# BiFormer: Vision Transformer with Bi-Level Routing Attention

# BiFormer：具有双级路由注意力的视觉变换器

Lei Zhu Xinjiang Wang Zhanghan Ke Wayne Zhang Rynson Lau

朱磊 王新疆 柯章汉 张维恩 刘仁松

City University of Hong Kong SenseTime Research

香港城市大学 商汤科技研究院

1zhu68-c, zhanghake2-c }@my.cityu.edu.hk, wangxinjiang, wayne.zhang }@sensetime.com Rynson.Lau@cityu.edu.hk

1zhu68-c, zhanghake2-c }@my.cityu.edu.hk, wangxinjiang, wayne.zhang }@sensetime.com Rynson.Lau@cityu.edu.hk

Abstract

摘要

As the core building block of vision transformers, attention is a powerful tool to capture long-range dependency. However, such power comes at a cost: it incurs a huge computation burden and heavy memory footprint as pairwise token interaction across all spatial locations is computed. A series of works attempt to alleviate this problem by introducing handcrafted and content-agnostic sparsity into attention, such as restricting the attention operation to be inside local windows, axial stripes, or dilated windows. In contrast to these approaches, we propose a novel dynamic sparse attention via bi-level routing to enable a more flexible allocation of computations with content awareness. Specifically, for a query, irrelevant key-value pairs are first filtered out at a coarse region level, and then fine-grained token-to-token attention is applied in the union of remaining candidate regions (i.e., routed regions). We provide a simple yet effective implementation of the proposed bi-level routing attention, which utilizes the sparsity to save both computation and memory while involving only GPU-friendly dense matrix multiplications. Built with the proposed bi-level routing attention, a new general vision transformer, named BiFormer, is then presented. As BiFormer attends to a small subset of relevant tokens in a query adaptive manner without distraction from other irrelevant ones, it enjoys both good performance and high computational efficiency, especially in dense prediction tasks. Empirical results across several computer vision tasks such as image classification, object detection, and semantic segmentation verify the effectiveness of our design. Code is available at https://github.com/rayleizhu/BiFormer.

作为视觉变换器核心构建块的注意力，是一种强大的捕获长距离依赖的工具。然而，这种力量是有代价的：它带来了巨大的计算负担和沉重的内存占用，因为需要在所有空间位置进行成对的标记交互计算。一系列的工作尝试通过在注意力中引入手工制作的内容不可知稀疏性来缓解这个问题，例如将注意力操作限制在局部窗口内、轴向条纹或扩张窗口中。与这些方法不同，我们提出了一种通过双层路由实现的新型动态稀疏注意力，以实现具有内容意识的计算分配更加灵活。具体来说，对于每个查询，首先在粗略的区域级别过滤掉不相关的键-值对，然后在剩余候选区域（即路由区域）的并集上应用细粒度的标记到标记的注意力。我们提供了所提出双层路由注意力的一个简单而有效的实现，它利用稀疏性来节省计算和内存，同时只涉及GPU友好的密集矩阵乘法。使用所提出的双层路由注意力构建的新的通用视觉变换器，命名为BiFormer。由于BiFormer以自适应的方式关注查询中的一小部分相关标记，而不受其他不相关标记的干扰，因此它在性能和计算效率上都表现出色，特别是在密集预测任务中。在图像分类、目标检测和语义分割等多个计算机视觉任务上的实证结果验证了我们的设计的有效性。代码可在 https://github.com/rayleizhu/BiFormer 找到。

# 1. Introduction

# 1. 引言

Transformer has many properties that are suitable for building powerful data-driven models. First, it is able to capture long-range dependency in the data . Second, it is almost inductive-bias-free and thus makes the model more flexible to fit tons of data [15]. Last but not least, it enjoys high parallelism, which benefits training and inference of large models . Hence, transformer has not only revolutionized natural language processing but also shown very promising progress in computer vision.

Transformer具有许多适合构建强大数据驱动模型的属性。首先，它能够捕捉数据中的长距离依赖性 。其次，它几乎不包含归纳偏置，从而使模型更加灵活，能够适应大量数据 [15]。最后但同样重要的是，它具有高度的并行性，这有利于大型模型的训练和推理 。因此，transformer不仅革新了自然语言处理，而且在计算机视觉领域也显示出非常令人鼓舞的进展。

The computer vision community has witnessed an explosion of vision transformers in the past two years , . Among these works, a popular topic is to improve the core building block, i.e., attention. In contrast to convolution, which is intrinsically a local operator, a crucial property of attention is the global receptive field, which empowers vision transformers to capture long-range dependency [42]. However, such a property comes at a cost: as attention computes pairwise token affinity across all spatial locations, it has a high computational complexity and incurs heavy memory footprints.

计算机视觉领域在过去两年见证了视觉transformer的爆发式增长 ， 。在这些工作中，一个热门话题是改进核心构建块，即注意力机制。与本质上为局部操作符的卷积不同，注意力的一个关键属性是全球感受野，这使得视觉transformer能够捕捉长距离依赖性 [42]。然而，这样的属性是有代价的：由于注意力计算所有空间位置之间的成对token亲和力，它具有高计算复杂度，并产生大量的内存占用。

To alleviate the problem, a promising direction is to introduce sparse attention [6] to vision transformers, so that each query attends to a small portion of key-value pairs instead of all. In this fashion, several handcrafted sparse patterns have been explored, such as restricting attention in local windows [29], dilated windows [41, 46], or axial stripes [46]. On the other hand, there are also works trying to make the sparsity adaptive to data . However, while they use different strategies to merge or select key/- value tokens, these tokens are query-agnostic, i.e., they are shared by all queries. Nonetheless, according to the visualization of pretrained ViT [15] and DETR [1], queries in different semantic regions actually attend to quite different key-value pairs. Hence, forcing all queries to attend to the same set of tokens may be suboptimal.

为了缓解这个问题，一个有前景的方向是将稀疏注意力 [6] 引入视觉变换器中，使得每个查询只关注一小部分键-值对，而不是全部。按照这种方式，已经探索了几种手工设计的稀疏模式，例如限制在局部窗口 [29] 内的注意力，扩张窗口 [41, 46]，或者轴向条纹 [46]。另一方面，也有研究尝试使稀疏性适应数据 。然而，尽管它们使用不同的策略来合并或选择键/值令牌，这些令牌是查询无关的，即它们被所有查询共享。尽管如此，根据预训练的ViT [15] 和DETR [1]的可视化，不同语义区域中的查询实际上关注的是相当不同的键-值对。因此，强制所有查询关注同一组令牌可能是次优的。

In this paper, we seek an attention mechanism with dynamic, query-aware sparsity. Basically, we aim for each query to attend to a small portion of the most semantically relevant key-value pairs. The first problem comes as how

在本文中，我们寻求一种具有动态、查询感知的稀疏注意力机制。基本上，我们希望每个查询都能关注一小部分与其在语义上最相关的键-值对。第一个问题是怎样

https://epfml.github.io/attention-cnn/

https://epfml.github.io/attention-cnn/

https://colab.research.google.com/github/ facebookresearch/detr/blob/colab/notebooks/detr\_ attention.ipynb to locate these key-value pairs to attend. For example, if we select key-value pairs in a per-query manner as done in [17], it still requires evaluation of pairwise affinity between all queries and keys, and hence has the same complexity of vanilla attention. Another possibility is to predict attention offsets based on local context for each query , and hence pairwise affinity computation is avoided. However, in this way, it is problematic to model long-range dependency [48].

https://colab.research.google.com/github/facebookresearch/detr/blob/colab/notebooks/detr\_attention.ipynb 来定位这些要关注的键-值对。例如，如果我们像 [17] 中那样以每个查询的方式选择键-值对，它仍然需要评估所有查询和键之间的成对亲和力，因此具有与原始注意力相同的复杂性。另一种可能性是基于每个查询的局部上下文预测注意力偏移 ，从而避免成对亲和力的计算。然而，这种方式在建模长距离依赖性 [48] 时存在问题。

Corresponding author.

对应作者。

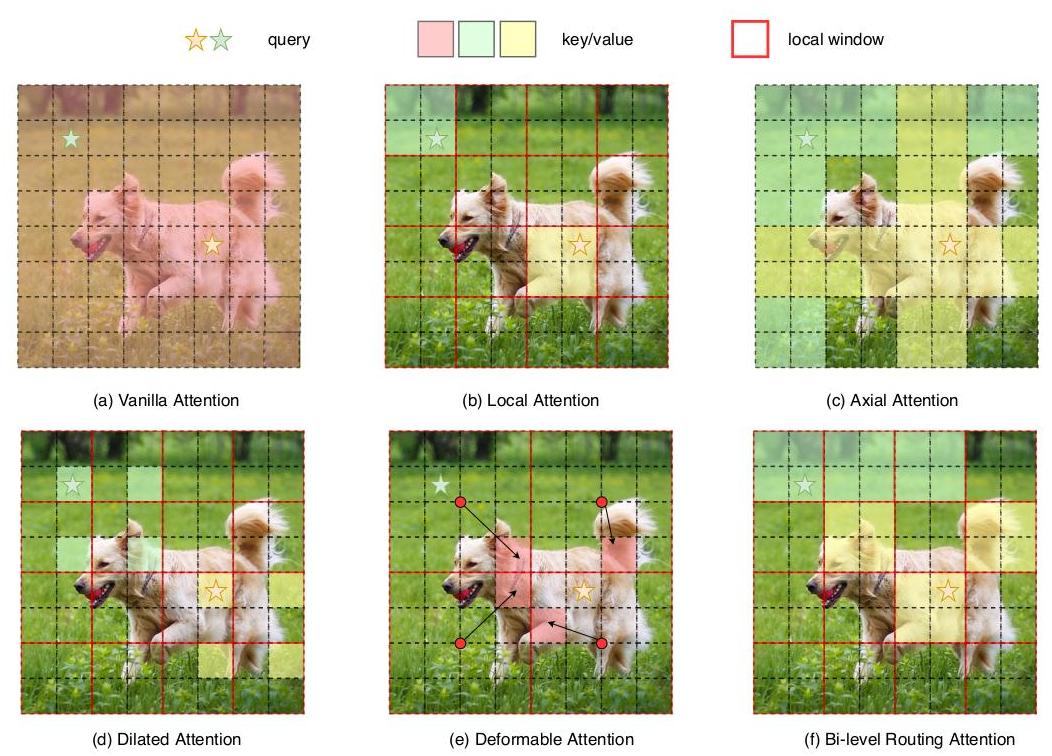


Figure 1. Vanilla attention and its sparse variants. (a) Vanilla attention operates gloabally and incurs high computational complexity and heavy memory footprint. (b)-(d) Several works attempt to alleviate the complexity by introducing sparse attention with different handcrafted patterns, such as local window , axial stripe , dilated window . (e) Deformable attention enables image-adaptive sparsity via deforming a regular grid. (f) We achieve dynamic, query-aware sparsity with bi-level routing attention, which first searches top- relevant regions, and then attends to the union of them.

