in the above section, we compare GPViT's performance with the latest state-of-the-art backbones with all trained on ImageNet-1K. We report the results in Table 4 and Table 3. For both detection methods our GPViT is able to surpass the other backbones by a large margin when they are in similar parameter amount ranges. With Mask R-CNN (Table 4), GPViT-L2 based model surpasses the Swin Transformer-S (Liu et al., 2021) based counterpart by 5.1 AP^{bb} and 3.0 AP^{mk} for the 1×

Table 4: Object detection performance on MS COCO val2017 with 1× and 3× schedule. The bold number indicates the best number within the section, and for MaskRCNN, both AP^b and AP^m are annotated.

Backbone	Params (M)	FLOPs (G)	1× AP^{bb}	1× AP^{mk}	3× AP^{bb}	3× AP^{mk}
Hierarchical						
ResNet101 (He et al., 2016)	63	336	40.4	36.4	42.8	38.5
ResNeXt101-32x4d (Xie et al., 2017)	62	340	41.9	37.5	44.0	39.2
RegionViT-B (Chen et al., 2022a)	92	287	44.2	40.8	47.6	43.4
PVT-B (Wang et al., 2021a)	81	364	42.9	39.5	44.5	40.7
Swin-S (Liu et al., 2021)	69	354	44.8	40.9	47.6	42.8
Swin-B (Liu et al., 2021)	107	496	45.5	41.3	-	-
MViTv2-S (Li et al., 2022b)	54	326	-	-	49.9	45.1
DaViT-Small (Ding et al., 2022)	69	351	47.7	42.9	49.5	44.3
DaViT-Base (Ding et al., 2022)	107	491	48.2	43.3	49.9	44.6
DAT-S (Xia et al., 2022)	69	387	47.1	42.5	49.0	44.0
DW-B (Ren et al., 2022a)	111	505	-	-	49.2	44.0
CMT-S (Guo et al., 2022)	44	231	44.6	40.7	-	-
CSwin-S (Dong et al., 2022)	54	342	47.9	43.2	50.0	44.5
CSwin-B (Dong et al., 2022)	97	526	48.7	43.9	50.8	44.9
MPViT-B (Lee et al., 2022)	95	503	-	-	49.5	44.5
Shunted-S(Ren et al., 2022b)	42	-	47.1	52.1	49.1	43.9
Shunted-B(Ren et al., 2022b)	59	-	48.0	43.2	50.1	45.2
GC ViT-T (Hatamizadeh et al., 2022)	47	263	-	-	46.5	41.8
Non-hierarchical						
ViT-Adapter-T (Chen et al., 2022b)	28	-	41.1	37.5	46.0	41.0
ViT-Adapter-S (Chen et al., 2022b)	47	-	44.7	39.9	48.2	42.8
ViT-Adapter-B (Chen et al., 2022b)	102	-	47.0	41.8	49.6	43.6
GPViT-L1	33	457	48.1	42.7	50.2	44.3
GPViT-L2	50	690	49.9	43.9	51.4	45.1
GPViT-L3	64	884	50.4	44.4	51.6	45.2
GPViT-L4	109	1489	51.0	45.0	52.1	45.7

training schedule while they shared a similar parameter amount. When we align the models by their FLOPs, our model also achieves better detection accuracy but with fewer parameters. For example, GPViT-L1 based Mask R-CNN models have similar FLOPs to DW-B (Ren et al., 2022a), but ours' achieve a 0.9 AP^{bb} and 0.3 AP^{mk} advantage and have much fewer parameters. On comparing with models that are also equipped with ViTAdapter (Chen et al., 2022b), GPViT achieves an better AP with fewer parameters, e.g., our smallest GPViT-L1 outperforms ViT-Adapter-B in both training schedules. These results suggest GPViT's effectiveness in extracting good regional features that are crucial for object detection and instance segmentation. A similar conclusion can be drawn from the single-stage RetinaNet detector. With RetinaNet (Table 3), GPViT-L1 has FLOPs similar to the recently proposed RegionViT-B (Chen et al., 2022a) method, but it outperforms RegionViT-B by 2.5 and 2.0 AP^{bb} in both 1× and 3× schedules with only 25% parameters.

