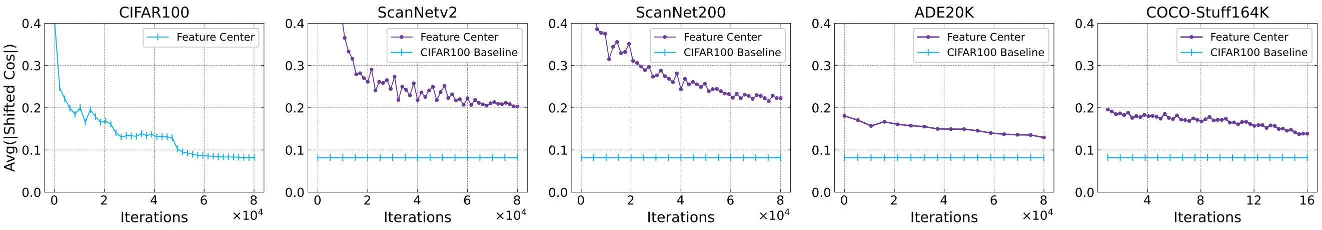
Understanding Imbalanced Semantic Segmentation Through Neural Collapse

Supplementary Material

A. More Results of Neural Collapse in Semantic Segmentation

In Sec. 3.2, we discuss neural collapse in semantic segmentation. We mainly verify the equiangular property of the feature centers and classifier weights, *i.e.*, the values of $\mathrm{Std}_{k\neq k'}(\cos(\hat{\mathbf{z}}_k, \hat{\mathbf{z}}_{k'}))$ and $\mathrm{Std}_{k\neq k'}(\cos(\hat{\mathbf{w}}_k, \hat{\mathbf{w}}_{k'}))$ for semantic segmentation are much larger than those for classification (as shown in Fig. 2). In this part, we follow Papyan et al. [49] and analyze the maximally separated property of the feature centers.

In [49], feature centers approach maximal-angle equiangularity as training progresses. To measure the maximal-angle degree, Papyan et al. calculate the average of shifted cosine across all distinct classes during the whole training process. Mathematically, denote $\mathrm{Avg}_{k,k'}|(\cos(\hat{\mathbf{z}}_k, \hat{\mathbf{z}}_{k'}) + 1/(K-1)|$. As training progresses, the convergence of these values to zero implies that all cosines converge to $-1/(K-1)$. This corresponds to the maximum separation possible for globally centered, equiangular vectors. We carry on a similar experiment on various semantic segmentation datasets. As shown in Fig. 5, the average values for semantic segmentation are two to three times larger than those for classification during the terminal phase of training. It means the feature centers in semantic segmentation are farther away from the maximal separation.



(a) (b) (c) (d) (e)

Figure 5. We plot in the vertical axis of each cell the quantities $\mathrm{Avg}_{k,k'}|(\cos(\hat{\mathbf{z}}_k, \hat{\mathbf{z}}_{k'}) + 1/(K-1)|$ on a classification dataset (a) and different semantic segmentation datasets (b-e). The feature centers in semantic segmentation are more difficult to reach the maximally separated structure of neural collapse than classification.

Taken together, Fig. 2 and Fig. 5 give evidence that both the feature centers and the classifier weights for semantic segmentation are harder than those for classification to converge to an equiangular and maximal separated structure. Moreover, the feature centers suffer a more difficult issue of converging to an ETF structure (neural collapse) than the classifier weights. We point out that, different from classification, semantic segmentation naturally brings contextual correlation and imbalanced distribution among classes, which may break the symmetric structure of neural collapse for both feature centers and classifiers. Noting that such an equiangular and maximally separated feature distribution will bring great benefit to the minor classes, we thus propose a feature center collapse regularizer to achieve this goal.

Experiment setting details of Fig. 2 and Fig. 5. For the CIFAR-100 classification task, we train a ResNet-101 [28] model. For the ScanNetv2 and ScanNet200 point cloud semantic segmentation tasks, we train two MinkoswskiNet-34 [15] models. For the ADE20K and COCO-Stuff164K image semantic segmentation tasks, we train two DeepLabv3+ [14] ResNet-101 models. All training hyperparameters and optimizers are followed the conventional training setting. The total number of iterations for the above five tasks are 80K, 80K, 80K, 80K, and 160K, respectively. All neural collapse statistics computations are following [49].