图 1. 标准注意力及其稀疏变体。(a) 标准注意力全局操作，导致高计算复杂度和较大的内存占用。(b)-(d) 有几项工作尝试通过引入具有不同手工设计模式的稀疏注意力来减轻复杂度，例如局部窗口 、轴向条纹 、扩张窗口 。(e) 可变形注意力 通过变形规则网格实现图像自适应稀疏性。(f) 我们通过双层路由注意力实现动态、查询感知的稀疏性，首先搜索顶部 相关区域，然后关注它们的并集。

To locate valuable key-value pairs to attend globally with high efficiency, we propose a region-to-region routing approach. Our core idea is to filter out the most irrelevant key-value pairs at a coarse-grained region level, instead of directly at the fine-grained token level. This is done by first constructing a region-level affinity graph and then pruning it to keep only top- connections for each node. Hence, each region only needs to attend to the top- routed regions. With the attending regions determined, the next step is to apply token-to-token attention, which is non-trivial as key-value pairs are now assumed to be spatially scattered. For this case, while the sparse matrix multiplication is applicable, it is inefficient in modern GPUs, which rely on coalesced memory operations, i.e., accessing blocks of dozens of contiguous bytes at once [31]. Instead, we propose a simple solution via gathering key/value tokens, where only hardware-friendly dense matrix multiplications are involved. We refer to this approach as Bi-level Routing Attention (BRA), as it contains a region-level routing step and a token-level attention step.

为了高效地定位有价值的关键-值对进行全局关注，我们提出了一种区域到区域的路由方法。我们的核心思想是在粗粒度的区域级别过滤掉最不相关关键-值对，而不是直接在细粒度的令牌级别进行。这是通过首先构建区域级别的亲和图，然后剪枝以保留每个节点仅有的顶部 连接来实现的。因此，每个区域只需要关注顶部 路由区域。确定了关注区域后，下一步是应用令牌到令牌的注意力，这在假设关键-值对在空间上分散的情况下并非易事。对于这种情况，虽然稀疏矩阵乘法适用，但在现代GPU上效率低下，因为它们依赖于合并的内存操作，即一次访问数十个连续字节的块 [31]。相反，我们提出了一种通过收集键/值令牌的简单解决方案，其中只涉及对硬件友好的密集矩阵乘法。我们称这种方法为双层路由注意力（BRA），因为它包含一个区域级别的路由步骤和一个令牌级别的注意力步骤。

By using BRA as the core building block, we propose BiFormer, a general vision transformer backbone that can be used for many applications such as classification, object detection, and semantic segmentation. As BRA enables Bi-Former to attend to a small subset of the most relevant key/- value tokens for each query in a content-aware manner, our model achieves a better computation-performance trade-off. For example, with 4.6G FLOPs computation, BiFormer-T achieves top-1 accuracy on ImageNet-1K classification, which is the best as far as we know under similar computation budgets without training with external data or distillation . The improvements are also consistently shown in downstream tasks such as instance segmentation and semantic segmentation.

通过使用BRA作为核心构建块，我们提出了BiFormer，这是一种通用的视觉变换器主干网络，可以用于多种应用，如分类、目标检测和语义分割。由于BRA使得Bi-Former能够以内容感知的方式关注每个查询的最相关键/值令牌的子集，我们的模型实现了更好的计算性能权衡。例如，在4.6G FLOPs计算下，BiFormer-T在ImageNet-1K分类上达到了 的top-1准确度，据我们所知，在相似的计算预算下，不使用外部数据训练或蒸馏 的情况下，这是最佳的。这些改进在下游任务如实例分割和语义分割中也一致地显示出来。

To summarize, our contributions are as follows. We introduce a novel bi-level routing mechanism to vanilla attention, which enables content-aware sparse patterns in a query-adaptive manner. Using the bi-level routing attention as the basic building block, we propose a general vision transformer named BiFormer. Experimental results on various computer vision tasks including image classification, object detection, and semantic segmentation show that the proposed BiFormer achieves significantly better performances over the baselines under similar model sizes.

总结来说，我们的贡献如下。我们引入了一种新颖的双层路由机制到普通注意力中，它使得内容感知的稀疏模式以查询适应的方式生效。使用双层路由注意力作为基本构建块，我们提出了名为BiFormer的通用视觉变换器。在各种计算机视觉任务上的实验结果，包括图像分类、目标检测和语义分割，都表明在相似的模型大小下，所提出的BiFormer比基线实现了显著更好的性能。

# 2. Related Works

# 2. 相关工作

Vision transformers. Transformers are a family of neural networks that adopt channel-wise MLP blocks for per-location embedding (channel mixing) and attention [42] blocks for cross-location relation modeling (spatial mixing). Transformers were originally proposed for natural language processing and then introduced to computer vision by pioneering works such as DETR [1] and ViT [15]. In comparison with CNNs, the biggest difference is that transformers use attention as an alternative to convolution to enable global context modeling. However, as vanilla attention computes pairwise feature affinity across all spatial locations, it incurs a high computation burden and heavy memory footprints, especially for high-resolution inputs. Hence, an important research direction is to seek more efficient attention mechanisms.

视觉变换器。变换器是一系列神经网络，它们采用逐位置嵌入（通道混合）的通道MLP块和注意力 [42] 块来建模位置间的关系（空间混合）。变换器最初被提出用于自然语言处理 ，然后由DETR [1] 和ViT [15] 等开创性工作引入计算机视觉领域。与卷积神经网络（CNNs）相比，最大的不同是变换器使用注意力作为卷积的替代，以实现全局上下文建模。然而，由于普通注意力计算所有空间位置之间的特征亲和力，它在处理高分辨率输入时产生了高计算负担和沉重的内存占用。因此，一个重要的研究方向是寻找更高效的注意力机制。

Efficient attention mechanisms. A large volume of works have been proposed to reduce the computation and memory complexity bottlenecks of vanilla attention by utilizing sparse connection patterns [6], low-rank approximations [43] or recurrent operations [11]. A thorough survey of these attention variants can be found at [39]. In the scope of vision transformers, sparse attention gains its popularity recently due to the tremendous success of Swin transformer [29]. In Swin transformer, attention is restricted to non-overlapping local windows, and the shift window operation is introduced to enable inter-window communication between adjacent windows. To enable larger and even quasi-global receptive fields under a reasonable computation budget, several follow-up works introduce different handcrafted sparse patterns, such as dilated windows [41, 46] or cross-shaped windows [14]. There are also works that try to make the sparse pattern adaptive to data, such as DAT [48], TCFormer [53] and DPT [5]. While these works reduce the number of key/value tokens via different merging or selection strategies, these key/value tokens are shared by all queries on an image. Instead, we explore query-aware key/value token selection. The key observation which motivates our work is that the attentive region for different queries may differ significantly according to the visualization of pretrained ViT [15] and DETR [1]. As we achieve the goal of query-adaptive sparsity in a coarse-to-fine manner, it shares some similarities with quad-tree attention [38]. Different from quad-tree attention, the goal of our bi-level routing attention is to locate a few most relevant key-value pairs, while quad-tree attention builds a token pyramid and assembles messages from all levels of different granularities. In addition, the quad-tree requires deep recursion to cover the whole feature map, which hurts parallelism, while our bi-level routing attention can be more efficiently implemented by key/value token gathering, followed by dense matrix multiplications. As a result, quad-tree transformer is much slower than our BiFormer.

高效注意力机制。已经提出了大量工作来通过利用稀疏连接模式 [6]、低秩近似 [43] 或循环操作 [11] 减少原生注意力的计算和内存复杂性瓶颈。关于这些注意力变体的详尽调查可以在 [39] 中找到。在视觉变换器范围内，由于 Swin 变换器 [29] 的巨大成功，稀疏注意力最近变得流行。在 Swin 变换器中，注意力被限制在非重叠的局部窗口内，并且引入了移窗操作以实现相邻窗口之间的通信。为了在合理的计算预算下实现更大甚至类全局的感受野，一些后续工作引入了不同的手工设计的稀疏模式，如扩张窗口 [41, 46] 或十字形窗口 [14]。还有一些工作尝试使稀疏模式适应数据，如 DAT [48]、TCFormer [53] 和 DPT [5]。虽然这些工作通过不同的合并或选择策略减少了 key/value 标记的数量，但这些 key/value 标记被图像上的所有查询共享。相反，我们探索了查询感知的 key/value 标记选择。激发我们工作的关键观察是，根据预训练的 ViT [15] 和 DETR [1] 的可视化，不同查询的注意力区域可能存在显著差异。由于我们在粗到细的方式中实现了查询自适应的稀疏性，它与四叉树注意力 [38] 有一些相似之处。与四叉树注意力不同，我们的双层路由注意力的目标是定位几个最相关的 key-value 对，而四叉树注意力构建了一个标记金字塔，并从不同粒度的所有级别组装信息。此外，四叉树需要深度递归来覆盖整个特征图，这损害了并行性，而我们的双层路由注意力可以通过 key/value 标记收集后跟密集矩阵乘法来更有效地实现。因此，四叉树变换器比我们的 BiFormer 慢得多。

# 3. Our Approach: BiFormer

# 3. 我们的方法：BiFormer

This section elaborates the proposed approach. We start by briefly summarizing the attention mechanism in Section 3.1. We then introduce our novel bi-level routing attention (BRA) mechanism, which enables dynamic and query-adaptive sparsity, in Section 3.2. We further show that BRA can achieve complexity with a proper region partition size in Section 3.3. Finally, using BRA as the core building block, we present a new hierarchical vision transformer, named BiFormer, in Section 3.4.

