Framework-agnostic Semantically-aware Global Reasoning for Segmentation

Mir Rayat Imtiaz Hossain[1,2] Leonid Sigal[1,2,3] James J. Little[1]
[1]University of British Columbia [2]Vector Institute for AI [3]Canada CIFAR AI Chair
{rayat137, lsigal, little}@cs.ubc.ca

Figure 1. **Ground truth segmentation mask and corresponding connected components.** The connected components are assumed to give a lower bound on the number of instances; *e.g.*, the illustrated image contains 6 people but the number of connected components that corresponds to person is only 3.

1. Implementation Details

1.1. Semantic Segmentation

As mentioned in our main paper, for experiments involving semantic segmentation, we add our SGR component at the end of the final layer of backbones, which are pre-trained on ImageNet, before passing the processed features with added global context to appropriate segmentation heads. For the Dilated FCN [2] and DeeplabV3 [2] heads, we use a multi-grid approach with dilated convolutions for Resnet backbones during training. The last two downsampling layers are removed, resulting in an output stride of 8. For the Swin-T [9] backbone with UperNet [11], we add the SGR component after the final Swin-T layer.

The models using Dilated-FCN and DeepLabV3 as segmentation heads are trained using the SGD optimizer with a momentum [10] of 0.9 and a weight decay of 0.0001. We train on the Cityscapes dataset with an initial learning rate of 0.006 and the **ADE-20K** [14] and **Coco-Stuffs-10K** [1] datasets with an initial learning rate of 0.004.

For the Maskformer [5] and Mask2Former [4] heads, we do not use multi-grid or dilated convolution (as mentioned in the paper); hence the output features have a resolution which is 32 times smaller than the input features. The outputs of contextualized features along with rest of the layers are passed to UperNet [11] head for segmentation. For both Maskformer [5] and UperNet [11], we used the AdamW optimizer. For Maskformer [5] and Mask2former [4] models we use the optimizers, learning rate and weight decay hyper-parameters as mentioned in the respective papers.

For all experiments, during training, we applied random horizontal flips, random scaling between [0.5-2.0] and random color jitter following [5, 12] for data augmentation. For **Cityscapes** [6], following the random data augmentation, the images are cropped from the center with a crop size of 768×768. For both **ADE-20K** [14] and **Coco-Stuffs-10K** [1] a center crop of crop size 512×512 is used following the abovementioned random image transformations during training. We train models on Cityscapes using a batch size of 8 and on the other two datasets using a batch size of 16. When trained across multiple GPUs, we apply synchronized batchnorm [13] to synchronize batch statistics following existing work [2, 3, 5, 7, 12]. We train on Cityscapes, COCO-Stuffs-10K, and ADE-20K for 240 epochs, 140 epochs and 120 epochs respectively. For all experiments, we used a polynomial learning rate policy where the learning rate decreases with the formula $(1 - \frac{iter}{total\_iter})^{0.9}$ with every iteration.

For models with Resnet backbones the initial base learning rate is multiplied by a factor of 10.0 for the parameters of the SGR component and the layers that correspond to the segmentation head. For the Swin-T backbone we use the same learning rate for both the backbone and segmentation head.

For all three datasets, we report both the single scale inference and multi-scale inference with horizontal flip at scales 0.5, 0.75, 1.0, 1.25, 1.5 and 1.75 following existing work [2, 5, 7, 12]. During multi-scale inference, the final output is calculated by taking the mean probabilities over each scale and their corresponding flipped inputs. Follow-

