Table 1: Detailed configurations of architecture variants of RTFormer.

Models	#Channels	#Blocks	Spatial size of cross-feature
RTFormer-Slim	$[32, 64, 64/128, 64/256, 64/256]$	$[2, 2, 1/2, 1, 1]$	8×8
RTFormer-Base	$[64, 128, 128/256, 128/512, 128/512]$	$[2, 2, 1/2, 1, 1]$	12×12

A Architecture of RTFormer

In this section, we describe the detailed configurations of RTFormer-Slim and RTFormer-Base, which are recorded in Table 1. For the number of channels and number of blocks, each array contains 5 elements, which are corresponding to the 5 stages respectively. Especially, the elements with two numbers are corresponding to the dual-resolution stages. For instance, $64/128$ means the number of channels is 64 for high-resolution branch and 128 for low-resolution branch. While $1/2$ means the number of basic convolution blocks is 1 for high-resolution branch and 2 for low-resolution branch. It is worth to be noted that, the last two elements in block number array denote the number of RTFormer blocks, and they are both 1 for RTFormer-Slim and RTFormer-Base. The spatial sizes of cross-feature are set as $64(8 \times 8)$ and $144(12 \times 12)$ for RTFormer-Slim and RTFormer-Base respectively.

Table 2: Training settings on ImageNet classification.

config	value
optimizer	AdamW
base learning rate	0.0005
weight decay	0.04
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	1024
learning rate schedule	cosine decay
minimum learning rate	5e-6
warmup epochs	5
warmup learning rate	5e-7
training epochs	300
augmentation	RandAug(9, 0.5)
color jitter	0.4
mixup	0.2
cutmix	1.0
random erasing	0.25
drop path	0.0

B ImageNet Pre-training

RTFormer is consist of several convolution blocks and RTFormer blocks, and RTFormer block contains different types of attention. Thus, we pre-train RTFormer on ImageNet-1K[2] mainly following the settings of training transformer network[9], and the detail configuration is provided in Table 2.

Table 3 shows the performance of RTFormer on ImageNet classification. Both RTFormer-Slim and RTFormer-Base outperform the corresponding DDRNet variants. In addition, RTFormer-Base achieves the best performance among the existing backbones adopted in real-time semantic segmentation task.

C More Experiments

C.1 Comparison with state-of-the-arts on COCOStuff

COCOStuff. COCOStuff[1] is a dense annotated dataset derived from COCO. It contains 10K images (9K for training and 1K for testing) with respect to 182 categories, including 91 thing and 91 stuff classes. And 11 of the thing classes have no annotations. We train RTFormer 110 epochs on COCOStuff with AdamW optimizer, and the initial learning rate and weight decay are set as 0.0001 and 0.05 respectively. In the training phase, we first resize the short side of image to 640 and randomly crop 640×640 patch for augmentation. While in the testing phase, we resize all images into 640×640. Other training settings are identical to Cityscapes.

Results. As shown in Table 4, our RTFormer-Base achieves 35.3 mIoU at 143.3 FPS, which outperforms the DDRNet-23 about 3% with a comparable inference speed and set a new state-of-the-art. In addition, when we set the spatial size of cross-feature in RTFormer-Base as 8×8, the performance drops to 34.6 while the inference speed only increases a little. This indicates that $144 = 12 \times 12$, which is close to the feature dimension in high-resolution branch(128 for RTFormer-Base), is a relative good setting for the spatial size of cross-feature in RTFormer-Base. And from this comparison, we can find the cross-resolution attention can also works well on COCOStuff.

Table 3: Classification accuracy on the ImageNet validation set. Performances are measured with a single 224×224 crop. "#Params" refers to the number of parameters. "FLOPs" is calculated under the input scale of 224×224.

Method	#Params↓	FLOPs↓	Top-1 Acc. ↑
ResNet-18[5]	11.2M	1.8G	69.0
RestNet-50[5]	23.5M	3.7G	75.3
DF1[8]	8.0M	0.7G	69.8
DF2[8]	17.5M	1.7G	73.9
MobileNetV2[10]	3.4M	0.3G	72.0
MobileNetV3[7]	5.4M	0.2G	75.2
Efficient-Net-B0[11]	5.3M	0.4G	76.3
STDC1[3]	8.4M	0.8G	73.9
STDC2[3]	12.5M	1.4G	76.4
DDRNet-23-slim[6]	7.6M	1.0G	70.2
DDRNet-23[6]	28.2M	3.9G	75.9
RTFormer-Slim	5.3M	0.8G	72.3
RTFormer-Base	20.5M	3.0G	77.4

Table 4: Comparisons with state-of-the-arts on COCOStuff. The #Params, FLOPs and FPS are measured at resolution 640×640. RTFormer-Base-8 means the spatial size of cross-feature in RTFormer-Base is set as 8×8.