本节详细阐述了我们提出的方法。我们首先在3.1节简要总结了注意力机制。然后在3.2节介绍了我们新颖的双层路由注意力（BRA）机制，该机制能够实现动态和查询自适应的稀疏性。进一步地，我们在3.3节展示了BRA在适当的区域划分大小下能够达到 复杂度。最后，在3.4节中，我们将BRA作为核心构建块，提出了一种新的分层视觉变换器，名为BiFormer。

# 3.1. Preliminaries: Attention

# 3.1.预备知识：注意力

Taking queries , keys , and values as input, an attention function transforms each query as a weighted sum of values, where the weights are computed as normalized dot products between the query and corresponding keys. It can be formally defined in a compact matrix form, as:

输入查询 、键 和值 ，注意力函数将每个查询转换为一个值的加权和，其中权重是查询与相应键之间归一化点积的计算结果。它可以正式地定义为一个紧凑的矩阵形式，如下：

Here, the scalar factor is introduced to avoid concentrated weights and gradient vanishing [42].

在这里，引入标量因子 以避免权重的集中和梯度消失 [42]。

In transformers, the de facto building block used is multihead self-attention (MHSA). By "self-attention", it means that queries , keys and values are derived as linear projections of the same input . (For vision transformers, is a spatially flattened feature map, i.e., , where and are the height and width, respectively, of the feature map.) As for "multi-head", it implies splitting the output into chunks (i.e., heads) along the channel dimension with each chunk using an independent group of projection weights. Formally,

在变换器中，实际使用的默认构建块是多头自注意力（MHSA）。所谓的“自注意力”意味着查询 、键 和值 是同一输入 的线性投影。（对于视觉变换器， 是一个空间展平的特征图，即 ，其中 和 分别是特征图的高度和宽度。）至于“多头”，它意味着沿通道维度将输出拆分为 个块（即头），每个块使用一组独立的投影权重。

where head is the output of the attention head. are corresponding input projection weights. An extra linear transformation with weight matrix is used to compose all heads.

其中头部 是 注意力头的输出。 是相应的输入投影权重。使用带有权重矩阵 的额外线性变换来组合所有头部。

MHSA has a complexity of , as there are queries and each query will attend to key-value pairs. Such a high complexity causes severe scalability issues w.r.t. the spatial resolution of the inputs.

MHSA 的复杂度为 ，因为有 个查询，每个查询都会关注 个键-值对。这种高复杂度导致在输入的空间分辨率方面出现了严重的可扩展性问题。

Algorithm 1 Pseudocode of BRA in a PyTorch-like style.

#input: features . Assume .

# output: features .

# S: square root of number of regions.

# : number of regions to attend.

#patchify input

patchify

# linear projection of query, key, value

query, key, value inear\_qkv .chunk

# regional query and key

query\_r, key\_r = query.mean (dim=1), key.mean (dim=1)

#adjacency matrix for regional graph

A\_r = mm(query\_r, key\_r.transpose )

# compute index matrix of routed regions

I\_r = topk . index

# gather key-value pairs

key\_g = gather(key, I\_r) # (S2̂, kHW/S2̂, C)

value\_g gather (value, ) #

# token-to-token attention

output dwconv (value)

# recover to shape

output unpatchify (output, patch\_size )

bmm: batch matrix multiplication; mm: matrix multiplication. dwconv: depthwise convolution.

# 3.2. Bi-Level Routing Attention (BRA)

# 3.2. 双层路由注意力（BRA）

To mitigate the scalability issue of MHSA, several works propose different sparse attention mechanisms, in which each query attends to only a small number of key-value pairs instead of all. However, these existing works either use handcrafted static patterns or share the sampled subset of key-value pairs among all queries, as shown in Figure 1. In this work, we explore a dynamic, query-aware sparse attention mechanism. Our key idea is to filter out most irrelevant key-value pairs in a coarse region level so that only a small portion of routed regions remain. We then apply fine-grained token-to-token attention in the union of these routed regions. To simplify the notations, we discuss the case of single-head self-attention with a single input, although we use multi-head self-attention [42] with batched input in practice. The whole algorithm is summarized with Pytorch-like [32] pseudo code in Algorithm 1. We give a detailed explanation as follows.

为了缓解 MHSA 的可扩展性问题，一些工作 提出了不同的稀疏注意力机制，在这些机制中，每个查询只关注少数键-值对，而不是全部。然而，这些现有工作要么使用手工设计的静态模式，要么如图1所示，在所有查询之间共享采样到的键-值对子集。在这项工作中，我们探索了一种动态的、查询感知的稀疏注意力机制。我们的关键想法是在粗略的区域级别过滤掉大多数不相关的键-值对，以便只有一小部分路由区域保留下来。然后，我们在这些路由区域的并集上应用细粒度的 token-to-token 注意力。为了简化符号，我们讨论了单头自注意力的单个输入情况，尽管在实践中我们使用带有批量输入的多头自注意力 [42]。整个算法在算法1中以 Pytorch-like [32] 伪代码进行了总结。我们接下来进行详细解释。

Region partition and input projection. Given a 2D input feature map , we start by dividing it into non-overlapped regions such that each region contains feature vectors. This step is done by reshaping as . We then derive the query, key, value tensor, , with linear projections:

区域划分和输入投影。给定一个2D输入特征图 ，我们首先将其划分为 个不重叠的区域，使得每个区域包含 个特征向量。这一步是通过将 重塑为 来完成的。然后，我们通过线性投影得到查询、键、值张量 。

where are projection weights for the query, key, value, respectively.

其中 分别是查询、键、值的投影权重。

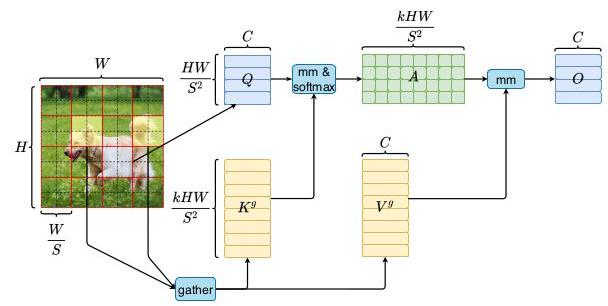


Figure 2. By gathering key-value pairs in top related windows, we utilize the sparsity to skip computations in the most irrelevant regions, while only GPU-friendly dense matrix multiplications are involved.

图 2。通过在顶部 相关窗口中收集键-值对，我们利用稀疏性来跳过最不相关区域的计算，同时只涉及 GPU 友好的密集矩阵乘法。

Region-to-region routing with directed graph. We then find the attending relationship (i.e., the regions that should be attended for each given region) by constructing a directed graph. Specifically, we first derive region-level queries and keys, , via applying per-region average on and , respectively. We then derive the adjacency matrix, , of region-to-region affinity graph via matrix multiplication between and transposed :

区域到区域的路由使用有向图。然后我们通过构建有向图来寻找关注关系（即每个给定区域应该关注的区域）。具体来说，我们首先通过在每个区域上应用平均操作分别得到区域级别的查询和键 。然后我们通过 和转置的 之间的矩阵乘法得到区域间亲和图的邻接矩阵 ：

Entries in the adjacency matrix, , measure how much two regions are semantically related. The core step that we perform next is to prune the affinity graph by keeping only top- connections for each region. Specifically, we derive a routing index matrix, , with the row-wise topk operator:

邻接矩阵中的条目 衡量两个区域在语义上有多么相关。我们接下来要执行的核心步骤是通过仅保留每个区域的前 个连接来修剪亲和图。具体来说，我们通过行向 top-k 操作得到一个路由索引矩阵 ：

Hence, the row of contains indices of most relevant regions for the region.

因此， 的 行包含 区域的最相关区域的索引。

Token-to-token attention. With the region-to-region routing index matrix , we can then apply fine-grained token-to-token attention. For each query token in region , it will attend to all key-value pairs residing in the union of routed regions indexed with . However, it is non-trivial to implement this step efficiently, as these routed regions are expected to be scattered over the whole feature map, while modern GPUs rely on coalesced memory operations that load blocks of dozens of contiguous bytes at once. We thus gather key and value tensor first, i.e.,

Token-to-token 注意力。有了区域到区域的路由索引矩阵 ，我们可以应用细粒度的 token-to-token 注意力。对于区域 中的每个查询 token，它将关注所有位于 路由区域联合体中的键-值对，这些区域由 索引。然而，由于这些路由区域预计会散布在整个特征图上，而现代 GPU 依赖于一次性加载数十个连续字节的合并内存操作，因此高效实现这一步并非易事。因此，我们首先收集键和价值张量，即

where are gathered key and value tensor. We can then apply attention on the gathered key-value pairs as:

其中 是聚集的关键和价值张量。然后我们可以在收集到的键-值对上应用注意力，如下所示：

Here, we introduce a local context enhancement term as in [37]. Function is parametrized with a depth-wise convolution, and we set the kernel size to 5 .在这里，我们引入了一个局部上下文增强项 ，如文献 [37] 中所示。函数 通过深度卷积进行参数化，并且我们将卷积核大小设置为 5。

# 3.3. Complexity Analysis of BRA

# 3.3. BRA的复杂性分析

The proposed bi-level routing attention enables direct long-range dependency modeling similar to vanilla attention. However, we show here that BRA has a much lower complexity of with a proper region partition factor compared to vanilla attention, which has a complexity of , and to quasi-global axial attention , which has a complexity of .