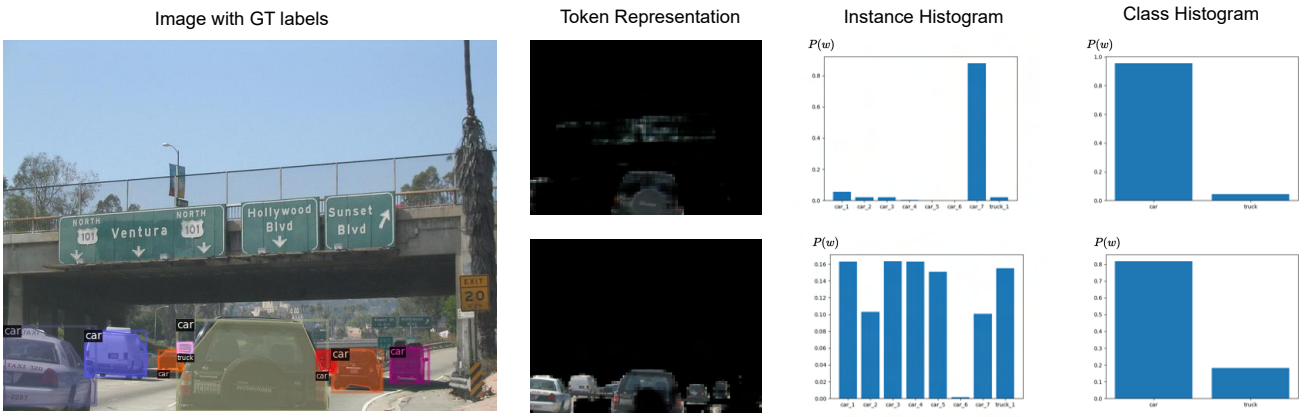

Figure 2. **Visualization of instance- and class-level histograms.** (Left) Image with ground truth instances of "things" classes. (2nd Column) Two different concept regions aggregating information. (3rd Column) Instance-level histograms. (Right) Class-level histograms. $P(w)$ indicates the probability that weight of a concept region belongs to a particular concept. While the bottom token captures a good class-level semantics, on instance-level the semantics are poor.

ing [5, 7, 12], for ADE-20K and COCO-Stuffs, we resize the shorter side of the image to the crop size followed by a center crop to ensure that all images are of the same size.

Hyper-parameters for training. For Hungarian matching, in training, we used $\rho = 1.0$ for dice loss (see Eq. (5) in the main paper). For matching, as mentioned in the paper, the value of L is set to 64. Hence, the top $L = 64$ tokens are matched using the greedy matching approach based on the cost matrix (Figure 3 of the main paper). Once matched, we used a weight of 0.25 for the hyperparameter β that controls the importance of binary mask losses with respect to cross-entropy loss or mask classification loss (depending on the framework we use) to train the models (see Eq. (7)).

1.2. Transfer to Downstream Tasks

For transfer to the downstream tasks, we removed the segmentation head from our semantic segmentation network trained on COCO-Stuffs-10K and use it as a backbone for Mask-RCNN [8] to fine-tune on the MS-COCO `train2017` subset, which has 118K images, for object detection and instance segmentation. The same approach was adopted while transferring the GloRE [3] based backbone pretrained for segmentation on COCO-Stuffs-10K. For the Res101-C4 backbone, however, we used the weights pretrained for classification on Imagenet. We reported our results on the `val2017` subset having 5K images. The authors of Mask-RCNN used a batch size of 16 and trained on the `trainval-135K` subset and reported results on the `minival` dataset which is the same as `val2017`. Therefore, for a fair comparison with other backbones, we trained them from scratch on MS-COCO `train2017` using the same batch size, learning rate and iterations. We used a batch size of 8, an initial learning rate of 0.02, and used SGD with a momentum of 0.9 and weight decay of 0.0001 to train the models. We trained for 270K iterations with a learning rate decreased by 0.1 at 210K and 250K iterations. Following Mask-RCNN [8], the RPN anchors span 5 scales and 3 aspect ratios. For all the reported backbones, 512 ROIs are sampled with a positive to negative ratio 1:3.

2. Ground Truth connected components

Figure 1 shows the result of applying connected component analysis on ground truth semantic segmentation masks. As can be seen in the figure, the class `person` is divided into three different components. There are altogether 6 people in total. Hence, we observe that generally connected components form a lower bound on the number of instances. Similarly, the "stuffs" class `ground` is divided into two different components and the class `banana` has only one component. For "stuffs" classes the notion of instances is not well defined, but connected components serve as a good proxy for disjoint regions that are often semantically meaningful within the scene.

