Hierarchical Lovász Embeddings for Proposal-free Panoptic Segmentation – Supplementary Material –

Tommi Kerola[1] Jie Li[2] Atsushi Kanehira[1] Yasunori Kudo[1] Alexis Vallet[1] Adrien Gaidon[2]
[1]Preferred Networks, Inc. [2]Toyota Research Institute (TRI)

A. Evaluation Metrics

For completeness, and making the paper self-contained, we review the standard panoptic segmentation metrics used for our experiments in this section.

We evaluate our method using the standard metric panoptic quality (PQ) [24] as well as its variations, $PQ^{\dagger}$ [43] and parsing covering (PC) [53]. PQ formulates the quality of the predicted panoptic segmentation in terms of intersection over union (IoU), true positives (TP), false positives (FP) and false negatives (FN).

$$PQ = \frac{\sum_{(p,q)\in\texttt{TP}} \mathrm{IoU}(p,q)\mathbb{1}_{\mathrm{IoU}(p,q)>0.5}}{|\texttt{TP}| + \frac{1}{2}|\texttt{FP}| + \frac{1}{2}|\texttt{FN}|} , \tag{A}$$

where (p, q) is the tuple of the predicted and ground-truth mask, respectively. Additionally, thing- and stuff-specific PQ are denoted as PQ_{th} and PQ_{st}. While popular in the literature, the PQ metric has two downsides that has been pointed out in previous work [43, 53].

First, the PQ metric is harsh towards stuff classes and requires IoU overlap larger than 0.5 even for stuff, treating it like an instance. The $PQ^{\dagger}$ metric [43] aims to mitigate this by relaxing the IoU threshold to 0 for stuff classes, and calculates the PQ metric as usual for thing classes. Second, the PQ metric treats objects the same regardless of size, making it very sensitive to small false positives.

The parsing covering (PC) [53] metric targets applications where larger objects are more important, such as autonomous driving, and is defined as

$$PC = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \frac{\sum_{R\in\mathcal{R}_c} |R| \max_{R'\in\mathcal{R'}_c} \mathrm{IoU}(R', R)}{\sum_{R\in\mathcal{R}_c} |R|} , \tag{B}$$

where $\mathcal{R}_c$, $\mathcal{R'}_c$ are the ground-truth and predicted regions of class c, respectively.

B. Additional Metrics Experimental Results

In this section, we describe the experimental results on Cityscapes with the alternative panoptic segmentation metrics $PQ^{\dagger}$ [43] and parsing covering (PC) [53], as well as the related tasks semantic segmentation and instance segmentation metrics $mIoU$ and AP. We include PQ_{th} to facilitate comparison with AP.

The results for $PQ^{\dagger}$ and PC can be seen in Tables A and B. We can see that our proposed method is able to get competitive results in terms of $PQ^{\dagger}$ and PC, even when comparing with proposal-based methods. Particularly, we note that our method outperforms the method of Porzi et al. [43] in terms of the $PQ^{\dagger}$ metric, indicating that our model is handling stuff classes well, which is also illustrated by our model outperforming all others in terms of the PQ_{st} stuff metric. Our method being competitive in terms of the PC metric indicates that large objects are segmented well.

In Table C, we report the sub-task metrics mIoU and AP for reference. We noticed that although our method achieves similar PQ_{th} with the others, the gap in AP is relatively apparent. The difference between PQ and AP in evaluating instance segmentation performance lies in how the acceptance threshold for objectness score (or detection confidence) is handled. Unlike PQ, which uses a fixed threshold, the AP metric relies heavily on the score estimation of each instance mask to be able to estimate the optimal threshold during evaluation. Notably, PQ is a quite different metric from AP, where false positives matter a lot. Consider the definition of the AP metric:

$$AP = \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \max_{\hat{r}\geq r} P(\hat{r}) , \tag{C}$$

where $P(r)$ is the precision at recall r and $\mathcal{R}$ is the set of recall levels. The AP evaluation protocol uses all possible thresholds that provide the requested recall levels in order to evaluate the model, while PQ evaluation requires finding a single threshold (*e.g.* $r = r_0$) that works for all images in the dataset. Arguably, it can be said that PQ evaluation is closer to representing the performance of the model in a real production environment, where a single fixed threshold must be set for inference. It would be an interesting future direction to explore how we can improve AP at the same time under our algorithm paradigm.