所提出的双层路由注意力能够像原生注意力一样直接建模长距离依赖。然而，我们在这里证明，在适当的区域划分因子 下，BRA的复杂度为 ，相比原生注意力的复杂度 和准全局轴向注意力的复杂度 ，要低得多，后两者的复杂度分别为 。

The computation of BRA consists of three parts: linear projection, region-to-region routing, and token-to-token attention. The total amount of computations is therefore:

BRA的计算包括三个部分：线性投影、区域到区域的路由和令牌到令牌的注意力。因此，总的计算量为：

where is the token embedding dimension (i.e., number of channels of the feature map), and is the number of regions to attend (" " in "top- "). Here, the inequality of arithmetic and geometric means has been applied. The equality in Eq. 8 holds if and only if . Therefore:

其中 是令牌嵌入维度（即特征图的通道数）， 是要关注的区域数（"top- " 中的 " "）。在这里，已经应用了算术平均数和几何平均数的不等式。等式 8 中的等号成立当且仅当 。因此：

In other words, BRA achieves complexity if we scale the region partition factor w.r.t. the input resolution according to Eq. 9.

换句话说，如果我们根据等式 9 按比例调整区域划分因子 以适应输入分辨率，BRA就能实现 的复杂度。

# 3.4. Architecture Design of BiFormer

# 3.4. BiFormer的架构设计

Using BRA as a basic building block, we propose a new general vision transformer, BiFormer. As shown in Figure 3 , we follow the recent state-of-the-art vision transformers to use a four-stage pyramid structure. Specifically, in stage , we use an overlapped patch embedding in the first stage and a patch merging module in the second to fourth stages to reduce the input spatial resolution while increasing the number of channels, followed by consecutive BiFormer blocks to transform the features. In each BiFormer block, we follow recent works [7, 25, 41] to use a depthwise convolution at the beginning to encode relative position information implicitly. We then apply a BRA module and 2-layer MLP module with expansion ratio sequentially for cross-location relation modeling and per-location embedding, respectively.

使用 BRA 作为基本构建块，我们提出了一种新的通用视觉变换器，BiFormer。如图 3 所示，我们遵循最近最先进的视觉变换器 使用四阶段金字塔结构。具体来说，在阶段 中，我们在第一阶段使用重叠的补丁嵌入，并在第二至第四阶段使用补丁合并模块 以降低输入空间分辨率的同时增加通道数，然后是 连续的 BiFormer 块来转换特征。在每个 BiFormer 块中，我们遵循最近的工作 [7, 25, 41] 在开头使用 深度卷积隐式地编码相对位置信息。然后我们依次应用 BRA 模块和 2 层 MLP 模块，其扩展比为 ，分别用于跨位置关系建模和每个位置嵌入。

We instantiate BiFormer with 3 different model sizes by scaling the network width (i.e., the number of base channels ) and depth (i.e., the number of BiFormer blocks used in each stage, , as listed in Table 1. They share other configurations. We set each attention head to 32 channels, and MLP expansion ratio . For BRA, we use topk for the 4 stages, and region partition factor for classification/semantic segmentation/object detection task, due to different input resolutions.

我们通过调整网络宽度（即基础通道数 ）和深度（即每个阶段使用的 BiFormer 块数量 ，如表 1 所示）实例化了三种不同大小的 BiFormer 模型。它们共享其他配置。我们将每个注意力头设置为 32 个通道，MLP 扩展比为 。对于 BRA，我们在四个阶段中使用 topk ，并对分类/语义分割/目标检测任务使用区域划分因子 ，因为输入分辨率不同。

| Models | #Channels. | #Blocks | Params | FLOPs |
| --- | --- | --- | --- | --- |
| BiFormer-T | 64 |  | 13M | 2.2G |
| BiFormer-S | 64 |  | 26M | 4.5G |
| BiFomrer-B | 96 |  | 57M | 9.8G |

Table 1. Network width and depth of different model variants. The FLOPs are calculated with input.

表 1。不同模型变体的网络宽度和深度。FLOPs 是使用 输入计算的。

| Model | FLOPs (G) | Params (M) | Top-1 Acc. (%) |
| --- | --- | --- | --- |
| ResNet-18 [19] | 1.8 | 11.7 | 69.8 |
| RegNetY-1.6G [34] | 1.6 | 11.2 | 78.0 |
| PVTv2-b1 [45] | 2.1 | 13.1 | 78.7 |
| Shunted-T [37] | 2.1 | 11.5 | 79.8 |
| QuadTree-B-b1 [38] | 2.3 | 13.6 | 80.0 |
| BiFormer-T | 2.2 | 13.1 | 81.4 |
| Swin-T [29] | 4.5 | 29 | 81.3 |
| CSWin-T [14] | 4.5 | 23 | 82.7 |
| DAT-T [48] | 4.6 | 29 | 82.0 |
| CrossFormer-S [46] | 5.3 | 31 | 82.5 |
| RegionViT-S [2] | 5.3 | 31 | 82.6 |
| QuadTree-B-b2 [38] | 4.5 | 24 | 82.7 |
| MaxViT-T [41] | 5.6 | 31 | 83.6 |
| ScalableViT-S [50] | 4.2 | 32 | 83.1 |
| Uniformer-S\* | 4.2 | 24 | 83.4 |
| Wave-ViT-S\* [51] | 4.7 | 23 | 83.9 |
| BiFormer-S | 4.5 | 26 | 83.8 |
| BiFormer-S\* | 4.5 | 26 | 84.3 |
| Swin-B [29] | 15.4 | 88 | 83.5 |
| CSWin-B [14] | 15.0 | 78 | 84.2 |
| CrossFormer-L [46] | 16.1 | 92 | 84.0 |
| ScalableViT-B [50] | 8.6 | 81 | 84.1 |
| Uniformer-B\* [25] | 8.3 | 50 | 85.1 |
| Wave-ViT-B\* [51] | 7.2 | 34 | 84.8 |
| BiFormer-B | 9.8 | 57 | 84.3 |
| BiFormer-B\* | 9.8 | 58 | 85.4 |

Table 2. Comparison of different backbones on ImageNet-1K. All models are trained and evaluated on images of resolution . "\*" indicates that the model is trained with token labeling [23]. Methods are grouped by the amount of computations.

表 2。不同主干网络在 ImageNet-1K 上的比较。所有模型都在分辨率 的图像上进行训练和评估。"\*" 表示模型是使用标记标签 [23] 进行训练的。方法按计算量分组。

In the final stage, topk means that we use full self-attention.

在最终阶段，topk 意味着我们使用完全自注意力。

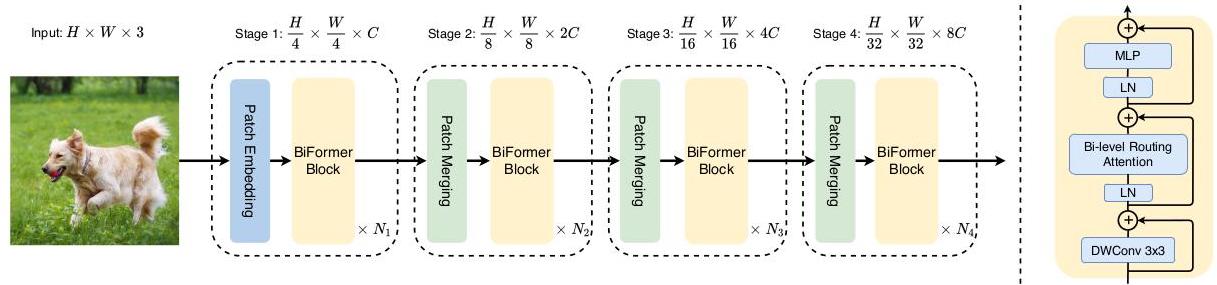


Figure 3. Left: The overall architecture of our BiFormer. Refer to Table 1 for configurations. Right: Details of a BiFormer Block.

图 3。左侧：我们 BiFormer 的整体架构。配置详情请见表 1。右侧：BiFormer 块的细节。

# 4. Experiments

# 4. 实验

We evaluate the effectiveness of our proposed BiFormer experimentally on a series of mainstream computer vision tasks including image classification (Sec. 4.1), object detection and instance segmentation (Sec. 4.2), and semantic segmentation (Sec. 4.3). Specifically, we train from scratch on ImageNet1K [12] for image classification. We then fine-tune the pretrained backbones on COCO [28] for object detection and instance segmentation, and on ADE20K [55] for semantic segmentation. Additionally, we conduct ablation study to verify the effectiveness of the proposed bi-level routing attention and other architecture design choices of BiFormer in Sec. 4.4. Finally, to verify query-adaptive, sparse patterns are achieved by bi-level routing attention, we visualize the attention map in Sec. 4.5.

我们通过实验评估了我们提出的 BiFormer 在一系列主流计算机视觉任务上的有效性，包括图像分类（第 4.1 节）、目标检测和实例分割（第 4.2 节）以及语义分割（第 4.3 节）。具体来说，我们在 ImageNet1K [12] 上从头开始训练用于图像分类。然后，我们在 COCO [28] 上微调预训练的主干网络用于目标检测和实例分割，在 ADE20K [55] 上微调用于语义分割。此外，我们在第 4.4 节进行了消融研究，以验证所提出的双级路由注意力和 BiFormer 的其他架构设计选择的有效性。最后，为了验证查询自适应，稀疏模式是通过双级路由注意力实现的，我们在第 4.5 节可视化了注意力图。

# 4.1. Image Classification on ImageNet-1K

# 4.1. 在 ImageNet-1K 上的图像分类

Settings. We conduct image classification experiments on the ImageNet-1K [12] dataset, following the experimental settings of DeiT [40] for fair comparison. Specifically, each model is trained 300 epochs with input size of . We take AdamW as the optimizer with weight decay of 0.05 , and apply cosine decay learning rate schedule with an initial learning rate of 0.001 , while the first 5 epochs are utilized for linear warm-up [16]. The batch size is set to 1024. To avoid overfitting, we apply regularization techniques including RandAugment [9] (rand-m9-mstd0.5- inc1), MixUp [54] (prob ), CutMix [52] (prob ), Random Erasing (prob ), and increasing stochastic depth [21] (prob for BiFormer-T/S/B, respectively). To fairly compare the models trained with token labeling [23], including Uniformer [25] and Wave-ViT [51], we also provide a version trained with the same recipe provided by WaveViT.