3. Visualization of histograms for tokens

Figure 2 shows the visualization of class-level and instance-level histograms for two different tokens, which we use to compute class- and instance-level semantics metric (defined in the main text). The lower the entropy of each of these histograms, the more semantically meaningful the tokens are at class or instance level of granularity. As can be observed in Figure 2, the first token has high instance and class level semantics since it mostly aggregates information from a single car, in this case, car_7. The lower token, despite being highly semantic at class-level (having lower entropy at class-level), is poor at capturing instance-level semantics. Hence, a token which is semantic at an

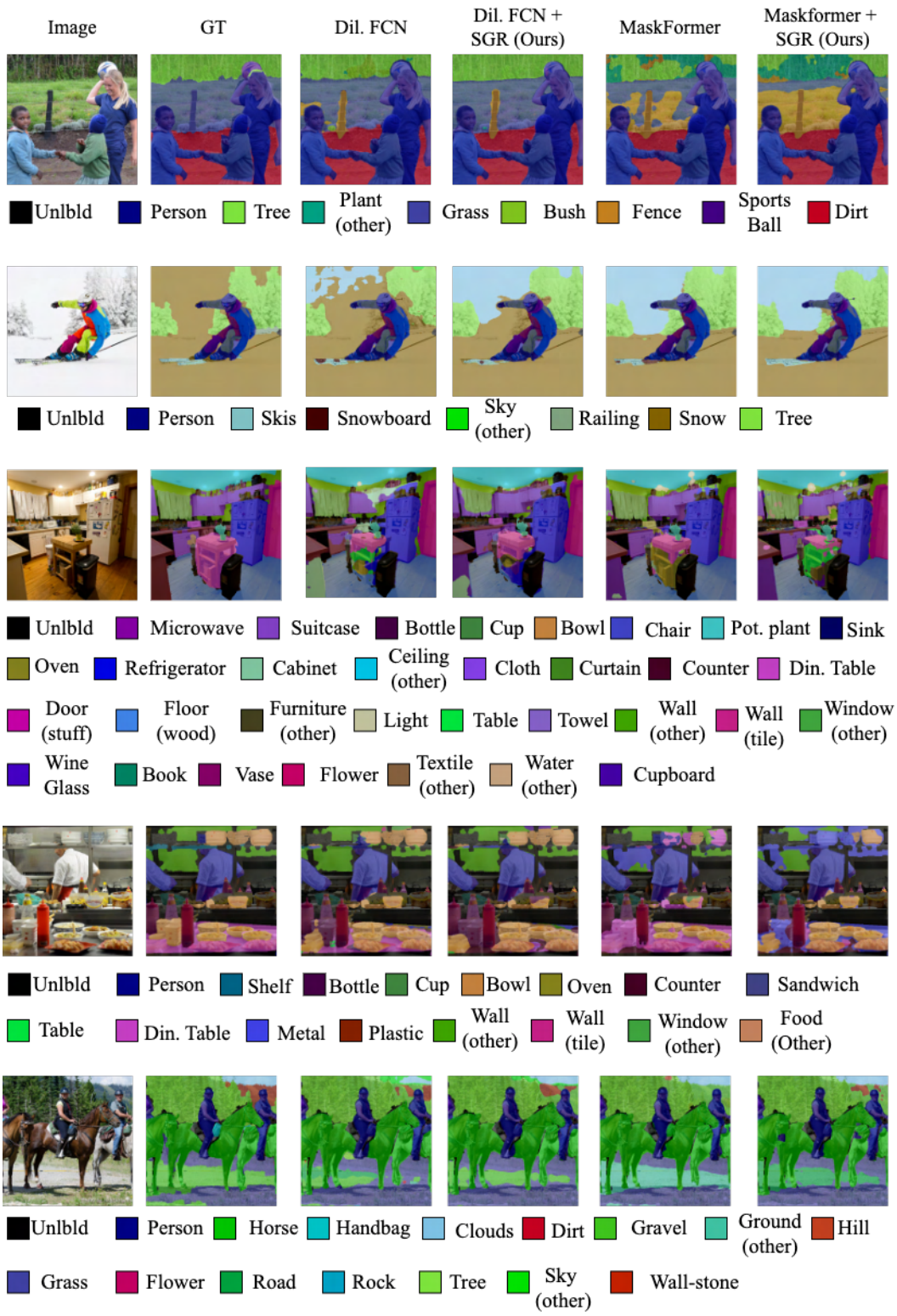

Figure 3. **Qualitative results on COCO-Stuffs-10K**. The leftmost two columns correspond to the image and ground truth semantic segmentation; the third column shows the predictions of the Dilated-FCN head; the fourth column shows the predictions of our SGR component added with Dilated-FCN [2]; the fifth column shows predictions from Maskformer [5] and the last column shows the predictions of our model added on top of Maskformer. The colors representing the class are also shown below the images.