B. Proof for Lemma 1: Equiangular & Maximal Separated Property

Sufficiency. Since $\mathbf{W}$ is a normalized matrix, *i.e.*, $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_K] \in \mathbb{R}^{d\times K}, d \geq K, \mathbf{w}_k^\top \mathbf{w}_k = 1, \forall k = 1, ..., K$, we have $\cos(\mathbf{w}_k, \mathbf{w}_{k'}) = \mathbf{w}_k^\top \mathbf{w}_{k'}$. We construct the following form:

$$(\mathbf{W}\mathbf{1}_K)^\top (\mathbf{W}\mathbf{1}_K) = \mathbf{1}_K^\top \mathbf{W}^\top \mathbf{W}\mathbf{1}_K = \underbrace{\sum_{k=1}^{K}\sum_{k'=1}^{K} \mathbf{w}_k^\top \mathbf{w}_{k'}}_{K^2 \text{ terms}} = \underbrace{\sum_{k=1}^{K} \mathbf{w}_k^\top \mathbf{w}_k}_{K \text{ terms}} + \underbrace{\sum_{k\neq k'} \mathbf{w}_k^\top \mathbf{w}_{k'}}_{K(K-1) \text{ terms}} \geq 0. \tag{9}$$

Recalling that $\mathbf{w}_k^\top \mathbf{w}_k = 1, \forall k = 1, ..., K$, we have $\sum_{k\neq k'} \mathbf{w}_k^\top \mathbf{w}_{k'} \geq -K$. Thus we can get:

$$\max_{k\neq k'} \cos(\mathbf{w}_k, \mathbf{w}_{k'}) = \max_{k\neq k'} \mathbf{w}_k^\top \mathbf{w}_{k'} \geq -\frac{K}{K(K-1)} = -\frac{1}{K-1}. \tag{10}$$

Thus, the maximal separation value, $\max_{k\neq k'} \cos(\mathbf{w}_k, \mathbf{w}_{k'})$, is greater or equals to $-\frac{1}{K-1}$. In the following, we will give a proof that $\max_{k\neq k'} \cos(\mathbf{w}_k, \mathbf{w}_{k'}) = -\frac{1}{K-1}$. Here we let:

$$\mathbf{W} = \sqrt{\frac{K}{K-1}}\mathbf{U}\left(\mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right), \tag{11}$$

where $\mathbf{U}$ is a rotation matrix satisfied $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_K$. We can calculate:

$$\mathbf{W}^\top\mathbf{W} = \frac{K}{K-1}\left(\mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)^\top \left(\mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right) = \begin{bmatrix} 1 & -\frac{1}{K-1} & \cdots & -\frac{1}{K-1} \\ -\frac{1}{K-1} & 1 & \cdots & -\frac{1}{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{K-1} & -\frac{1}{K-1} & \cdots & 1 \end{bmatrix}. \tag{12}$$

According Eq. (12), it means that $\mathbf{w}_k^\top \mathbf{w}_k = 1, \forall k = 1, ..., K$, and $\mathbf{w}_k^\top \mathbf{w}_{k'} = -\frac{1}{K-1}, \forall k \neq k'$. When $\mathbf{W}$ satisfies Eq. (11) and is simplex ETF structured, the equality in Eq. (10) holds. Obviously, it enjoys the equiangular property, *i.e.*, $\forall k \neq k', \cos(\mathbf{w}_k, \mathbf{w}_{k'}) = -\frac{1}{K-1}$, where $\frac{1}{K-1}$ is a constant.

Necessity. Since $\max_{k\neq k'} \cos(\mathbf{w}_k, \mathbf{w}_{k'}) = -\frac{1}{K-1}$, we have $\cos(\mathbf{w}_k, \mathbf{w}_{k'}) \leq -\frac{1}{K-1}, \forall k \neq k'$, and then $\sum_{k\neq k'} \cos(\mathbf{w}_k, \mathbf{w}_{k'}) \leq -K$. Given Eq. (9), we have $\sum_{k\neq k'} \cos(\mathbf{w}_k, \mathbf{w}_{k'}) \geq -K$. As a result, we have $\sum_{k\neq k'} \cos(\mathbf{w}_k, \mathbf{w}_{k'}) = -K$, and the following equation,

$$\mathbf{W}^\top\mathbf{W} = \begin{bmatrix} 1 & -\frac{1}{K-1} & \cdots & -\frac{1}{K-1} \\ -\frac{1}{K-1} & 1 & \cdots & -\frac{1}{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{K-1} & -\frac{1}{K-1} & \cdots & 1 \end{bmatrix}. \tag{13}$$