Table 5: Comparison to ViTDet on COCO using Mask R-CNN. Our detector is trained with 36 epochs while ViTDet is trained for 100 epochs.

Model	Pre-train	Param (M)	FLOPs(T)	AP^{bb}	AP^{mk}
ViTDet-ViT-B	MAE	111	0.8	51.6	45.9
ViTDet-ViT-B	Sup	111	0.8	47.6	42.4
GPViT-L3	Sup	50	0.8	51.6	45.2

In Table 5, we compare our Mask R-CNN with the recently proposed ViTDet (Li et al., 2021a) that also uses a non-hierarchical ViT as the backbone network. Here we continue to use the commonly used 3× (36 epochs) training recipe for GPViT. The results show that under similar FLOPs, even if ViTDet is equipped with more parameters (111M), advanced masked-auto-encoder (MAE) pre-training (He et al., 2022), a longer training schedule (100 epochs), and heavy regularizations like large-scale jittering (Ghiasi et al., 2021), our model can still achieve a comparable performance, which further validates the effectiveness of GPViT.

4.3 ADE20K Semantic Segmentation

Setting: We follow previous works (Liu et al., 2021) and use UperNet (Xiao et al., 2018) as the segmentation network. We also report performance when using the recently proposed SegFormer (Xie et al., 2021) model. For both models, we train the model for 160k iterations.

Results: We summarise the segmentation performance of GPViT and other state-of-the-art backbone networks in Table 6. For UperNet, we report results with the largest available model size for the competing methods to show how far we can go in the segmentation task. Thanks to its high-resolution design, GPViT outperforms all competing methods in mIoU with less FLOPs and much fewer parameters. For example, GPViT-L1 only have 37M parameters but it can achieve comparable mIoU to methods with only half the number of FLOPs. This result tells us that for tasks requiring the

perception of fine-grained details, scaling-up feature resolution is a better strategy compared to scaling up model size. On the other hand, GPViT is also good for the SegFormer design, which has a simpler structure than UperNet, hence the backbone ability is more highlighted. Specifically, GPViT achieves better mIoU than the recently proposed vision transformers with a similar parameter counts, including HRViT (Gu et al., 2021) that was particularly designed for the semantic segmentation task and also uses high-resolution features. We attribute the promising results to GPViT's high-resolution design and its effective encapsulation of global information.

4.4 Ablation Studies and Visualization

Setting: We conduct ablation studies using two types of local attention: the simple window attention Liu et al. (2021) and the more advanced LePE attention Dong et al. (2022), which we used in previous experiments. We use the L1 level models (Param $<$ 10 M) for all experiments. All models are pre-trained on ImageNet classification task for 300 epochs using the same setting as in Section 4.1. We report both ImageNet Top-1 accuracy and ADE20K SegFormer semantic segmentation performance measurement. Please refer to our appendix for more ablation experiments.

Building GPViT step by step. Here we show how we build GPViT step by step and present the results in Table 7 (Left). We start building our GPViT from a low-resolution vanilla DeiT with patch sizes of 16 and embedding channels of 216 (same as GPViT-Tiny). It achieves 77.4 top-1 accuracy on ImageNet and 42.2 mIoU on ADE20K. Then we make it to high-resolution by shrinking the patch size to 8. The FLOPs on ImageNet and ADE20K are raised by 4.4-fold and 7.0-fold respectively. Although the ImageNet accuracy is improved to 79.2, we failed to train this model on the segmentation task because of training instability. The result suggests that simply enlarging the feature resolution using global self-attention can lead to FLOPs explosion and convergence difficulty. Then we replace self-attention with window attention (Liu et al., 2021) and the more advanced LePE attention (Dong et al., 2022). For both local attention, the FLOPs of ImageNet and ADE20K drop to 5.8 G and 34G respectively. Then we add GP Block two both of them, we can see that the accuracy and mIoU are all improved for both local attention while the FLOPs are unchanged. These results suggest the effectiveness of using high-resolution features to boost visual recognition performance and the necessity of combining local attention blocks and our GP Block to maintain a reasonable computation cost.