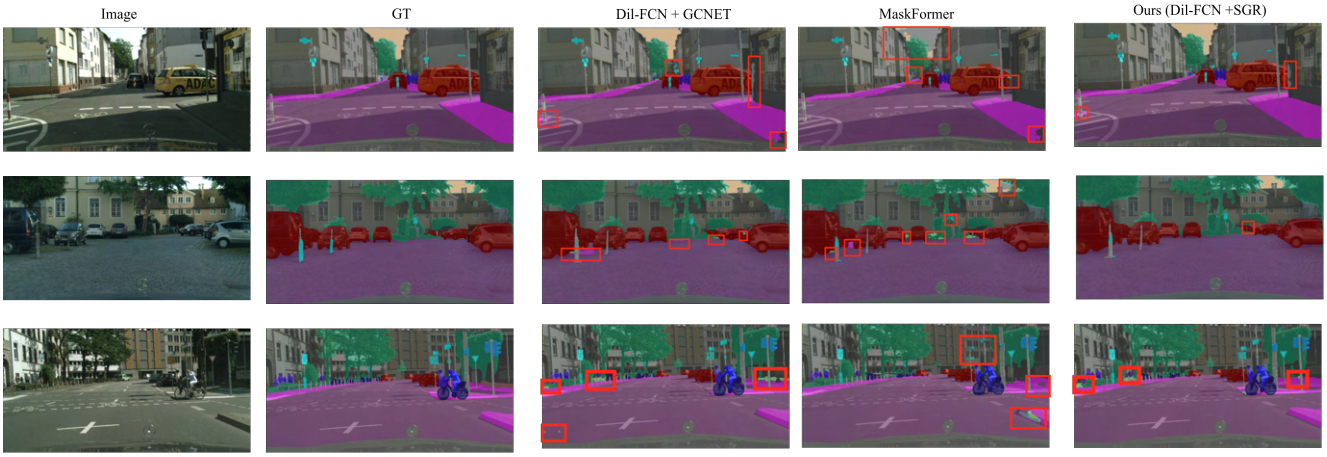

Figure 4. **Qualitative results on Cityscapes**. The leftmost two columns correspond to the image and ground truth semantic segmentation; the third column shows the predictions of Dilated-FCN+GCNET [3]; the four column shows predictions from Maskformer [5] and the final column shows predictions of our SGR component on top of Dilated-FCN. Red rectangles on the images indicate locations where the models failed to correctly segment the pixels.

instance-level is also highly semantic at class-level but not the other way around.

Method		FLOPs	mIOU(m.s)
Dil-FCN [14]	w/o SGR	224G	38.9
	w SGR	236G (**+12G**)	39.7 (**+0.8**)
MaskFormer [19]	w/o SGR	75G	39.3
	w SGR	80G (**+5G**)	39.9 (**+0.6**)

Table 1. **Computation Overhead for adding SGR.** Flops computed for Image size of 512 $\times$ 512.

K	L	mIOU (m.s.)	S_c	D_c	S_I	D_I
512	64	39.7	0.226	0.389	**0.315**	**0.316**
512	32	39.6	0.242	0.364	0.344	0.284
256	64	39.5	**0.222**	**0.399**	0.317	0.314
256	32	39.7	0.236	0.376	0.329	0.297
128	64	**39.8**	0.231	0.383	0.325	0.302

Table 2. **Ablation for different values of K and L**. All experiments on Dil-FCN+SGR with R101 backbone

4. Computation Overhead and Hyper-parameter sensitivity

Table 1 below shows the computational overhead for our component. As can be seen, adding the SGR component consistently gives a performance boost at minimal computational burden regardless the type of framework.

We have performed an ablation for different values of K (number of tokens) and L (number of tokens matched) in Table 2 to analyze the sensitivity of our component to those hyper-parameters. As we observe, the performance of our component is not sensitive to exact values of K and L.