We define $\mathbf{M} = \sqrt{\frac{K-1}{K}}\mathbf{W}$. We have:

$$\mathbf{M}^\top\mathbf{M} = \frac{K-1}{K}\mathbf{W}^\top\mathbf{W} = \frac{K-1}{K}\begin{bmatrix} 1 & -\frac{1}{K-1} & \cdots & -\frac{1}{K-1} \\ -\frac{1}{K-1} & 1 & \cdots & -\frac{1}{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{K-1} & -\frac{1}{K-1} & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \frac{K-1}{K} & -\frac{1}{K} & \cdots & -\frac{1}{K} \\ -\frac{1}{K} & \frac{K-1}{K} & \cdots & -\frac{1}{K} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{K} & -\frac{1}{K} & \cdots & \frac{K-1}{K} \end{bmatrix} = \mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K.$$

Note that $\mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K$ is the centering matrix $\mathbf{C}_K$, which has the eigenvalue 1 of multiplicity $K-1$ and eigenvalue 0 of multiplicity 1. Note that $\mathbf{C}_K^2 = \mathbf{C}_K^\top\mathbf{C}_K = \mathbf{C}_K$, we conclude that, $\mathbf{C}_K = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{V} = \left[\mathbf{V}', \frac{1}{\sqrt{K}}\mathbf{1}_K\right]$, and $\mathbf{V}'(\mathbf{V}')$ is the projector on the $(K-1)$-dimension subspace perpendicular to $\frac{1}{\sqrt{K}}\mathbf{1}_K$, *i.e.*,

$$\mathbf{V}'(\mathbf{V}')^\top = \mathbf{C}_K, \quad \mathbf{\Sigma} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}.$$

Due to the uniqueness of the orthogonal projector, $\mathbf{M}$ shares the same column space as $\mathbf{C}_K$, namely, we have,

$$\mathbf{M} = \mathbf{U}_M \mathbf{\Sigma} \left[\mathbf{V}' \mathbf{Q}_{k-1}, \frac{1}{\sqrt{K}} \mathbf{1}_K \right]^\top = \mathbf{U}_M \mathbf{\Sigma} \mathbf{Q}^\top \mathbf{V}^\top, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{k-1} & \\ & 1 \end{bmatrix},$$

where $\mathbf{Q}_{K-1} \in \mathbb{R}^{K-1 \times K-1}$ and $\mathbf{U}_M \in \mathbb{R}^{d \times K}$ are the arbitrary orthogonal matrices. It is easy to verify that $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$ and $\mathbf{\Sigma}\mathbf{Q}^\top = \mathbf{Q}^\top \mathbf{\Sigma}$, thus we have,

$$\mathbf{M} = \mathbf{U}_M \mathbf{\Sigma} \mathbf{Q}^\top \mathbf{V}^\top = \mathbf{U}_M \mathbf{Q}^\top \mathbf{\Sigma} \mathbf{V}^\top = \mathbf{U}_M \mathbf{Q}^\top \mathbf{V}^\top \mathbf{V} \mathbf{\Sigma} \mathbf{Q}^\top \mathbf{V}^\top = \mathbf{U}_M \mathbf{Q}^\top \mathbf{V}^\top \mathbf{C}_K.$$

We let $\mathbf{U} = \mathbf{U}_M \mathbf{Q}^\top \mathbf{V}^\top$, and we can get,

$$\mathbf{U}_M^\top \mathbf{U}_M = \mathbf{V}\mathbf{Q}\mathbf{U}^\top \mathbf{U} \mathbf{Q}^\top \mathbf{V}^\top = \mathbf{I}_K.$$

Hence, we can conclude that $\mathbf{M} = \mathbf{U}\mathbf{C}_K$, where $\mathbf{U}$ is an orthogonal matrix. Finally, due to $\mathbf{M} = \sqrt{\frac{K-1}{K}} \mathbf{W}$,we prove the solution of Eq. (13) is

$$\mathbf{W} = \sqrt{\frac{K}{K-1}} \mathbf{M} = \sqrt{\frac{K}{K-1}} \mathbf{U} \left(\mathbf{I}_K - \frac{1}{K} \mathbf{1}_K \mathbf{1}_K^\top \right).$$

In summary, a normalized matrix is maximal equiangular separated ***if and if only*** the matrix is a simplex ETF. □

C. Detailed Empirical Evidence of Better Rebalance in CeCo

In Sec. 4.3, we list the main conclusions about better rebalance in CeCo. Here, we give a more detailed discussion.

