A. More Experimental Details

A.1. Datasets

Cityscapes. Cityscapes [9] is extensively utilized in urban scene understanding and autonomous driving research. It consists of 19 classes and a total of 5000 images, with 2975 designated for training, 500 for validation, and 1525 for testing. The test set is unlabeled, and model predictions must be uploaded to a specific website for evaluation. Each image has been meticulously annotated and possesses a resolution of 1024 × 2048 pixels.

CamVid. CamVid [1] is the first video dataset to include semantic labels. It comprises 701 images, with 367 designated for training, 101 for validation, and 233 for testing. Each image possesses a resolution of 720 × 960 pixels and features 32 class labels, though only 11 are typically used for training and evaluation. Notably, many current research [21, 32, 35] utilize the training and validation sets for training and the test set for evaluation, and we adopt this same strategy.

Pascal VOC 2012. Pascal VOC 2012 [14] is primarily utilized for image classification, object detection, and image segmentation tasks. It includes 20 classes along with 1 background class. For the semantic segmentation task, the dataset consists of 2913 images, with 1464 allocated to the training set and 1449 to the validation set. Unlike Cityscapes and CamVid, the resolution of these images varies.

A.2. Implementation Details

Computing Platform. The hardware of the computing platform comprises an AMD EPYC 7742 CPU and four NVIDIA A100 GPUs. The software stack includes Ubuntu 20.04.6, CUDA 11.3, PyTorch 1.12.1, TorchVision 0.13.1, MMEngine 0.10.2, and MMSegmentation 1.2.2 [8]. During the training phase, both GPUs are utilized, while for the inference phase, only a single GPU is used, with the batch size set to 1.

Training. Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay of 0.0005 was employed as the optimizer. Additionally, a polynomial learning rate decay strategy with a power of 0.9 was utilized. For data augmentation, we applied random scaling within the range of 0.5 to 2.0, random cropping, and random flipping with a probability of 0.5. The training iterations, initial learning rate, random crop size, and batch size for the Cityscapes, CamVid, and Pascal VOC 2012 datasets were configured as follows: [120000, 0.01, 1024 × 1024, 12], [7800, 0.001, 720 × 960, 12], and [24400, 0.001, 512 × 512, 16], respectively. Notably, our model was not pretrained on the ImageNet [10] dataset, and the Cityscapes model weights were used during training on the CamVid and Pascal VOC 2012 datasets.

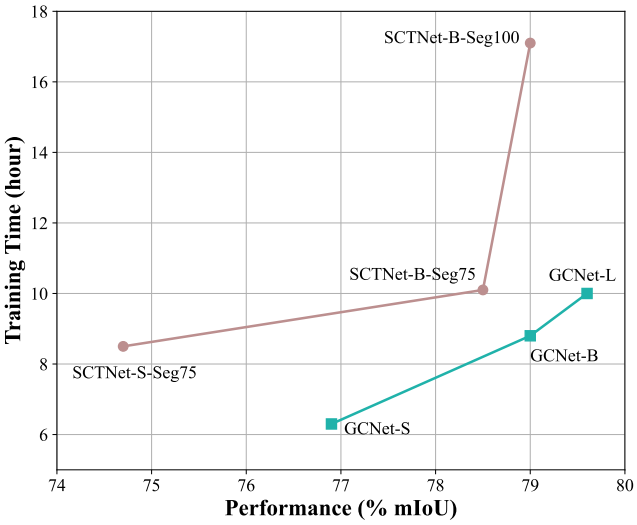

Figure 7. Comparison of training time between GCNet and SCTNet on the Cityscapes dataset. Four A100s were used for training and the training time was recorded.

Inference. The inference image sizes for the Cityscapes, CamVid, and Pascal VOC 2012 datasets were configured at 1024 × 2048, 720 × 960, and 512 × 2048, respectively. Since the image resolution in the Pascal VOC dataset is not fixed, images are adjusted to an approximate resolution of 512 × 2048 to maintain the aspect ratio. The inference speed of the same model varies across different CPUs and software environments, even when using the same GPU. To ensure a fair comparison, we made every effort to reimplement the other models.

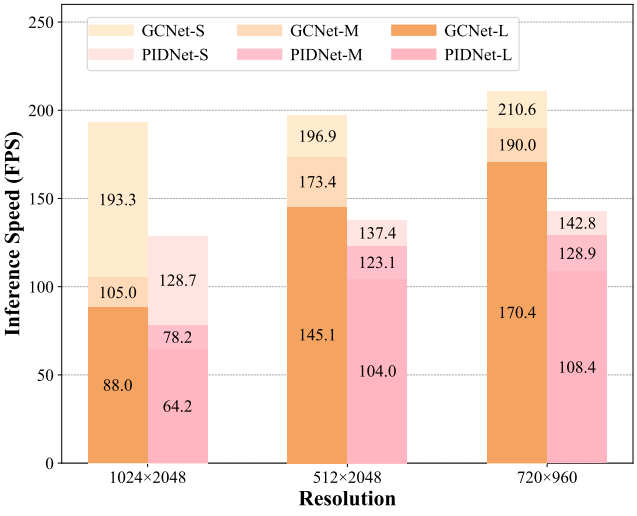

Figure 8. Inference speed of different segmentation models for varying resolutions. The GPU used is A100.

Table 5. Comparison of inference speed on varying GPUs using the Cityscape validation set. "*" indicates that the model was reimplemented and retested by us (with convolution and batch normalization fused). "ImageNet" indicates whether the model utilizes ImageNet pretrained weight.