设置。我们在ImageNet-1K [12]数据集上进行图像分类实验，遵循DeiT [40]的实验设置以进行公平比较。具体来说，每个模型在输入尺寸为 的情况下训练300个周期。我们使用AdamW优化器，权重衰减为0.05，并应用余弦衰减学习率计划，初始学习率为0.001，而前5个周期用于线性预热[16]。批量大小设置为1024。为了避免过拟合，我们应用了正则化技术，包括RandAugment [9]（rand-m9-mstd0.5-inc1）、MixUp [54]（概率 ）、CutMix [52]（概率 ）、随机擦除（概率 ）以及增加随机深度[21]（BiFormer-T/S/B的概率分别为 ）。为了公平比较使用标记训练的模型[23]，包括Uniformer [25]和Wave-ViT [51]，我们还提供了一个使用WaveViT提供的相同配方训练的版本。

Results. We compare our method with several closely related methods and/or recent state-of-the-arts. Quantitative results are listed in Table 2, where models are grouped by the amount of computations (FLOPs). In all 3 groups, our model consistently outperforms other compared ones. For example, for models in the smallest group ( FLOPs), our BiFormer-T achieves top-1 accuracy, better than the most competitive QuadTree-b1 [38]. For models in the second group ( FLOPs), BiFormer-S achieves top-1 accuracy. To the best of our knowledge, this is the best result without extra training data or training tricks. In addition, using the distillation technique named token labeling [23], the accuracy of BiFormer-S can be further boosted to , which implies that there is a huge potential for the proposed architecture. For models in the largest group ( FLOPs), BiFormer-B achieves an even better performance than those of existing models with the amount of computations reaching up to FLOPs, such as Swin-B [29], CSWin-B [14] and CrossFormer-L [46].

结果。我们将我们的方法与几个密切相关的方法以及/或最近的最先进技术进行了比较。定量结果列在表2中，其中模型根据计算量（FLOPs）分组。在所有3个组别中，我们的模型始终优于其他对比的模型。例如，对于最小组别（ FLOPs）中的模型，我们的BiFormer-T达到了 top-1准确度， 优于最具竞争力的QuadTree-b1 [38]。对于第二组别（ FLOPs）中的模型，BiFormer-S达到了 top-1准确度。据我们所知，这是在没有额外训练数据或训练技巧的情况下最好的结果。此外，使用名为token labeling [23]的蒸馏技术，BiFormer-S的准确度可以进一步提升到 ，这表明所提出架构具有巨大的潜力。对于最大组别（ FLOPs）中的模型，BiFormer-B在计算量达到 FLOPs时，甚至比现有的模型性能更佳，如Swin-B [29]、CSWin-B [14]和CrossFormer-L [46]。

# 4.2. Object Detection and Instance Segmentation

# 4.2. 目标检测与实例分割

Settings. We evaluate the models for object detection and instance segmentation on COCO 2017 [28]. For a fair comparison, all experiments are conducted with the MMDetec-tion [3] toolbox. RetinaNet [27] and Mask R-CNN [18] frameworks are used for object detection and instance segmentation, respectively. Before training on COCO, we initialize the backbone with weights pretrained on ImageNet- , while leaving all other layers randomly initialized. The models are trained with the standard schedule (12 epochs) provided by MMDetection, except that we use the AdamW optimizer [30], instead of SGD. We use an initial learning rate of , and a batch size of 16, while the weight decay is set as and for RetinaNet and Mask R-CNN, respectively. During training, we resize the input images by fixing the shorter side to 800 pixels while keeping the longer side not exceeding 1,333 pixels.

设置。我们在 COCO 2017 [28] 上评估了对象检测和实例分割模型。为了公平比较，所有实验都使用 MMDetection [3] 工具箱进行。RetinaNet [27] 和 Mask R-CNN [18] 框架分别用于对象检测和实例分割。在 COCO 上训练之前，我们使用在 ImageNet- 上预训练的权重初始化主干网络，而其他所有层则随机初始化。模型按照 MMDetection 提供的标准 训练计划（12 个周期）进行训练，不同之处在于我们使用 AdamW 优化器 [30]，而不是 SGD。我们使用初始学习率 ，批量大小为 16，权重衰减对 RetinaNet 和 Mask R-CNN 分别设置为 和 。在训练过程中，我们将输入图像的短边固定为 800 像素，同时保持长边不超过 1,333 像素。

Results. We list results in Table 3. For object detection with RetinaNet, we report mean Average Precision , Average Precision at different IoU thresholds , ) and for three object sizes (i.e. small, medium, and large (S/M/L)). From the results, we can see that while the overall performance of BiFormer is only comparable to some most competitive existing methods, such as WaveViT and QuadTree-B, the performance on small objects outperforms these methods significantly. This may be because the BRA saves computations via sparse sampling instead of downsampling. Hence, it preserves fine-grained details, which are crucial for small objects. For instance segmentation with Mask R-CNN, we report bounding box and mask Average Precision and at different IoU thresholds . As shown in Table 3, our method shows a clear advantage in this task on all metrics.

结果。我们在表3中列出结果。对于使用RetinaNet的目标检测，我们报告了平均平均精度 、在不同IoU阈值 的平均精度 （ ），以及对于三种对象大小（即小、中、大（S/M/L））。从结果中我们可以看到，尽管BiFormer的整体性能仅与一些最具竞争力的现有方法，如WaveViT和QuadTree-B相当，但在小对象上的性能 显著优于这些方法。这可能是因为BRA通过稀疏采样而不是下采样来节省计算。因此，它保留了对于小对象至关重要的细粒度细节。对于使用Mask R-CNN的实例分割，我们报告了不同IoU阈值下的边界框和掩码平均精度 和 。如表3所示，我们的方法在所有指标上均显示出明显优势。

| Backbone | RetinaNet schedule | | | | | | Mask R-CNN 1× schedule | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| Swin-T [29] | 41.5 | 62.1 | 44.2 | 25.1 | 44.9 | 55.5 | 42.2 | 64.6 | 46.2 | 39.1 | 61.6 | 42.0 |
| DAT-T [48] | 42.8 | 64.4 | 45.2 | 28.0 | 45.8 | 57.8 | 44.4 | 67.6 | 48.5 | 40.4 | 64.2 | 43.1 |
| CSWin-T [14] | - | - | - | - | - | - | 46.7 | 68.6 | 51.3 | 42.2 | 65.6 | 45.4 |
| CrossFormer-S [46] | 44.4 | 55.3 | 38.6 | 19.3 | 40.0 | 48.8 | 45.4 | 68.0 | 49.7 | 41.4 | 64.8 | 44.6 |
| QuadTree-B2 [38] | 46.2 | 67.2 | 49.5 | 29.0 | 50.1 | 61.8 | - | - | - | - | - | - |
| WaveViT-S\* [51] | 45.8 | 67.0 | 49.4 | 29.2 | 50.0 | 60.8 | 46.6 | 68.7 | 51.2 | 42.4 | 65.5 | 45.8 |
| BiFormer-S | 45.9 | 66.9 | 49.4 | 30.2 | 49.6 | 61.7 | 47.8 | 69.8 | 52.3 | 43.2 | 66.8 | 46.5 |
| Swin-S [29] | 44.5 | 65.7 | 47.5 | 27.4 | 48.0 | 59.9 | 44.8 | 66.6 | 48.9 | 40.9 | 63.4 | 44.2 |
| DAT-S [48] | 45.7 | 67.7 | 48.5 | 30.5 | 49.3 | 61.3 | 47.1 | 69.9 | 51.5 | 42.5 | 66.7 | 45.4 |
| CSWin-S [14] | - | - | - | - | - | - | 47.9 | 70.1 | 52.6 | 43.2 | 67.1 | 46.2 |
| CrossFormer-B [46] | 46.2 | 67.8 | 49.5 | 30.1 | 49.9 | 61.8 | 47.2 | 69.9 | 51.8 | 42.7 | 66.6 | 46.2 |
| QuadTree-B3 [38] | 47.3 | 68.2 | 50.6 | 30.4 | 51.3 | 62.9 | - | - | - |  | - | - |
| Wave-ViT-B\* [51] | 47.2 | 68.2 | 50.9 | 29.7 | 51.4 | 62.3 | 47.6 | 69.1 | 52.4 | 43.0 | 66.4 | 46.0 |
| BiFormer-B | 47.1 | 68.5 | 50.4 | 31.3 | 50.8 | 62.6 | 48.6 | 70.5 | 53.8 | 43.7 | 67.6 | 47.1 |

Table 3. Comparison based on the object detection (left group) and instance segmentation (right group) tasks, on the COCO 2017 dataset.

表3。基于COCO 2017数据集的目标检测（左侧组）和实例分割（右侧组）任务的比较。

| Backbone | S-FPN | Upernet | |
| --- | --- | --- | --- |
|  |  |  | MS mIOU(%) |
| Swin-T [29] | 41.5 | 44.5 | 45.8 |
| DAT-T [48] | 42.6 | 45.5 | 46.4 |
| CSWin-T [14] | 48.2 | 49.3 | 50.7 |
| CrossFormer-S [46] | 46.0 | 47.6 | 48.4 |
| Shunted-S [37] | 48.2 | 48.9 | 49.9 |
| WaveViT-S\* [51] | - |  | 49.6 |
| BiFormer-S | 48.9 | 49.8 | 50.8 |
| Swin-S [29] | - | 47.6 | 49.5 |
| DAT-S [48] | 46.1 | 48.3 | 49.8 |
| CSWin-S [14] | 49.2 | 50.4 | 51.5 |
| CrossFormer-B [46] | 47.7 | 49.7 | 50.6 |
| Uniformer-B [25] | 48.0 | 50.0 | 50.8 |
| WaveViT-B\* [51] | - | - | 51.5 |
| BiFormer-B | 49.9 | 51.0 | 51.7 |

Table 4. Comparison based on semantic segmentation with two segmentation heads (Semantic FPN and UpperNet), on ADE20K.