Reduction the degree of imbalance. In our center collapse regularizer, we get two important parameters, feature centers $\bar{\mathbf{Z}}$ and center's labels $\bar{\mathbf{y}}$. Semantic segmentation datasets naturally suffer a more severe class imbalance issue. However, We observe that the degree of feature center imbalance (the distribution of $\bar{\mathbf{y}}$) is greatly relieved than that of point/pixel imbalance (the distribution of $\mathbf{y}$). We collect these two statistics from the most popular semantic segmentation benchmarks. Following the convention [19, 42], we calculate the imbalance factor (IF) as $\beta = \frac{n_{\max}}{n_{\min}}$, where $n_{\max}$ and $n_{\min}$ are the numbers of training samples for the most frequent class and the least frequent class. As shown in Fig. 6, for ADE20K, the point/pixel imbalance factor (PIF) is nearly three times the center imbalance factor (CIF). For COCO-Stuff164K, PIF is nearly five times RIF. For ScanNet-v2, PIF is nearly 10 times CIF. For ScanNet200, PIF is nearly 60 times CIF. As our method relieves the class imbalance issue via the center collapse branch, it conveniently benefits from less imbalance than point/pixel would, and this in turn promotes more effective rebalancing for semantic segmentation.

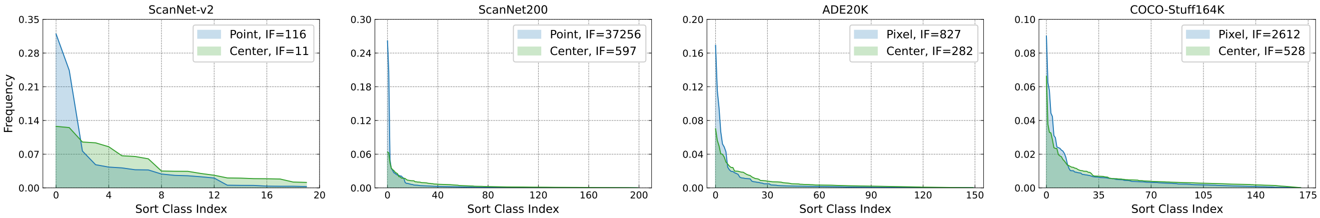
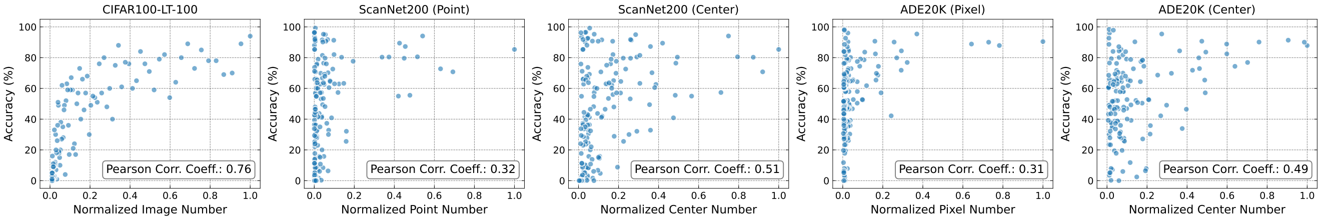


Figure 6. Comparison between point/pixel imbalance and class center imbalance. Histogram statistics of point/center frequency over sorted class indexes on ScanNet-v2, ScanNet200 (left), and pixel/center frequency on ADE20K, COCO-Stuff164K (right). The imbalance factor (IF) is defined as $\frac{n_{\max}}{n_{\min}}$, where $n_{\max}$ and $n_{\min}$ are the numbers of samples (points, pixels, centers) for the most frequent class and the least frequent class of dataset. It can be seen that the center imbalance is greatly alleviated compared with the point/pixel imbalance.

Reducing the contextual correlation and improving the effective number. In this part, we will analyze the correlation influence in the center regularization branch. For any $\bar{\mathbf{z}}_k$, it relates to n_k points/pixels from the original input in the point/pixel branch. Thus, $\bar{\mathbf{z}}_k$ is a higher-level semantic feature and contains more general information compared with the point/pixel-level feature $\mathbf{z}_i$. For the point/pixel-level pair $(\mathbf{z}, \mathbf{y})$, as we mentioned in the introduction part and Fig. 7, plenty of common information (similar color, close position) is shared among neighboring pixels/points. Neighboring points/pixels are highly correlated, which leads to the class accuracy having a lower correlation with the number of points/pixels. By contrast, the class center pair $(\bar{\mathbf{z}}, \bar{\mathbf{y}})$, is a higher and more global level semantic representation, which increases diversity between different samples. To proof this, we calculate the Pearson correlation coefficient [7] between the class accuracy $\mathbf{a} \in \mathbb{R}^K$ and the class frequency $\mathbf{f} = [\mathbf{f}_1, ..., \mathbf{f}_K] \in \mathbb{R}^K, \mathbf{f}_k = \frac{n_k}{n_{\max}}$:

$$\rho_{\mathbf{a},\mathbf{f}} = \frac{\mathrm{Cov}(\mathbf{a}, \mathbf{f})}{\sigma(\mathbf{a}) \cdot \sigma(\mathbf{f})} = \frac{\mathbb{E}[(\mathbf{a} - \mu(\mathbf{a}))(\mathbf{f} - \mu(\mathbf{f}))]}{\sigma(\mathbf{a}) \cdot \sigma(\mathbf{f})},$$

where $\mu(\cdot)$, $\sigma(\cdot)$, $\mathrm{Cov}(\cdot)$ and $\mathbb{E}(\cdot)$ are the functions for mean, standard deviation, covariance, and expectation, respectively.