Method	Resolution	RTX 4080	RTX 3090	V100	A100	ImageNet	mIoU
ICNet* [38]	1024 × 2048	92.3	108.7	76.7	181.3	✗	71.6
Fast-SCNN* [23]	1024 × 2048	125.3	211.9	173.6	325.9	✗	71.0
CGNet* [30]	1024 × 2048	38.3	53.1	48.5	73.9	✗	68.3
BiSeNetV1* [34]	1024 × 2048	62.7	64.3	56.5	116.8	✓	74.4
BiSeNetV2* [35]	1024 × 2048	68.7	69.6	64.4	132.4	✗	73.6
STDC1* [15]	1024 × 2048	101.1	103.3	89.4	183.7	✗	71.8
STDC2* [15]	1024 × 2048	75.8	73.8	64.7	132.8	✗	74.9
DDRNet-23-Slim* [21]	1024 × 2048	94.2	116.5	97.6	166.4	✗	76.3
DDRNet-23* [21]	1024 × 2048	54.2	53.4	47.1	106.0	✗	78.0
PIDNet-S* [32]	1024 × 2048	60.1	84.2	76.3	128.7	✗	76.4
PIDNet-M* [32]	1024 × 2048	41.7	41.3	35.4	78.2	✗	78.2
PIDNet-L* [32]	1024 × 2048	30.0	31.1	27.9	64.2	✗	78.8
SCTNet-S-Seg50* [33]	512 × 1024	78.9	124.2	108.3	169.1	✗	71.0
SCTNet-S-Seg75* [33]	768 × 1536	78.7	124.0	99.7	168.7	✗	74.7
SCTNet-B-Seg50* [33]	512 × 1024	77.3	119.0	97.6	162.6	✗	75.0
SCTNet-B-Seg75* [33]	768 × 1536	73.9	101.9	83.1	157.3	✗	78.5
SCTNet-B-Seg100* [33]	1024 × 2048	63.6	63.9	53.3	117.0	✗	79.0
GCNet-S	1024 × 2048	110.1	130.9	114.1	193.3	✗	77.3
GCNet-M	1024 × 2048	47.4	50.1	44.2	105.0	✗	79.0
GCNet-L	1024 × 2048	38.3	40.7	36.2	88.0	✗	79.6

A.3. Metrics

We adopt mIoU (mean Intersection over Union), number of parameters, GFLOPs (Giga Floating Point Operations), and FPS (Frames Per Second) as metrics. mIoU is commonly used in image segmentation tasks, where it measures the average overlap between predicted results and ground truth annotations, serving as an indicator of model performance. Number of parameters refers to the total number of trainable parameters in the model, providing a measure of its size. GFLOPs quantifies the computational load of the model, reflecting its computational complexity. FPS represents the number of image frames the model processes per second, serving as a metric for inference speed. In general, number of parameters and GFLOPs are not the decisive factors influencing FPS. This paper focuses on optimizing mIoU and FPS, rather than number of parameters and GFLOPs.

B. More Experiments

B.1. Training Time of GCNet and SCTNet

The paper mentions that SCTNet requires a high-performance segmentation model for knowledge distillation training, which is quite time-consuming. To verify that GCNet demands less training time compared to SCTNet, we recorded the training times of various versions of both models, as shown in Figure 7. Since SCTNet-S/B-Seg50 and SCTNet-S/B-Seg75 share the same training configuration, differing only in inference settings, we only recorded the

training time for SCTNet-S-Seg75 and SCTNet-B-Seg75. The figure reveals that GCNet not only requires less training time but also achieves higher performance than SCTNet. Specifically, SCTNet-B-Seg100 takes 17.1 hours to reach 79.0% mIoU, while GCNet-B achieves this in only 8.8 hours.

B.2. Inference Speed With Varying Resolution

To provide an intuitive understanding of the inference speed of GCNet at varying resolutions, we visualized its FPS, as shown in Figure 8. The figure reveals that as resolution decreases, the FPS of both GCNet and PIDNet increases significantly, particularly for the M and L versions. Surprisingly, GCNet-M and GCNet-L achieve outstanding FPS at lower resolutions (512×2048 and 720×960), with GCNet-L even surpassing PIDNet-S. This may be attributed to the benefits of reduced memory access enabled by the single-path block, as memory access represents a significant computational cost in computer systems.

B.3. Inference Speed With Varying GPUs

To demonstrate GCNet's versatility across varying GPUs, we conducted speed tests on the RTX 4080, RTX 3090, V100, and A100, as shown in Table 5. The results reveal that GCNet performs well on both consumer-grade GPUs (RTX 4080 and RTX 3090) and professional-grade GPUs (V100 and A100). Interestingly, smaller models show faster inference speeds on the RTX 3090 compared to the RTX 4080, while larger models perform similarly on both. As a single-branch architecture model, SCTNet is relatively insensitive to changes in lower resolutions, with comparable inference speeds for Seg50 and Seg75. In contrast, multi-branch models show substantial speed improvements with lower resolutions. As illustrated in Figure 8, inference speeds for multi-branch models, especially GCNet, increase significantly as the resolution decreases. We attribute this to the high-resolution branch in multi-branch architectures, which requires the maintenance of larger feature maps and thus more computations. In future work, we plan to further investigate GCNet on lower resolutions using the Cityscapes dataset.