Method	GPU	#Params↓	FLOPs↓	FPS↑	test mIoU(%)↑
PSPNet50[15]	-	-	-	6.6	32.6
ICNet[14]	TitanX M	-	-	35.7	29.1
BiSeNetV2[13]	GTX 1080Ti	-	-	87.9	25.2
BiSeNetV2-L[13]	GTX 1080Ti	-	-	42.5	28.7
DDRNet-23	RTX 2080Ti	20.1M	28.1G	146.1	32.1
RTFormer-Base-8	RTX 2080Ti	16.8M	26.6G	146.6	34.6
RTFormer-Base	RTX 2080Ti	16.8M	26.6G	143.3	35.3

C.2 More ablation studies on different types of attention

In this section, we extend the ablation study about different types of attention by adding multi-head self-attention. Meanwhile, we supplement experimental details and analysis about different types of attention.

Experimental Details. We introduce a type of multi-head self-attention which is improved by[12] for comparison. In contrast to the traditional self-attention, this type of self-attention shrinks the spatial size of key and value as $\frac{1}{\sigma}$ of the input feature, which can reduce the computation cost caused by the large input resolution. Concretely, in this ablation study, multi-head self-attention is also used for replacing the attention operations in all RTFormer blocks. And we set $\sigma = 4$ for the self-attention in high-resolution branch, while $\sigma = 1$ for low-resolution branch, following the settings for feature maps with stride=8 and stride=32 in[12].

For both multi-head self-attention and multi-head external attention, which are denoted as SA and EA in Table 5, we set the number of heads as 2 and 8 for high-resolution and low-resolution branches respectively. Similarly, for the GPU-Friendly attention, we set the number of groups as 2 and 8 separately for high-resolution and low-resolution branches. For the case of GFA+CA, the number of groups of the GPU-Friendly attention in low-resolution is still set as 8, while the cross-resolution attention has no multi-head calculation.

Especially, we give three results of multi-head external attention with r=$[0.125, 0.25, 1]$. And when r=0.25, the parameter dimension of multi-head external attention M in low-resolution branch is 64, which is identical to the setting in[4]. And the other two results are used for showing more

Table 5: Comparison among different types of attention on ADE20K. SA, EA, GFA, CA denote multi-head self-attention, multi-head external attention, GPU-Friendly attention and cross-resolution attention respectively. For example, GFA+CA means adopting GFA in low-resolution branch and CA in high-resolution branch. r is a ratio for adjusting the parameter dimension M in multi-head external attention.

Attention	GPU	FPS↑	val mIoU(%)↑
SA+SA	RTX 2080Ti	97.4	32.7
EA+EA (r=0.125)	RTX 2080Ti	196.9	31.9
EA+EA (r=0.25)	RTX 2080Ti	189.6	32.0
EA+EA (r=1)	RTX 2080Ti	180.8	32.2
GFA+GFA	RTX 2080Ti	189.8	32.8
GFA+CA	RTX 2080Ti	187.9	33.0

variations of the trade-off between performance and inference speed. For GPU-Friendly attention, we set $M_g = d$ constantly.

Analysis. As illustrated in Table 5, we can find that multi-head self-attention achieves 32.7 mIoU, which performs better than multi-head external attentions with different settings of r. But, the inference speed of multi-head self-attention is not competitive, which is mainly caused by the quadratic complexity and multi-head mechanism. Multi-head external attention can achieve a good inference speed, which is benefit from its linear complexity and the design of sharing external parameter for multiple heads. Associated with the above two properties, multi-head external attention adopts a low parameter dimension $M(\ll d)$, which reduces the total computation cost further. However, the performance of multi-head external attention is suboptimal, as the network capacity is limited by those designs. Yet, the multi-head mechanism still remains, which is not friendly for running on GPU-like devices and leads to a relative worse efficiency than single head situation. As a example, when we let M to be equal to d, the performance is still worse than multi-head self-attention, and the inference speed drops about 10FPS than $M = 0.25d$. While, GPU-Friendly attention, which is derived from multi-head external attention, can achieve both relative good performance and inference speed. It is because that, GPU-Friendly attention discards the multi-head mechanism and the grouped double normalization makes the matrix multiplication to be integrated and friendly for GPU calculation. Therefore, the external parameters can be enlarged for increasing the network capacity without great loss of inference speed. Finally, the combination of GPU-Friendly attention and cross-resolution attention improves the performance further, and it outperforms multi-head self-attention in both accuracy and efficiency, which validates the effectiveness of our proposed attentions.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[2] 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. Ieee, 2009.

[3] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9716–9725, 2021.

[4] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv preprint arXiv:2105.02358*, 2021.

[5] 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.

[6] Yuanduo Hong, Huihui Pan, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint arXiv:2101.06085*, 2021.

[7] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[8] Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9145–9153, 2019.

[9] 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.

[10] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

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

[12] 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, 2021.

[13] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021.

[14] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[15] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.