表4。基于ADE20K数据集的语义分割，使用两个分割头（语义FPN和UpperNet）的比较。

# 4.3. Semantic Segmentation on ADE20K

# 4.3. 在ADE20K上的语义分割

Settings. Following existing works, we conduct our semantic segmentation experiments on the ADE20K [55] dataset based on MMSegmentation [8]. We do comparisons under both Semantic FPN [24] and UperNet [49] frameworks. In both cases, the backbone is initialized with ImageNet-1K pretrained weights, and other layers use random initialization. Models are optimized with the AdamW optimizer and the batch size is set as 32 . For a fair comparison, our Semantic FPN experiments use the same setting as PVT [44] to train the model steps. Our Upernet experiments use the same setting as Swin Transformer [29] to train the model iterations.

设置。遵循现有工作，我们在基于MMSegmentation [8] 的ADE20K [55] 数据集上进行语义分割实验。我们在 Semantic FPN [24] 和 UperNet [49] 架构下进行对比。在这两种情况下，主干网络使用 ImageNet-1K 预训练权重进行初始化，其他层使用随机初始化。模型使用 AdamW 优化器进行优化，批量大小设置为 32。为了公平比较，我们的 Semantic FPN 实验使用与 PVT [44] 相同的设置来训练模型 步。我们的 UperNet 实验使用与 Swin Transformer [29] 相同的设置来训练模型 次迭代。

| Sparse Attention | IN1K Top1(%) | ADE20K |
| --- | --- | --- |
| Sliding window [35] | 81.4 | - |
| Shifted window [29] | 81.3 | 41.5 |
| Spatially Sep [7] | 81.5 | 42.9 |
| Sequential Axial [20] | 81.5 | 39.8 |
| Criss-Cross [22] | 81.7 | 43.0 |
| Cross-shaped window [14] | 82.2 | 43.4 |
| Deformable [48] | 82.0 | 42.6 |
| Block-Grid [41] | 81.8 | 42.8 |
| Bi-level Routing | 82.7 | 44.8 |

Table 5. Ablation study on different attention mechanisms. All models follow the architecture design of the Swin-T model.

表 5。不同注意力机制的消融研究。所有模型遵循 Swin-T 模型的架构设计。

Results. Table 4 shows the results of the two different frameworks. It shows that with the Semantic FPN framework, our BiFormer-S/B achieves 48.9/49.9 mIoU, respectively, improving CSWin-T/S by 0.7 mIoU. A similar performance gain for the UperNet framework is also observed.

结果。表 4 显示了两种不同架构的结果。它显示，在 Semantic FPN 架构下，我们的 BiFormer-S/B 分别达到了 48.9/49.9 mIoU，比 CSWin-T/S 提高了 0.7 mIoU。在 UperNet 架构下也观察到了类似的性能提升。

# 4.4. Ablation Study

# 4.4. 消融研究

The effectiveness of BRA. We compare BRA with several existing sparse attention mechanisms. Following [14], we align macro architecture designs with Swin-T [29] for a fair comparison. Specifically, we use 2, 2, 6, 2 blocks for the four stages, non-overlapped patch embedding, set the initial patch embedding dimension and MLP expansion ratio . The results are reported in Table 5. Our bi-level routing attention has significantly better performance than existing sparse attention mechanisms, in terms of both image classification and semantic segmentation.

BRA 的有效性。我们将 BRA 与几种现有的稀疏注意力机制进行了比较。遵循 [14]，我们与 Swin-T [29] 对齐宏观架构设计以进行公平比较。具体来说，我们为四个阶段分别使用 2、2、6、2 个块，非重叠的补丁嵌入，设置初始补丁嵌入维度 和 MLP 扩展比例 。结果报告在表 5 中。我们的双层路由注意力在图像分类和语义分割方面，比现有的稀疏注意力机制具有显著更好的性能。

Other architecture design choices. Using the Swin-T layout as the baseline, we present a summary of other modifications that we have applied, which further boost our BiFormer-S model to state-of-the-art performances on the ImageNet-1K dataset. These modifications include: (1) replacing non-overlapped patch embedding [29] with overlapped one using deeper layout (i.e. stacking more blocks in each stage, while reducing the base channels from 96 to 64 and MLP expansion ratio from 4 to 3 to keep similar FLOPs.), (3) adding convolution position encoding at the beginning of the BiFormer blocks, and (4) applying token labeling training technique. As shown in Table 6, simply using a deeper layout can improve the performance significantly. However, this factor is usually not discussed in existing works.

其他架构设计选择。以Swin-T布局作为基线，我们总结了其他我们已经应用的修改，这些修改进一步将我们的BiFormer-S模型提升到ImageNet-1K数据集上的最新性能水平。这些修改包括：（1）将非重叠的补丁嵌入[29]替换为使用更深布局的重叠补丁嵌入 （即在每一阶段堆叠更多块，同时将基础通道数从96减少到64，并将MLP扩展比例从4降低到3以保持类似的浮点运算数FLOPs），（3）在BiFormer块的开始处添加卷积位置编码 ，以及（4）应用标记训练技术 。如表6所示，仅使用更深的布局就可以显著提高性能。然而，这个因素在现有作品中通常未被讨论。



Figure 4. Visualization of the attention maps for two scenes. For each scene, we visualize two query positions on the input image (left), corresponding routed regions (middle), and a final attention heatmap (right).

图4. 两个场景的注意力图可视化。对于每个场景，我们在输入图像（左）上可视化两个查询位置，相应的路由区域（中），以及最终的注意力热图（右）。

| Architecture design | Params (M) | FLOPs (G) | IN1K Top1 (%) |
| --- | --- | --- | --- |
| Baseline (Swin-T layout) | 29 | 4.6 | 82.7 |
| +Overlapped patch emb. | 31 | 4.9 |  |
| +Deeper layout | 25 | 4.5 |  |
| +Convolution pos. enc. | 26 | 4.5 |  |
| +Token Labling | 29 | 4.9 |  |

Table 6. Ablation path from Swin-T [29] layout architecture to BiFormer-S. Note that the modifications are applied sequentially.

表6. 从Swin-T [29]布局架构到BiFormer-S的消融路径。请注意，修改是按顺序应用的。

# 4.5. Visualization of Attention Map

# 4.5. 注意力图的可视化

To further understand how bi-level routing attention works, we visualize routed regions and attention response w.r.t. query positions. For this visualization, we use the routing indices and attention scores extracted from the final BiFormer block of the stage, which is the major stage consuming most computations. We demonstrate two scenes in Figure 4. In both cases, we can clearly observe that semantically related regions are successfully located. For example, in the first scene, which is a street view, if the query position is on a building or a tree, the corresponding routed regions cover the same or similar entities. In the second indoor scene, when we place the query position on the mouse, the routed regions contain part of the host, keyboard, and display, even though these regions are not adjacent to each other. This implies that our bi-level routing attention can capture long-range inter-object relationships.

为了进一步理解双层路由注意力是如何工作的，我们可视化路由区域和相对于查询位置的注意力响应。对于这种可视化，我们使用了从 阶段的最后一个BiFormer块中提取的路由索引和注意力分数，这个阶段是消耗计算量最大的主要阶段。我们在图4中展示了两种场景。在这两种情况下，我们都可以清楚地观察到成功定位了语义相关的区域。例如，在第一个场景中，这是一个街道视图，如果查询位置在建筑物或树上，相应的路由区域覆盖了相同的或类似的实体。在第二个室内场景中，当我们将查询位置放在鼠标上时，路由区域包含了主机、键盘和显示器的一部分，尽管这些区域彼此不相邻。这表明我们的双层路由注意力能够捕捉到长距离的物体间关系。

# 5. Limitation and Future Work

# 5. 局限性与未来工作

Compared to sparse attention with simple static patterns, we introduce an extra step to locate the regions to attend, where we build and prune a region-level graph and gather key-value pairs from the routed regions. While this step does not incur much computation as it operates at a coarse region level, it inevitably incurs extra GPU kernel launch and memory transactions. Hence, BiFormer has lower throughput than some existing models with similar FLOPs on GPU due to overheads of kernel launch and memory bottleneck. Nonetheless, this problem can be mitigated via engineering efforts, such as GPU kernel fusion. We will explore efficient sparse attention and vision transformer with hardware awareness in our future works.

与具有简单静态模式的空间稀疏注意力相比，我们引入了一个额外的步骤来定位要关注的区域，其中我们构建并修剪一个区域级图并从路由区域收集键值对。尽管这一步骤在粗略的区域级别上操作，不会引起太多的计算，但它不可避免地会导致额外的GPU核心启动和内存交易。因此，与在GPU上具有相似FLOPs的现有模型相比，BiFormer由于核心启动和内存瓶颈的额外开销而有较低的吞吐量。尽管如此，这个问题可以通过工程努力来缓解，例如GPU核心融合。在未来的工作中，我们将探索硬件感知的效率化稀疏注意力和视觉变换器。

# 6. Conclusion

# 6. 结论

We propose bi-level routing attention to enable efficient allocation of computations in a dynamic, query-aware manner. The core idea of BRA is to filter out the most irrelevant key-value pairs at a coarse region level. It is achieved by first building and pruning a region-level directed graph, and then applying fine-grained token-to-token attention in the union of routed regions. We have analyzed the computational complexity of BRA and demonstrated that it achieves with a proper region partition size. Using BRA as the core building block, we propose BiFormer, a new vision transformer that has shown superior performances on four popular vision tasks, image classification, object detection, instance segmentation, and semantic segmentation.