Table 6: ADE20K semantic segmentation. The FLOPs are computed using size 512×2048 for UperNet and 512×512 for SegFormer.

UperNet			
Backbone	Params (M)	FLOPs (G)	mIoU
Hierarchical			
Focal-B (Yang et al., 2021)	126	1354	49.0
Swin-B (Liu et al., 2021)	121	1188	49.7
Twins-SVT-L (Chu et al., 2021)	133	-	48.8
DAT-T (Xia et al., 2022)	121	1212	50.5
DaViT-Base (Ding et al., 2022)	121	1175	49.4
MPViT-B (Lee et al., 2022)	105	1186	50.3
RegionViT-B (Chen et al., 2022a)	78	-	47.5
GC ViT-B (Hatamizadeh et al., 2022)	125	1348	49.0
DW-B (Ren et al., 2022a)	125	1200	48.7
XCiT-M24/16 (Ali et al., 2021)	109	-	47.6
PatchConv (Touvron et al., 2021a)	229	1550	49.4
ShiftViT-B (Wang et al., 2022)	121	1174	49.2
Non-hierarchical			
Deit-B (Touvron et al., 2020)	120	786	45.3
ViT-Adapter-B (Chen et al., 2022b)	134	-	48.1
GPViT-L1	37	568	49.1
GPViT-L2	53	775	50.2
GPViT-L3	66	944	51.7
GPViT-L4	107	1469	52.5

SegFormer			
Backbone	Params (M)	FLOPs (G)	mIoU
Hierarchical			
MiT-B2 (Xie et al., 2021)	24	64	46.5
MiT-B4 (Xie et al., 2021)	61	96	50.4
CSwin-T(Dong et al., 2022)	22	28	47.8
CSwin-S(Dong et al., 2022)	37	78	49.9
HRViT-b3(Gu et al., 2021)	29	68	50.2
HILA+MiT-B1 (Leung et al., 2022)	22	31	45
HILA+MiT-B2 (Leung et al., 2022)	31	76	46
Non-hierarchical			
GPViT-L1	9	34	46.9
GPViT-L2	24	83	49.2
GPViT-L3	36	123	50.8
GPViT-L4	76	249	51.3

Global Information Exchange. In this section, we compare our GP Block with other blocks that can exchange global information between image features. The competing methods include global attention block, convolution propagation block (Li et al., 2022a), and shifting window-attention block (Liu et al., 2021) designed for window attention. We follow ViTDet (Li et al., 2021a) to build the convolution propagation block that stakes two 3×3 convolution layers with a residual connection. We use the original version of the shifting window-attention block as in Liu et al. (2021). The resulting models are acquired by putting those blocks in the same place as we put GP Block. We report the results in Table 7 (Right). We observe that simply replacing the local attention layers with convolution layers causes severe performance drops in both local attentions. Also, while replacing local attention with global attention can improve accuracies and mIoU, it will also cause a rapid increase in FLOPs. For

Table 7: Left: Ablation studies on feature resolution and GP Block. Right: Ablation study on different global information propagation methods.

Setting	FLOPs (G)	Top-1 Acc	FLOPs (G)	mIoU
ViT-D216-P16	1.8	77.4	16	42.2
ViT-D216-P8	8.8	79.2	113	-
+ Win attn	5.8	78.1	34	41.7
+ GP Block	5.8	79.8	34	45.5
+ LePE attn	5.8	79.5	34	46.2
+ GP Block	5.8	80.5	34	46.9

Attention	Global Info	FLOPs (G)	Top-1 Acc	FLOPs (G)	mIoU
Window	None	5.8	78.2	34	41.7
	Conv	6.6	75.8	38	39.5
	Global attn	6.8	78.8	62	44.0
	Win shift	5.8	78.1	34	40.7
	GP Block	5.8	79.8	34	45.5
LePE	None	5.8	79.5	34	46.2
	Conv	6.6	77.7	38	45.5
	Global attn	6.8	80.4	62	46.7
	GP Block	5.8	80.5	34	46.9

window attention, we found that using the shifting window strategy slightly harms the performance. We guess this is caused by the insufficient shifting window layer: half of Swin Transformer layers are shifting window layers, but we only place four here. On the other hand, for both local attentions, GP Block achieves the best performance on ImageNet and ADE20K while keeping FLOPs nearly unchanged compared with models without a global information propagation layer. These results show GP Block's effectiveness in propagating global information.