(a) (b) (c) (d) (e)

Figure 7. Illustration of effective number for recognition and semantic segmentation. (a): Class accuracy is **positively** correlated with the class image number (normalized by the maximum image number among classes). (b) and (d): Class accuracy is **weakly** correlated with the class point/pixel number (normalized by the maximum point/pixel number among classes). (c) and (e): Class accuracy is **relatively** correlated with the class center number (normalized by the maximum center number among classes).

In the bottom right part of Fig. 7, we list the Pearson coefficients between class accuracy and image frequency on CIFAR-100-LT, pixel frequency on ADE20K, point frequency on ScanNet200, and feature center frequency on ScanNet200. The correlation between class accuracy and image frequency is the largest. Many previous studies proposed class-balanced strategies [10, 19, 30, 45, 57] guided by class frequency, and achieved good results In long-tailed image recognition. Feature center frequency has a stronger correlation with class accuracy than point/pixel frequency, as it eliminates the effects of correlations among neighboring pixels, further indicating its greater suitability for semantic segmentation rebalancing.

Different from the image classification task (CIFAR100-LT, see Fig. 7a), the category accuracy has a **low correlation** with the number of pixels/points (ScanNet200 and ADE20K, Fig. 7b and 7d). The reason may lie in the effective number of training samples [19], which is defined as the number of samples that contribute. Unlike image classification where each image is a unique instance, two pixels in a similar context may contribute similarly in training semantic segmentation models, and thus the actual number of pixels may not necessarily be the number of effective pixels. Any two different images of the same class offer significantly different training signals, whereas in semantic segmentation two pixels of the same semantic class may contribute similarly. By contrast, the center regularization branch significantly reduces the imbalance severity to increase the effective number, as shown in Fig. 7c and 7e.

D. Datasets Description and Implementation Details

D.1. Datasets

ScanNet200. The ScanNet [20] Benchmark has provided an active online benchmark evaluation for 3D semantic segmentation, but only considers 20 class categories, which is insufficient to capture the diversity of many real-world environments. Thus, Rozenberszki et al. presented the ScanNet200 [59] Benchmark Benchmark for 3D semantic segmentation with 200 class categories, an order of magnitude more than the previous. In order to better understand performance under the natural class imbalance of the ScanNet200 benchmark, they further split the 200 categories into sets of 66, 68, and 66 categories, based on the frequency number of labeled surface points in the train set: head, common, and tail respectively.

ADE20K. ADE20K [77] is a challenging dataset often used to validate transformer-based neural networks on downstream tasks such as semantic segmentation. It contains 22K densely annotated images with 150 fine-grained semantic concepts. The training and validation sets consist of 20K and 2K images, respectively.

COCO-Stuff164K. COCO-Stuff164K [9] is a large-scale scene understanding benchmark that can be used for evaluating semantic segmentation, object detection, and image captioning. It includes all 164K images from COCO 2017. The training and validation sets contain 118K and 5K images, respectively. It covers 171 classes: 80 thing classes and 91 stuff classes.

D.2. Implementation Details

3D semantic segmentation. We implement CeCo based on the CSC [29] and LG [59] codebase. We use the SGD optimizer with a batch size of 32 for a total of 20K steps. The initial learning rate is around 0.1, with polynomial decay with a power of 0.9. For all experiments, we use data parallel on 8 GPUs. For the backbone, we follow [29, 59] use the same and widely-used voxel-based network, MinkowskiNet-34 [15].

2D semantic segmentation. We implement the proposed method in the mmsegmentation codebase [1] and follow the commonly used training settings for each dataset. More details are described in the following.

For backbones, we use CNN-based ResNet-50c and ResNet-101c, which replace the first 7 × 7 convolution layer in the original ResNet-50 and ResNet-101 with three consecutive 3×3 convolutions. Both are popular in the semantic segmentation community [75]. For OCRNet, we adopt HRNet-W18 and HRNet-W48 [65]. For transformer-based neural networks, we adopt the popular Swin transformer [41] and BEiT [5]. BEiT achieves the most recent state-of-the-art performance on the ADE20K validation set. It is worth noting that Swin-B is pre-trained on ImageNet-22K.