Method	Backbone	Pretrain.	$PQ^{\dagger}$
Proposal-based			
Seamless [43]	ResNet50	ImageNet	59.6
Proposal-free			
HLE (Ours)	ResNet50	ImageNet	**61.3**

Table A. Single-scale experimental results on the Cityscapes validation set.

Method	Backbone	Pretrain.	PC
Proposal-free			
DeeperLab [53]	Xception71	ImageNet	75.6
DeeperLab [53]	Wider MNV2	ImageNet	74.0
DeeperLab [53]	L. W. MNV2	ImageNet	67.9
HLE (Ours)	ResNet50	ImageNet	**76.6**

Table B. Single-scale experimental results on the Cityscapes validation set.

Method	Backbone	Pretrain.	$mIoU$	AP	PQ_{th}
Proposal-based					
Seamless [43]	ResNet50	ImageNet	**77.5**	33.6	**56.1**
Real-time PS [20]	ResNet50	ImageNet	77.0	29.8	52.1
UPSNet [52]	ResNet50	ImageNet	75.2	33.3	54.6
Pan. FPN [23]	ResNet50	ImageNet	75.0	32.0	51.6
Attn.-Guid. [29]	ResNet50	ImageNet	73.6	33.6	52.7
Li et al. [28]	ResNet101	ImageNet	71.6	24.3	39.6
PANet [32]	ResNet50	ImageNet	-	**36.5**	-
Proposal-free					
SSAP [15]	ResNet50	ImageNet	-	**32.8**	-
HLE (Ours)	ResNet50	ImageNet	**77.3**	23.9	**51.1**

Table C. Single-scale experimental results on the Cityscapes validation set.

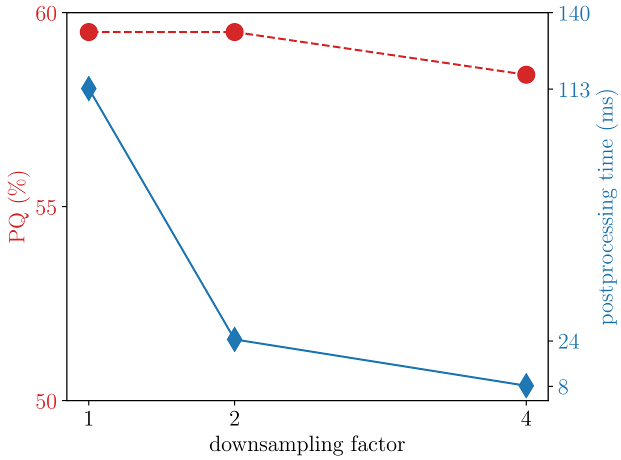

Figure A. Postprocessing time (solid line) vs. PQ (dashed line) as a function of the downsampling factor used in the postprocessing.

C. Alternative Postprocessing Algorithm with Downsampling

In this section, we discuss an alternative postprocessing algorithm which increases speed at small cost of PQ. It is possible to speed up the postprocessing of our method by operating on a downsampled version of the embedding space. The resulting postprocessing time and how it affects Cityscapes validation set PQ can be seen in Figure A. The downsampling factor refers to how much smaller the spatial size of the embedding space we operate postprocessing on becomes. For example, a 1024×2048 size embedding space with downsampling factor 4 becomes 256×512, reducing postprocessing time to 8 ms, while only reducing Cityscapes validation set PQ to 58.4. This simple modification of the postprocessing algorithm can increase inference speed at slight cost of accuracy. Therefore, it can be decided whether to weigh accuracy or speed higher, or have a mixture of both, with this simple modification.