Number of group tokens In Table 8 we compare different combinations of the number of groups tokens in GP Block. We find using a large number of group tokens across the whole network can give us higher accuracy on ImageNet but at additional computational cost. However, using too few group e.g. 16 tokens will harm performance. Therefore, in GPViT we choose to progressively decrease the number of group tokens from 64 to 16 gradually. This strategy gives us a good trade-off between accuracy and computational cost.

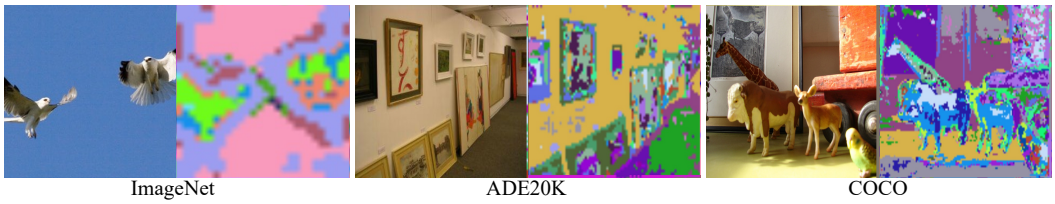
Table 8: Ablation study on the group tokens number combinations.

Combination	FLOPs (G)	Top-1 Acc
{16, 16, 16, 16}	5.7	79.9
{32, 32, 32, 32}	5.8	80.3
{64, 64, 64, 64}	6.0	80.7
{16, 32, 32, 64}	5.8	80.0
{64, 32, 32, 16}	5.8	80.5

Grouped Features Propagation In Table 9 we compare different ways of global information propagation. We note that even when we add nothing to explicitly propagate global information, the image features are grouped and ungrouped so the global information can still be exchanged in these two operations, which achieves 79.8% accuracy on ImageNet. We also find that self-attention can achieve a similar accuracy with MLPMixer (80.4 v.s. 80.5), but it incurs more computation to the model. Therefore, in GPViT we use MLPMixer for propagating global information.

Table 9: Ablation study on the propagation approach of grouped features.

Method	FLOPs (G)	Top-1 Acc
None	5.7	79.8
SelfAttn	6.2	80.7
MLPMixer	5.8	80.5



ImageNet ADE20K COCO

Figure 4: Feature grouping visualisation using model trained on ImageNet-1k, ADE20K and COCO.

Visualising groups. In Figure 4, we visualise the feature grouping results using models trained on ImageNet, ADE20K and COCO. We observe that the feature grouping can separate a image's foreground and background in all three datasets. When the model receives fine-grained supervision on ADE20K and COCO, the feature grouping can correspond to more details in the image.

5 Conclusion

In this paper, we have presented the Group Propagation Vision Transformer (GPViT): a non-hierarchical vision transformer designed for high-resolution visual recognition. The core of GPViT is the GP Block, which was proposed to efficiently exchange global information among high-resolution features. The GP Block first forms grouped features and then updates them through *Group Propagation*. Finally, these updated group features are queried back to the image features. We have shown that GPViT can achieve better performance than previous works on ImageNet classification, COCO object detection and instance segmentation, and ADE20K semantic segmentation.

6 REPRODUCIBILITY STATEMENT

All experiments reported in this paper are reproducible and we are committed to releasing the code and pre-trained model weights once accepted.