With CNN-based models, we use SGD and the poly learning rate schedule [69, 75] with an initial learning rate of 0.01 and a weight decay of 0.001. If not stated otherwise, we use a crop size of 512 × 512, a batch size of 16, and train all models for 160K iterations on ADE20K and 320K iterations on COCO-Stuff164K. For the Swin transformer and BEiT, we use their default optimizer, learning rate setup, and training schedule. In the training phase, the standard random scale jittering between 0.5 and 2.0, random horizontal flipping, random cropping, as well as random color jittering are used as data augmentations [1]. For inference, we report the performance of both single-scale (s.s.) inference and multi-scale (m.s.) inference with horizontal flips and scales of 0.5, 0.75, 1.0, 1.25, 1.5, 1.75.

E. Visual Comparison

In this section, we demonstrate the advantages of CeCo with quantitative visualizations on ScanNet200, ADE20K, and COCOStuff164K shown in Figures 8, 9, and 10, respectively. We observe that CeCo well captures the contextual and geometry information and obtains more precise semantic segmentation masks for both common and tail classes.

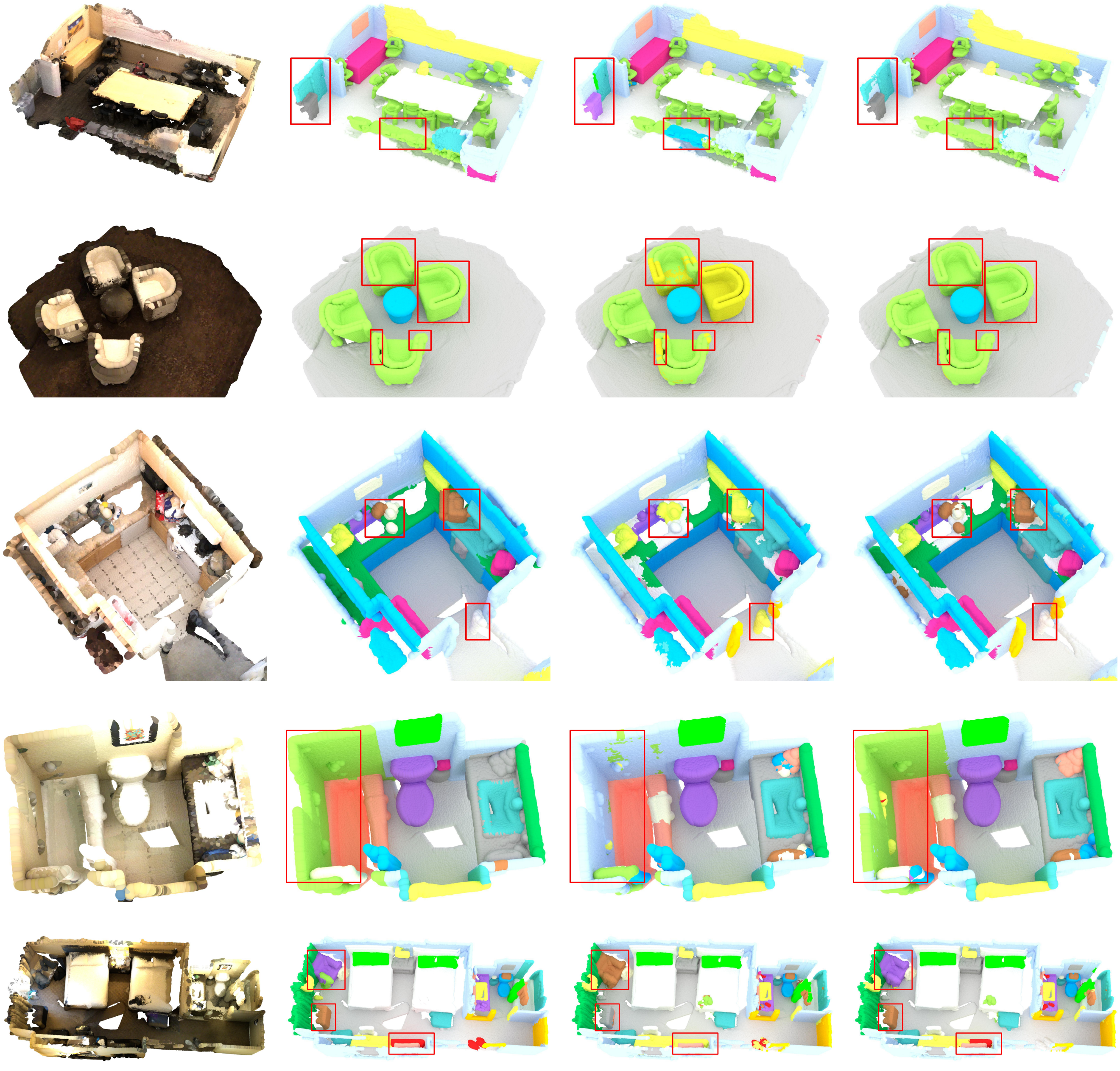


Figure 8. Visualization comparisons between the CE baseline and CeCo on ScanNet200.