我们提出了双层路由注意力机制，以实现动态、查询感知的计算分配效率。BRA的核心思想是在粗略区域级别过滤掉最不相关的键值对。这是通过首先构建并修剪区域级别的有向图，然后在路由区域的并集上应用细粒度的token-to-token注意力实现的。我们已经分析了BRA的计算复杂度，并证明了在适当的区域划分大小下，它能够实现 。使用BRA作为核心构建块，我们提出了BiFormer，这是一种新的视觉变换器，已在四个流行的视觉任务中显示出优越性能：图像分类、目标检测、实例分割和语义分割。

# Appendix

# 附录

# A. Discussion on Regional Representations

# A. 关于区域表示的讨论

In our proposed bi-level routing attention, we derive the regional representations and with average pooling for region-to-region routing. We justify the choice here.

在我们提出的双层路由注意力中，我们通过平均池化推导区域表示 和 以进行区域到区域的路由。我们在这里为这一选择提供理由。

In fact, as the goal of region-to-region routing is to find the most related tokens for token-to-token attention in the next step, it is reasonable to maximize the average token-to-token affinity scores between the two regions. However, this is equivalent to maximizing the affinity score between the average tokens of the two regions, because

事实上，由于区域到区域路由的目标是在下一步中为token-to-token注意力找到最相关的tokens，因此最大化两个区域之间的平均token-to-token亲和力分数是合理的。然而，这等价于最大化两个区域的平均tokens之间的亲和力分数，因为

where we denote the set of token indices of the two regions with and .

其中，我们用 和 表示两个区域的token索引集合。

# B. Throughput Comparison

# B. 吞吐量比较

To demonstrate the computation efficiency of the proposed bi-level routing attention, we compare the through-puts of models using different attention mechanisms. Specifically, we replace the shift window attention modules in Swin-T [29] with quad-tree attention [38] modules to form QuadTree-STL, and with our bi-level routing attention modules to form BiFormer-STL. We then use the widely used timm [47] script to benchmark the training and inference throughput on a Tesla V100 GPU with a batch size of 128 and image resolution of .

为了证明所提出双层路由注意力的计算效率，我们比较了使用不同注意力机制的模型的吞吐量。具体来说，我们用四叉树注意力 [38] 模块替换了Swin-T [29] 中的移窗注意力模块，形成了QuadTree-STL，并用我们的双层路由注意力模块形成了BiFormer-STL。然后，我们使用广泛使用的timm [47] 脚本在配备 Tesla V100 GPU、批量大小为128和图像分辨率为 的情况下，对训练和推理吞吐量进行基准测试。

As shown in Figure 5, Swin-T has the highest throughput due to its simplicity. Switching to our bi-level routing attention(BRA), the training and inference throughput of BiFormer-STL decrease by and respectively in comparison with Swin-T. This is caused by extra GPU kernel launch and memory transactions caused by the routing process (i.e. locating the regions to attend and gather key-value pairs). Nonetheless, BiFormer-STL is still faster than QuadTree-STL. This is due to that on the one hand the recursive nature of quad-tree attention hurts the parallelism, on the other hand quad-tree attention relies on sparse matrix multiplications which are inefficient on GPUs, while our BRA can be efficiently implemented with key-value token gathering followed by GPU-friendly dense matrix multiplications.

如图5所示，由于Swin-T的简单性，其吞吐量最高。切换到我们的双级路由注意力（BRA）后，BiFormer-STL的训练和推理吞吐量分别比Swin-T减少了 和 。这是由路由过程（即定位要关注和收集键值对的区域）引起的额外GPU内核启动和内存事务造成的。尽管如此，BiFormer-STL仍然比QuadTree-STL快 倍。这是因为一方面四叉树注意力的递归性质损害了并行性，另一方面四叉树注意力依赖于在GPU上效率低下的稀疏矩阵乘法，而我们的BRA可以通过键值令牌收集后接GPU友好的密集矩阵乘法高效实现。

It is worth noting that, the overheads of both memory transactions and kernel launch incurred by the routing process can be reduced via engineering efforts such as GPU kernel fusion. We leave this optimization to our future work.

值得注意的是，通过工程努力（如GPU内核融合）可以减少路由过程引起的内存事务和内核启动的开销。我们将这个优化留到未来的工作中。

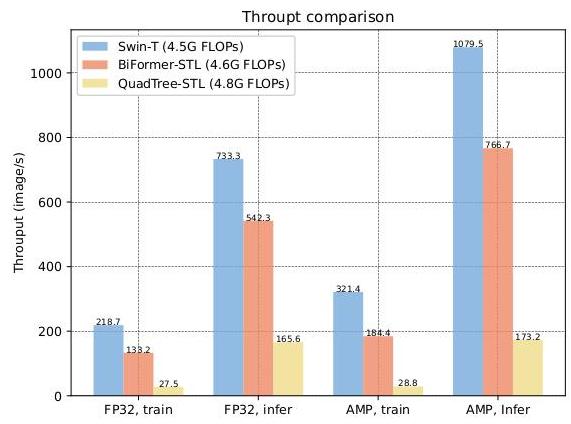


Figure 5. Throughput comparison on a 32GB Tesla V100 GPU. The suffix "STL" denotes Swin-T Layout, which means we use Swin-T [29] backbone with only attention module being replaced. We report results under both FP32 precision and automatic mixed precision (AMP) modes.

图5. 在32GB Tesla V100 GPU上的吞吐量比较。后缀“STL”表示Swin-T布局，这意味着我们使用仅替换注意力模块的Swin-T [29] 主干网络。我们在FP32精度和自动混合精度（AMP）模式下报告结果。

|  |  | #tokens to attend | Acc | im/s (FP32) |
| --- | --- | --- | --- | --- |
| 7 | 1,4,16,49 | 64,64,64,49 | 82.7 | 522.3 |
| 7 | 1,2,8,32 | 64,32,32,32 | 82.4 | 563.2 |
| 7 | 2,8,32,49 | 128,128,128,49 | 82.6 | 419.9 |
| 8,4,2,1 | 2,2,2,1 | 98,98,98,49 | 82.3 | 606.2 |

Table 7. Ablation study on top- and partition factor .

表7. 关于顶部 和分区因子 的消融研究。

# C. Choices of top- and partition factor

# C. 顶部 和分区因子 的选择

In the paper, and were chosen more with consideration of engineering issues. (1) is chosen as a divisor of the training size to avoid padding, which slows down the training and may also degrade the performance. For example, in image classification where the resolution is , we use so that it is a divisor of the size of feature maps in every stage. This is similar to SWinTrans-former [29], which uses a window size of 7. (2) In dense prediction tasks, we use larger to balance the complexity of region-level routing and token-level attention to achieve overall lower complexity. One can find hints from Eq. 9 of the paper, though we do not strictly follow the scaling rule due to the size divisor constraint. (3) We gradually increase to keep a reasonable number of tokens to attend as the region size becomes smaller in later stages.

在本文中， 和 的选择更多地考虑了工程问题。（1） 被选为训练大小的除数以避免填充，这会减慢训练速度并可能降低性能。例如，在图像分类中，分辨率是 ，我们使用 以确保它是每个阶段特征图大小的除数。这与 SWinTransformer [29] 类似，它使用 7 的窗口大小。（2）在密集预测任务中，我们使用较大的 来平衡区域级路由和标记级注意力的复杂性，以达到整体较低的复杂性。尽管由于大小除数约束，我们没有严格遵循论文中公式 9 的缩放规则，但可以从中获得提示。（3）随着后期阶段区域大小的减小，我们逐渐增加 以保持合理的关注标记数量。

It is possible to try different combinations of and . We show ablation results on IN-1K in Table 7, based on BiFormer-STL (as in the paper). A key observation from these experiments is that increasing the number of tokens to attend may even hurt the accuracy. This implies the explicit sparsity constraint may serve as a regularization to avoid distractions from the background.

可以尝试不同的 和 组合。我们在表格 7 中展示了基于 BiFormer-STL（如论文中所示）的 IN-1K 的消融实验结果。这些实验的一个关键观察是，增加关注标记的数量甚至可能会降低准确性。这表明显式的稀疏约束可能充当正则化，以避免来自背景的分心。