REFERENCES

Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021. 8

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. Computer Vision – ECCV 2020, pp. 213 – 229, 2020. ISBN 978-3-030-58452-8. 3

Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *International Conference on Learning Representations*, 2022a. 2, 3, 6, 7, 8

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 16

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022b. 6, 7, 8, 16

Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The Vision-friendly Transformer. *arxiv.org*, April 2021. 6

Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 3

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting Spatial Attention Design in Vision Transformers. *arXiv.org*, April 2021. 8

MMClassification Contributors. Openmmlab's image classification toolbox and benchmark. https://github.com/open-mmlab/mmclassification, 2020a. 6, 16

MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020b. 16

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020. 6, 16

Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases. *arXiv.org*, cs.CV, 03 2021. URL arXiv.org. 2, 6

Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *ECCV*, 2022. 6, 7, 8

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12124–12134, June 2022. 2, 3, 5, 6, 7, 8

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy. 2, 3

Gamaleldin F Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *arXiv preprint arXiv:2206.07764*, 2022. 3

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6824–6835, 2021. 2

Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, 2021. 7

Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference. *arXiv.org*, April 2021. 2

Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, Yu-Hsin Chen, Liangzhen Lai, Vikas Chandra, and David Z Pan. Hrvit: Multi-scale high-resolution vision transformer. *arXiv preprint arXiv:2111.01236*, 2021. 3, 8

Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12175–12185, June 2022. 2, 6, 7

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021. 3

Ali Hatamizadeh, Hongxu Yin, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. *arXiv preprint arXiv:2206.09959*, 2022. 2, 3, 6, 7, 8

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 6, 7

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022. 7

Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 603–612, 2019. 3

Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International conference on machine learning*, pp. 4487–4499. PMLR, 2021. 3

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 3

Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021. 3

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 3

Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7287–7296, June 2022. 2, 6, 7, 8

Gary Leung, Jun Gao, Xiaohui Zeng, and Sanja Fidler. Hila: Improving semantic segmentation in transformers using hierarchical inter-level attention. *arXiv:2207.02126*, 2022. 8

Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021a. 7, 8

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022a. 1, 2, 5, 6, 8

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4804–4814, June 2022b. 2, 3, 6, 7

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. LocalViT: Bringing Locality to Vision Transformers. *arXiv.org*, April 2021b. 6

Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. *Advances in Neural Information Processing Systems*, 31, 2018. 3

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv.org*, March 2021. 1, 2, 3, 6, 7, 8, 16

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 3

Haozhi Qi, Xiaolong Wang, Deepak Pathak, Yi Ma, and Jitendra Malik. Learning long-term visual dynamics with region proposal interaction networks. *arXiv preprint arXiv:2008.02265*, 2020. 3

Pengzhen Ren, Changlin Li, Guangrun Wang, Yun Xiao, Qing Du, Xiaodan Liang, and Xiaojun Chang. Beyond fixation: Dynamic window visual transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11987–11997, June 2022a. 6, 7, 8

Sucheng Ren, Daquan Zhou, Shengfeng He, Jiashi Feng, and Xinchao Wang. Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10853–10862, June 2022b. 2, 5, 6, 7, 16

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5693–5703, 2019. 3

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34: 24261–24272, 2021. 2, 4

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 2, 6, 8

Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Piotr Bojanowski, Armand Joulin, Gabriel Synnaeve, and Hervé Jégou. Augmenting convolutional networks with attention-based aggregation. *arXiv preprint arXiv:2112.13692*, 2021a. 8

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with Image Transformers. *arXiv.org*, cs.CV, 03 2021b. URL arXiv.org. 6

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2, 3, 4

Guangting Wang, Yucheng Zhao, Chuanxin Tang, Chong Luo, and Wenjun Zeng. When shift operation meets vision transformer: An extremely simple alternative to attention mechanism. In *AAAI*, 2022. 6, 8

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 3

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, 2021a. 1, 2, 6, 7

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021b. 6

Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 399–417, 2018. 3

Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. *Advances in neural information processing systems*, 30, 2017. 3

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvT: Introducing Convolutions to Vision Transformers. *arXiv.org*, March 2021a. 1, 6

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, 2021b. 2

Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4794–4803, June 2022. 6, 7, 8