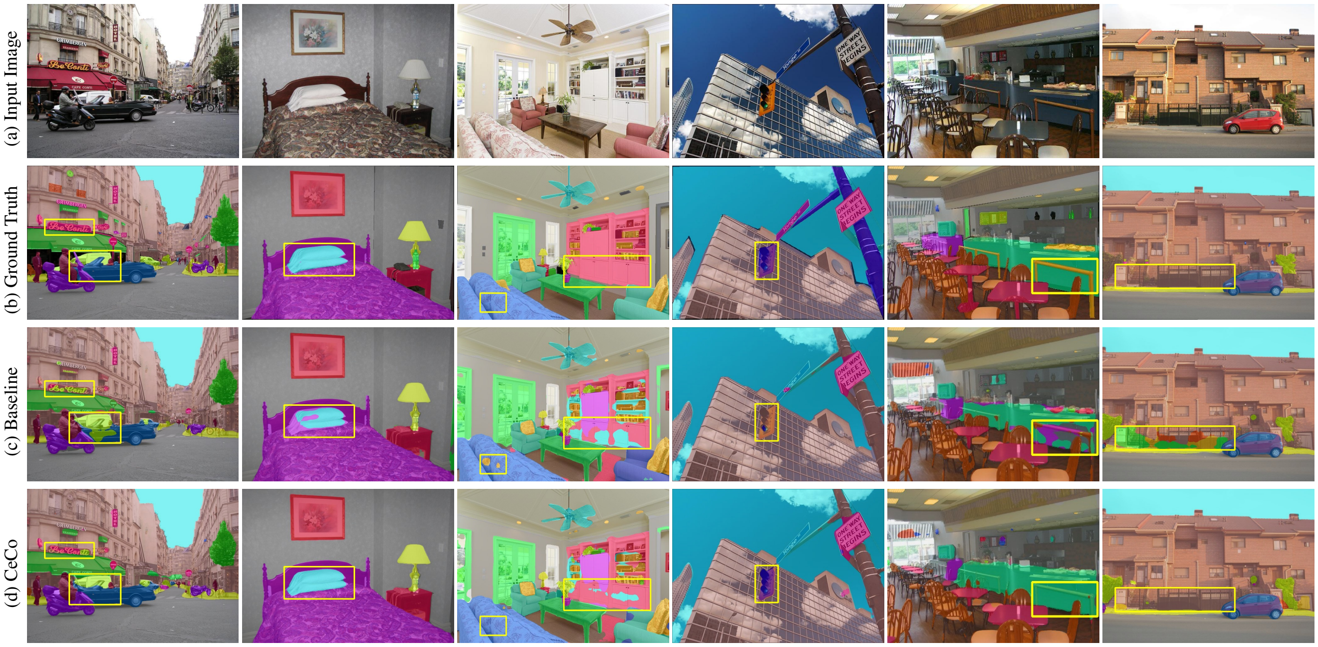


Figure 9. Visualization comparisons between the CE baseline and CeCo on ADE20K.