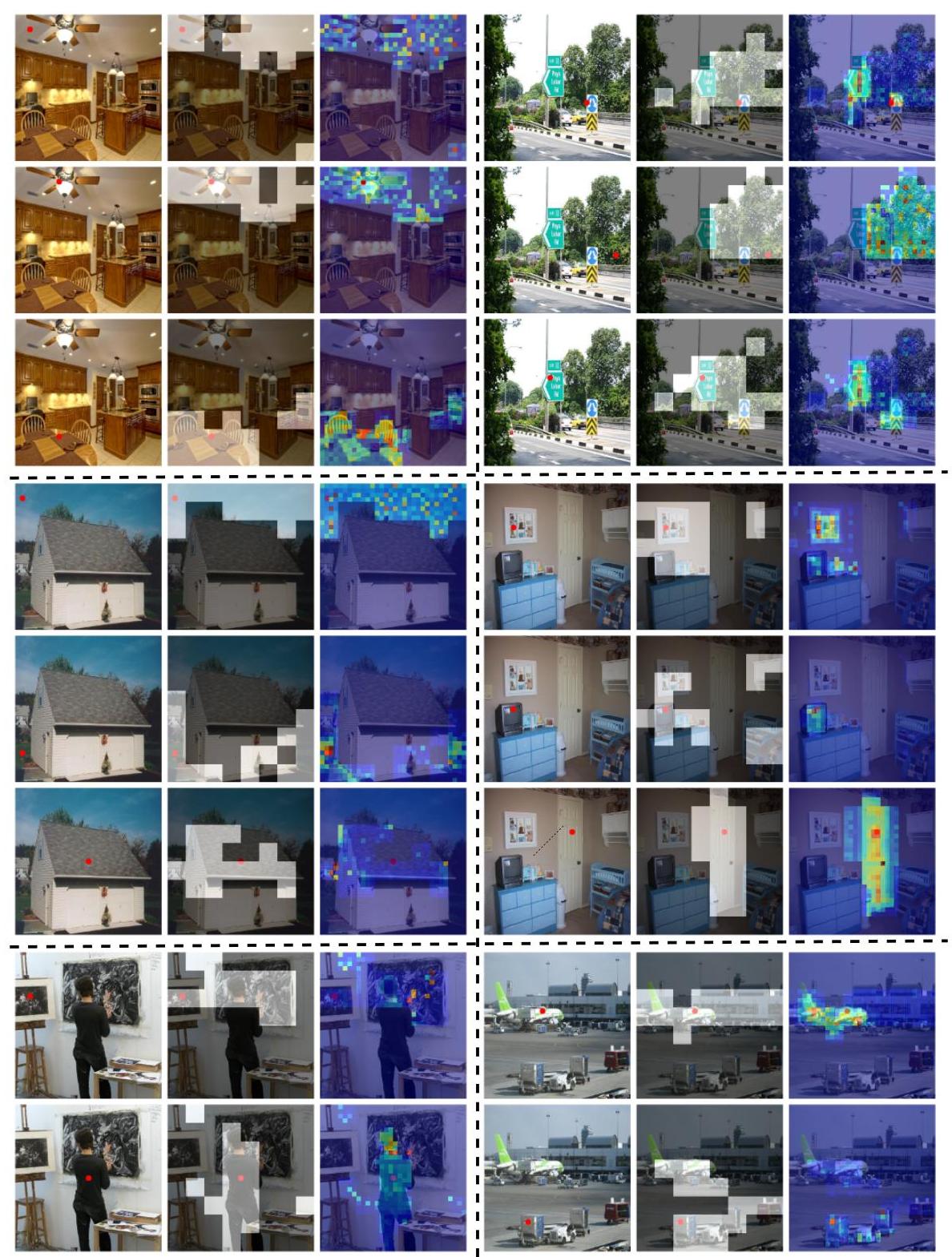


Figure 6. More attention map visualization results. For each scene, We demonstrate 2-3 query positions on the input image (left), corresponding routed regions (middle) and final attention heat map (right).

图 6。更多注意力图的可视化结果。对于每个场景，我们在输入图像（左）上展示了 2-3 个查询位置，相应的路由区域（中）和最终的注意力热图（右）。

# D. Adapting Pretrained Plain ViT with BRA

# D. 使用 BRA 适应预训练的普通 ViT

Recently, to take advantage of large-scale pretraining with masked image modeling, a new research direction emerges to adapt plain ViT [15] for dense prediction tasks . Here we explore adapting pre-trained plain ViT [15] for semantic segmentation with our proposed BRA.

最近，为了利用带有遮蔽图像建模的大规模预训练，一个新的研究方向出现了，即通过我们的提出的BRA来适应普通的ViT [15] 以进行密集预测任务 。在这里，我们探索将预训练的普通ViT [15] 适应于语义分割，使用我们提出的BRA。

Specifically, we replace all or part of full multi-head self-attention (MHSA) modules in DeiT-B [40] with our BRA and directly load the weights pre-trained on ImageNet before training on ADE20K dataset for semantic segmentation. In this way, the linear projection weights of BRA modules are initialized with those of the original MHSA. We compare such an adaptation with those proposed in [26], i.e. using local window attention (window size ) together with several global attention or convolution propagation blocks. We set window size (which is equivalent to region partition size since the feature map has a resolution of ) and the number of regions to attend , hence each query attends to key-value pairs, which is comparable to the local window attention where each query attends to key-value pairs.

具体来说，我们用BRA替换了DeiT-B [40] 中的全部或部分完整的多头自注意力（MHSA）模块，并直接加载在ImageNet上预训练的权重，然后在与ADE20K数据集上进行语义分割训练。这样，BRA模块的线性投影权重被初始化为原始MHSA的权重。我们将这种适应方法与[26]中提出的适应方法进行比较，即使用局部窗口注意力（窗口大小 ）并结合几个全局注意力或卷积传播块。我们设置窗口大小 （由于特征图的分辨率为 ，这与区域划分大小 等效）和关注区域数量 ，因此每个查询关注 个键值对，这与局部窗口注意力中每个查询关注 个键值对相当。

Table 8 shows the results. Without propagation blocks, the architecture using BRA significantly surpasses the one with local window attention by . When further equipped with 4 global propagation blocks, the performance of both architectures is improved, while the one using BRA still has an advantage of .

表8显示了结果。在没有传播块的情况下，使用BRA的架构显著超过了使用局部窗口注意力的架构 。当进一步配备4个全局传播块时，两种架构的性能都得到了提高，而使用BRA的架构仍然具有 的优势。

| attention function | mIoU(%) |
| --- | --- |
| local window attention | 43.55 |
|  | 45.92 |
| local window attention +4 conv prop. blks. | 44.68 |
| local window attention +4 global prop. blks | 46.64 |
| BRA + 4 global prop. blks. | 46.84 |

Table 8. Adapting pretrained ViT [15] with BRA for semantic segmentation on ADE20K. For decoder, we use the Simple Feature Pyramid [26] followed by with Upernet [49] head.

表8。使用BRA适应预训练的ViT [15] 以在ADE20K上进行语义分割。对于解码器，我们使用Simple Feature Pyramid [26] ，然后接上Upernet [49] 头。

# E. More Visualization Results

# E. 更多可视化结果

To further show how BRA works, we demonstrate more visualization results in Figure 6.

为了进一步展示BRA的工作原理，我们在图6中展示了更多的可视化结果。

# References

# 参考文献

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In European Conference on Computer Vision, pages 213-229, 2020. 1, 3

[2] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Region-vit: Regional-to-local attention for vision transformers. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. Open-Review.net, 2022. 5

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. arXiv:1906.07155, 2019. 6

[4] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong , Jifeng Dai, and Qiao. Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534, 2022. 11

[5] Zhiyang Chen, Yousong Zhu, Chaoyang Zhao, Guosheng , Wei Zeng, Jinqiao Wang, and Ming Tang. Dpt: Deformable patch-based transformer for visual recognition. In Proceedings of the 29th ACM International Conference on Multimedia, pages 2899-2907, 2021. 1, 3

[6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. arXiv:1904.10509, 2019. 1, 3

[7] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haib-ing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. Advances in Neural Information Processing Systems, 34:9355-9366, 2021. 5, 7, 8

[8] MMSegmentation Contributors. Openmmlab semantic segmentation toolbox and benchmark. https://github. com/open-mmlab/mmsegmentation, 2020. 7

[9] 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, pages 702-703, 2020. 6

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 764-773, 2017. 2

[11] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers, pages 2978-2988, 2019. 3

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248-255, 2009. 6

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171- 4186, June 2019. 1, 3

[14] 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, pages 12124-12134, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Syl-vain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth words: Transformers for image recognition at scale. In International Conference on Learning Representations, ICLR 2021, 2021, 2021. 1, 3, 11

[16] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noord-huis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv:1706.02677, 2017. 6

[17] Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient transformers via top-k attention. In Nafise Sadat Moosavi, Iryna Gurevych, Angela Fan, Thomas Wolf, Yufang Hou, Ana Marasovic, and Sujith Ravi, editors, Proceedings of the Workshop on Simple and Efficient Natural Language Processing, 2021, pages 39-52, 2021. 2

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Gir-shick. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 2961-2969, 2017. 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770-778, 2016. 5

[20] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. arXiv:1912.12180, 2019. 7

[21] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In European Conference on Computer Vision, pages 646-661, 2016. 6

[22] 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, pages 603-612, 2019. 5, 7

[23] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. Advances in Neural Information Processing Systems, 34:18590-18602, 2021. 2, 5, 6, 8

[24] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6399-6408, 2019. 7

[25] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. arXiv:2201.04676, 2022. 5, 6, 7, 8

[26] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. arXiv preprint arXiv:2203.16527, 2022. 11

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 2980-2988, 2017. 6

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740-755. Springer, 2014. 6

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012-10022, 2021. 1, 2, 3, 4, 5, 6,

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. 6

[31] Nvidia. How to access global memory efficiently in cuda c/c++ kernels. https://developer.nvidia. com / blog / how - access - global - memory - efficiently - cuda - c - kernels/. Accessed: 2022-10-25. 2

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of Advances in Neural Information Processing Systems, volume 32, 2019. 4

[33] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 1

[34] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10428- 10436, 2020. 5

[35] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In Proceedings of Advances in Neural Information Processing Systems, volume 32, 2019. 7

[36] Mr D Murahari Reddy, Mr Sk Masthan Basha, Mr M Chin-naiahgari Hari, and Penchalaiah. Dall-e: Creating images from text. 2021. 1

[37] 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, pages 10853- 10862, 2022. 5, 7, 8

[38] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In The International Conference on Learning Representations, ICLR 2022, 2022, 2022. 3, 5, 6, 7, 9

[39] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. ACM Computing Surveys (CSUR), 2020. 3

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In International Conference on Machine Learning, pages 10347-10357. PMLR, 2021. 2, 6, 11

[41] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In ECCV, 2022. 1, 2, 3, 4, 5, 7

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017. 1, 3, 4

[43] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. arXiv:2006.04768, 2020. 3

[44] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 568-578, 2021. 1, 7

[45] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. Computational Visual Media, 8(3):415-424, 2022. 5, 8

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

[47] Ross Wightman. Pytorch image models. https : //github.com/rwightman/pytorch-image-models, 2019. 9

[48] 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, pages 4794-4803, 2022. 1, 2,

[49] 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), pages 418-434, 2018. 7, 11

[50] Rui Yang, Hailong Ma, Jie Wu, Yansong Tang, Xuefeng Xiao, Min Zheng, and Xiu Li. Scalablevit: Rethinking the context-oriented generalization of vision transformer. arXiv:2203.10790, 2022. 5

[51] Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-vit: Unifying wavelet and transformers for visual representation learning. In European Conference on Computer Vision, pages 328-345. Springer, 2022. 5, 6, 7, 8

[52] 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, pages 6023-6032, 2019. 6

[53] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are

equal: Human-centric visual analysis via token clustering transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11101- 11111, 2022. 3

[54] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In International Conference on Learning Representations, ICLR 2018, 2018. 6

[55] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fi-dler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision, 127(3):302-321, 2019. 6, 7