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018. 2, 7

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 7, 8

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 7

Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18134–18144, 2022. 3

Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9981–9990, 2021a. 2, 3

Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34: 28522–28535, 2021b. 2

Chenhongyi Yang, Zehao Huang, and Naiyan Wang. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13668–13677, June 2022. 3

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. In *NeurIPS*, 2021. 6, 8

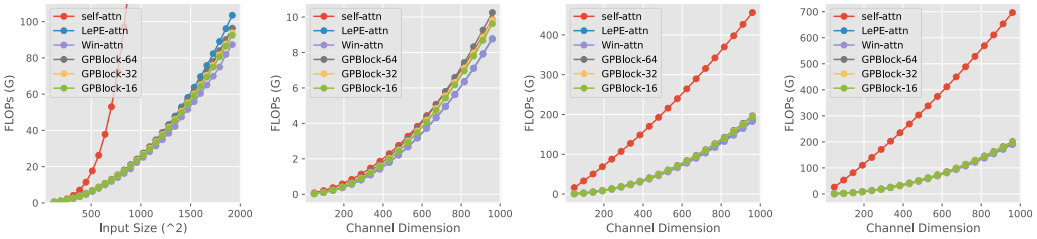
Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *Advances in Neural Information Processing Systems*, 34:7281–7293, 2021. 3

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019. 6, 16

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 6, 16

Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Ö Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3417–3425, 2022. 3

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, 2020. 6, 16



(a) Channel Dimension = 348 (b) Input size = (224, 224) (c) Input size = (800, 1280) (d) Input Size = (512, 2048)

Figure 5: Layer-wise FLOPs comparison between different attention blocks on high-resolution features. "GPBlock-N" denotes GP Block with N group tokens. (a) FLOPs v.s. Input size when using feature channel dimension of 348 (same as GPViT-s1); (b-d): FLOPs v.s. feature channel dimension on three typical input sizes using by ImageNet classification, COCO object detection and instance segmentation, and ADE20K semantic segmentation. Our GP Block can effectively gather and propagate global information like a self-attention block while having similar or even fewer FLOPs with local attention blocks when the model or input scale-up.

Table 10: Inference speed comparison on three typical input sizes: 224×224 for ImageNet-1K classification, 800×1280 for COCO object detection and instance segmentation, and 512 × 2048 for ADE20K semantic segmentation. The results are evaluated on NVIDIA 2080Ti GPUs. ('OOM' denotes out of GPU memory.)

Model	Param (M)	Inference Time (ms) 224×224	800×1280	512×2048
Low-resolution Baseline				
ViT-D216-P16	7.4	0.9	155	150
ViT-D348-P16	18.5	1.4	167	162
ViT-D432-P16	28.5	1.8	177	173
ViT-D624-P16	57.9	2.9	206	203
High-resolution Baseline				
ViT-D216-P8	7.4	7.5	OOM	OOM
ViT-D348-P8	18.5	9.5	OOM	OOM
ViT-D432-P8	28.5	11.3	OOM	OOM
ViT-D624-P8	57.9	15.9	OOM	OOM
GPViT-L1	9.3	2.7	83	87
GPViT-L2	23.8	5.1	132	137
GPViT-L3	36.2	7.0	174	182
GPViT-L4	75.4	11.1	281	290

A Further Ablation Studies

A.1 Study on Running Efficiency

In Table 10, we compare the inference speed of GPViT with ViT baselines. Specifically, for each variant of our GPViT, we compare it to ViT models with patch size 16 (low-resolution) and patch size 8 (high-resolution) while keeping the channel dimensions the same. We report inference time using three different input sizes, which correspond to the three typical input sizes used by ImageNet-1k classification, COCO object detection and instance segmentation, and ADE20K semantic segmentation. We draw three observations from the results:

- (i) When using small-sized inputs, GPViT runs slower than the low-resolution ViT. Despite the efficient design of our GP Block and the use of local attention, the high-resolution design still incurs a significant cost for forward passes, slowing down inference speed.
- (ii) When the models are applied to downstream tasks where they take larger-sized inputs, all but the largest GPViT models are faster than their low-resolution ViT counterparts. For example, when the model channel number is 216, GPViT takes 83 ms to process an 800×1280 sized image, while ViT-D216-P16 takes 155 ms. In this case, the self-attention operations with quadratic complexity severely slow down the speed of the ViT even with low resolution features. On the other hand, the computations in GP Block and local attentions grow much less than self-attention when the input scales up.