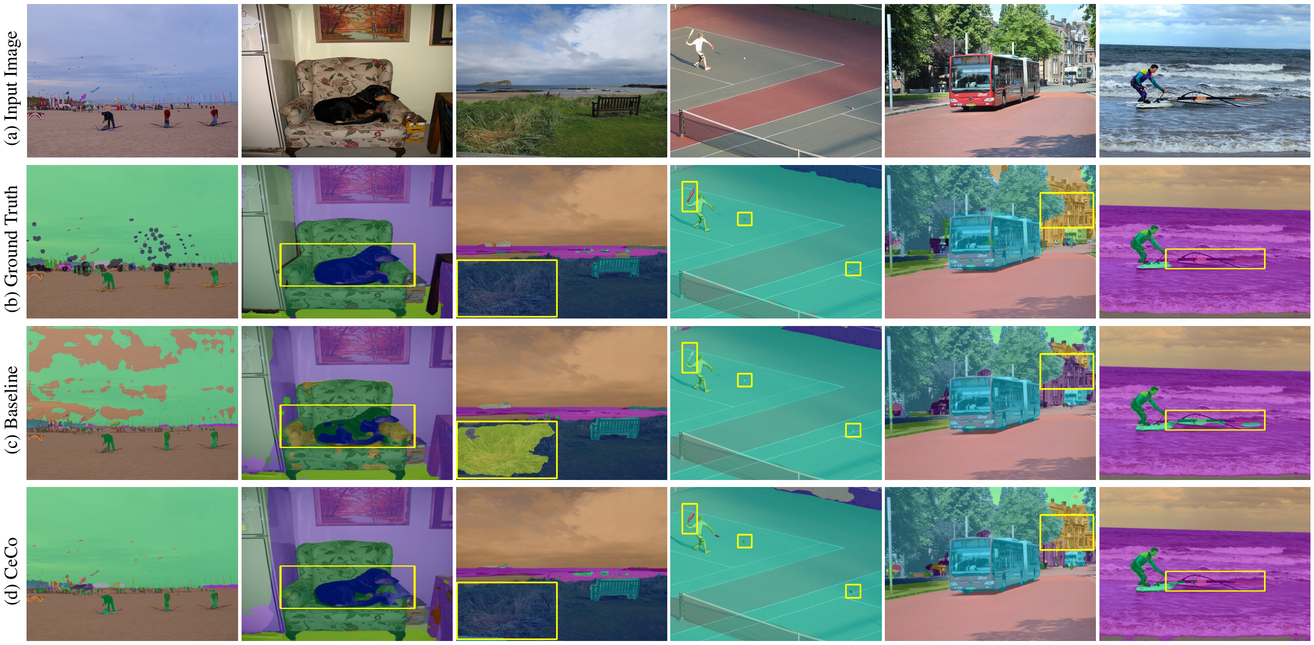


Figure 10. Visualization comparisons between the CE baseline and CeCo on COCO-Stuff164K.

F. Limitation Analysis

F.1. Simple Imbalance Cases

This work presents a new regularization for imbalanced semantic segmentation. However, when the class number of the dataset is small enough or the imbalanced severity is negligible, the performance of CeCo is quite incremental and even degraded. Here we use Cityscapes and ScanNet v2, two datasets as examples. ScanNet v2 just includes 20 classes, and Cityscapes only contains 30 classes. Following the above analysis, we compute the imbalanced factors on both datasets. As shown in Table 12, the center imbalanced ratios for these two datasets are much smaller than those on ScanNet200, ADE20K, and COCO-Stuff164K. Under simple imbalance cases, the improvement of CeCo is quite limited. The detailed performance results are shown in Table 9 and Table 10.

Method	Backbone	mIoU (s.s.)	mIoU (m.s.)
DLV3P	ResNet-18	76.3	77.9
+ CeCo	ResNet-18	77.5 **(+1.2)**	79.2 **(+1.3)**
DLV3P	ResNet-50	79.8	81.4
+ CeCo	ResNet-50	80.3 **(+0.5)**	81.5 **(+0.1)**

Table 9. Performance on Cityscapes.

Method	mIoU
PointTransformer [74]	70.6
SparseConvNet [24]	69.3
MinkowskiNet [15]	73.0
+ CeCo	73.7 **(+0.7)**

Table 10. Performance on ScanNet v2.

Method	Training	Inference
DLV3P (ResNet-101)	0.725s	0.294s
+ CeCo	0.793s **(+9.37%)**	0.294s
MinkowskiNet-34	5.844s	4.856s
+ CeCo	6.441s **(+10.2%)**	4.856s

Table 11. Training and inference time (batch-level) comparison.

Dataset	Imbal. Factor
Cityscapes (point)	373
Cityscapes (center)	**20**
ScanNet v2 (point)	116
ScanNet v2 (center)	**11**

Table 12. Imbalanced factor comparison.

F.2. Training Time and Inference Time Analysis

As we discussed in Sec. 4.2, CeCo can be easily integrated into any off-the-shelf segmentation architecture. In the evaluation of our method, we only preserve the point/pixel recognition branch. It means that just the point/pixel classifier is preserved, while the center regularization branch is discarded. Therefore, the evaluation of CeCo is very efficient. We list the training and inference time results in Table 11. All experiments are conducted on the Nvidia GeForce 2080Ti GPU. For DLV3P, we use ResNet-101 as the backbone with a batch of two images. For MinkowskiNet-34, we set the batch size to four. Although our method can greatly improve the imbalance performance and is efficient in inference, the introduction of the additional center regularization branch increases the training time cost by about 6 to 13%.