5. Qualitative Results

5.1. Semantic Segmentation

Qualitative Results on COCO-Stuffs-10K Figure 3 shows the qualitative result of semantic segmentation of on COCO-Stuffs-10K. In the first image, we observe that adding our component over Dil-FCN improves overall segmentation quality. When compared to Maskformer, our model generally misclassifies trees for plant-other, however it produces a consistent mask for fence. In fact, the ground truth is noisy in this case because there is clearly a barbed-wire fence in the image. Maskformer was able to capture the fence to a degree but produced an inconsistent map. For the second image, all methods misclassified the upper portion of the image as sky instead of snow. Compared to Dilated FCN, our component has much higher intersection over tree, person and ski classes. Maskformer makes consistent predictions however it has misclassified ski as snow-board at multiple locations. In the third image, adding the SGR component over Dilated FCN clearly produces more accurate segmentation. Maskformer misclassifies microwave and the walls and textile, which adding the SGR component improves. You can also observe that adding the SGR component also produces more consistent masks compared to Maskformer. For the fourth image, we can observe a general improvement over dilated FCN. Both Maskformer and our SGR + Maskformer perform poorly on this particular image. In the final image, Dilated FCN misclassified certain poriton of the gravel as ground-other and has generally poor intersection elsewhere (particularly for sky and hill) compared to SGR + Dilated FCN. Maskformer has incorrectly classified the gravel as ground-other and cannot segment the hill class properly. Adding SGR

Figure 5. **Qualitative results for downstream tasks of object detection and instance segmentation on MS-COCO**. The leftmost two columns correspond to the image and ground truth object locations and their corresponding segmentation; the center column shows the predictions of using our backbone; the rightmost two columns show predictions from using Res101-C4 and Res101-GloRE [3] backbones.

over Maskformer leads to better segmentation of hill although there is still misclassification of gravel as road.

Qualitative Results on Cityscapes. Figure 4 shows the qualitative results of semantic segmentation on Cityscapes. The red rectangles indicate locations where each of the models made incorrect predictions. In the first image, the dilated FCN misclassified the road sign and side of the pavement (rightmost corner). Both of our models (Dil-FCN +SGR and DeeplabV3+SGR) alleviate those mistakes – DeeplabV3+SGR is more accurate. Maskformer has made incorrect classification of sky, car and pavement. We can see similar trends in the rest of the two images where our models produce more consistent and accurate predictions.

5.2. Object Detection and Instance Segmentation

Figure 3 shows the qualitative result of object detection and instance segmentation of using our pre-trained backbone in Mask-RCNN [8] on MS-COCO, compared to pre-trained Res101-C4 and Res101-GloRE [3] backbones. In the first image, the other two backbones mis-classified the leftmost `elephant` as a `person` which we correctly identify and segment. Moreover, they missed the rightmost instance of the `elephant` which model using our backbone was able to detect. Overall segmentation quality of each elephant was also better for our backbone. In the second image, other backbones erroneously classified the `lamp`

on the left as parking meter. This is likely due to the lack of global reasoning needed to make a distinction between these two objects within the context of the scene that our backbone contains. Both of them also missed the backpack of the person on the right. The model using our backbone consistently identifies objects and segments them better.

In the third image, our backbone segments the couches better than the other backbones. In the fourth image, the instance segmentation of the person is better than the other two backbones. Moreover, the Res101-C4 backbone has missed the handbag altogether, while the Res-101-GloRE backbone cannot segment the handbag properly. In the final image, the Res-101-C4 backbone incorrectly labelled US flag as a chair. Besides, the instance segmentation quality is lower than our backbone. The Res-101-GloRE failed to identify the truck completely, identifying part of of it as motorcycle and inaccurately segmented it. The general quality of object segmentation is also worse. All these qualitative results demonstrate the fact that our SGR component, due to instance-like supervision through connected components, learns richer features that when transferred to downstream tasks improve performance in object detection and instance segmentation.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 1

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 3

[3] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, pages 433–442, 2019. 1, 2, 4, 5

[4] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1

[5] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NIPS*, 34:17864–17875, 2021. 1, 2, 3, 4

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1

[7] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 1, 2

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2, 5

[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 *ICCV*, pages 10012–10022, 2021. 1

[10] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, pages 1139–1147. PMLR, 2013. 1

[11] 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. 1

[12] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, pages 173–190. Springer, 2020. 1, 2

[13] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, pages 7151–7160, 2018. 1

[14] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 1