- (iii) GPViT is faster than the high-resolution ViT baselines when using small inputs. In addition, high-resolution ViTs are not even able to process large-sized inputs: we got *Out of Memory* errors when using a NVIDIA 2080Ti GPU with 11 GB memory. This highlights our technical contribution of efficiently processing high-resolution features with GPViT.

We further study how the computation cost for high-resolution features changes when the model size and input scale up by examining FLOP counts. The results are shown in Figure 5 where we compare GP Block with different group numbers to self-attention and local-attention operations: Self-attention and GP Block can both exchange global information between image features, but the computational cost of GP Block grows much slower than self-attention. Local attention operations have a similar level of efficiency to GP Block, but are unable to exchange global information because of their limited receptive field.

B Implementation Details

B.1 Model Details of GPViT

The model details of different GPViT variants are presented in Table 11. Different GPViT variants are main difference by their model width (channels) and share similar hyper-parameters in other architecture designs.

Table 11: GPViT model details for different variants.

Variants	GPViT-L1	GPViT-L2	GPViT-L3	GPViT-L4
Patch Size	8	8	8	8
Channel Dimension	216	348	432	624
Number of Transformer Layers	12	12	12	12
LePE Strip Size	2	2	2	2
Attention Heads	12	12	12	12
FFN Expansion	4	4	4	4
GP Block Positions	{1, 4, 7, 10}	{1, 4, 7, 10}	{1, 4, 7, 10}	{1, 4, 7, 10}
GP Block Group Numbers	{64, 32, 32, 16}	{64, 32, 32, 16}	{64, 32, 32, 16}	{64, 32, 32, 16}
Feature Grouping Attention Heads	6	6	6	6
Feature Ungrouping Attention Heads	6	6	6	6
MLPMixer Patch Expansion	0.5	0.5	0.5	0.5
MLPMixer Channel Expansion	4	4	4	4
ImageNet Drop Path Rate	0.2	0.2	0.3	0.3
Parameters (M)	9.3	23.6	36.2	75.4

B.2 Training Recipe for ImageNet

The ImageNet experiments are based on the MMClassification toolkit (Contributors, 2020a). The models are trained for 300 epochs with a batch size of 2048; AdamW optimizer was used with a weight decay of 0.05 and a peak learning rate of 0.002. The cosine learning rate schedule is adopted. The gradient clip is set to 5.0 (we also tested 1.0 and found it worked well too); data augmentation strategies are from (Liu et al., 2021) that include Mixup (Zhang et al., 2017), Cutmix (Yun et al., 2019), Random erasing (Zhong et al., 2020) and Rand augment (Cubuk et al., 2020).

B.3 Training Recipe for COCO

The COCO experiments are based on the MMDetection toolkit (Chen et al., 2019). Following commonly used training settings, both Mask R-CNN and RetinaNet models are trained for 12 epochs (1$\times$) and 36 epochs (3$\times$). For the 3$\times$ schedule, we follow previous works (Liu et al., 2021; Ren et al., 2022b) to use multi-scale input during training. AdamW optimizer was used with an initial learning rate of 0.0002 and weight decay of 0.05. We used ViTAdapter (Chen et al., 2022b) to generate multi-scale features and followed the default hyper-parameter settings in Chen et al. (2022b).

B.4 Training Recipe for ADE20K

The ADE20K experiments are based on the MMSegmentation toolkit (Contributors, 2020b). Following commonly used training settings, both UperNet and SegFormer models are trained for 160000 iterations. The input images are cropped to 512$\times$512 during training. AdamW optimizer was used with an initial learning rate of 0.00006 and weight decay of 0.01. We did not use ViTAdapter (Chen et al., 2022b) for segmentation experiments.