A APPENDIX

A.1 BENCHMARKS & IMPLEMENTATION DETAILS

Benchmarks: Cityscapes (Cordts et al., 2016) is a dataset that focuses on semantic understanding of urban street scenes and contains 19 semantic classes. It contains 5K annotated images with pixel-level fine annotations and 20K coarsely annotated images. The finely annotated 5K images are split into sets with numbers 2975, 500 and 1525 for training, validation and testing.

ADE20K (Zhou et al., 2017) is a challenging scene parsing dataset. It contains 150 categories and diverse scenes with 1038 image-level labels. The training and validation sets consist of 20K and 2K images, respectively.

Implementation Details. We initialize backbones with the weights pre-trained on ImageNet. For DeepLabv3+, we using CNN-based ResNet-50c and ResNet-101c as backbones, which switch the first 7×7 convolution layer to three 3×3 convolutions. HRNet-W48 (Wang et al., 2020a) is adopted for OCRNet. MIT-B3 as a Transformer-based backbone is adopted for SegFormer (Xie et al., 2021), both of which are popular in semantic segmentation. For CNN-based network, we use SDG and poly learning rate schedule (Zhao et al., 2017) with factor $\left(1-\frac{iter}{total_{iter}}\right)^{0.9}$. The initial learning rate is set as 0.01 and weight decay is 0.0005. We adopt AdamW with 6×10^{-5} learning rate and 0.01 weight decay. We set the image crop size to 512×1024, batch size as 8 and training iterations as 80K on Cityscapes by default. For ADE20K, the crop size of images is set as 512×512, the batch size is set as 16 and training iterations are set as 80K if not stated otherwise. In the training phase, we augment data samples with the standard random scale in the range of [0.5, 2.0], random horizontal flipping, random cropping, as well as random color jittering. For inference, the input image size of ADE20K is the same as the size during training, but for Cityscapes the input image is scaled to 1024×2048, no tricks(*e.g.*multi-scale with flipping) will be adopted during testing. All experiments are implemented on the Nvidia A6000.

A.2 ADDITIONAL PROOF OF THEORETICAL ANALYSIS BETWEEN MIOU AND MACC

Before we proceed with the additional proof, we introduce some formulas for mAcc and mIoU.

$$\begin{aligned} mAcc &= \frac{1}{C}\sum_{i=1}^{C} Acc_i = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i+FN_i}, \\ mIoU &= \frac{1}{C}\sum_{i=1}^{C} IoU_i = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i+FN_i+FP_i}. \end{aligned} \tag{11}$$

$$\sum_{i=1}^{C} FP_i = \sum_{i=1}^{C} FN_i \tag{12}$$

$$num_i = TP_i + FP_i \tag{13}$$

A.2.1 ADDITIONAL PROOF OF MACC IS THE BETTER TAIL-SENSITIVE METRIC

According to the prior information from experimental statistics, the proportions of pixel instances of the head, body, and tail categories in the semantic segmentation are: 80%, 15% and 5%. We analyzed the items in the mIoU and mAcc formulas and found that the difference between them was the FP_i. According to Eq. 12 We derive

$$\begin{aligned} &\sum_{i=1}^{C} FP_i = \sum_{i=1}^{C} FN_i = \sum^{C_h} FN_i + \sum^{C_b} FN_i + \sum^{C_t} FN_i \\ &\approx \sum^{C_h}(1-Acc_i)\times num_i + \sum^{C_b}(1-Acc_i)\times num_i + \sum^{C_t}(1-Acc_i)\times num_i \\ &= (1-mAcc_h)\sum^{C_h} num_i + (1-mAcc_b)\sum^{C_b} num_i + (1-mAcc_t)\sum^{C_t} num_i \\ &= (1-mAcc_h)\times 0.8NUM + (1-mAcc_b)\times 0.15NUM + (1-mAcc_t)\times 0.05NUM \end{aligned} \tag{14}$$

where, *h*, *b* and *t* refer to head, body, tail categoris and NUM refers to the pixel instance number of overall datasets.

From Eq. 14, we obtain the conclusion that due to the large instance base of the head categories, FN_i and *FN*-related FP_i items are dominated by head categories. Therefore, the IoU_i of each category *i* and *mIoU* will be dominated by head categories because of the item FP_i.

Instead of *mIoU*, for each category *i* of *mAcc*, the items of Acc_i are only related to its own category *i*, and *mAcc* will not be dominated by the head categories. Thus *mAcc* is a fair and tail-sensitive metric.

A.2.2 Additional proof of Remark 1

To better understand the correlation between mean IoU and mean Acc in long-tailed semantic segmentation and **Remark 1**. According to the equation Eq. 10 and the precondition from the experiments. The category Acc and IoU in our method become :

$$\begin{aligned} \hat{IoU}_i &\approx IoU_i, \\ \hat{Acc}_i &= (1+p)ACC_i. \end{aligned} \tag{15}$$

And it is clear from Eq.13, We obtain the result:

$$\begin{aligned} \hat{Acc}_i &= (1+p)ACC_i \\ \Rightarrow \hat{Acc}_i = \frac{\hat{TP}_i}{\hat{TP}_i + \hat{FN}_i} = \frac{\hat{TP}_i}{TP_i + FN_i} &= (1+p)\frac{TP_i}{TP_i + FN_i} = (1+p)ACC_i \\ \Rightarrow \hat{TP}_i &= (1+p)TP_i \end{aligned} \tag{16}$$

$$\begin{aligned} \hat{IoU}_i \approx IoU_i \Rightarrow \hat{IoU}_i = \frac{\hat{TP}_i}{\hat{TP}_i + \hat{FN}_i + \hat{FP}_i} = \frac{\hat{TP}_i}{TP_i + FN_i + \hat{FP}_i} &\approx \frac{TP_i}{TP_i + FN_i + FP_i} = IoU_i \\ \Rightarrow (1+p)TP_i \times (TP_i + FN_i + FP_i) = TP_i \times (TP_i + FN_i + \hat{FP}_i) \\ \Rightarrow (1+p)TP_i \times (num_i + FP_i) = TP_i \times (num_i + \hat{FP}_i) \\ \Rightarrow \Delta FP_i = p \times (num_i + FP_i) \end{aligned} \tag{17}$$

$$Acc_i = \frac{TP_i}{TP_i + FN_i} \;\&\; IoU_i = \frac{TP_i}{TP_i + FN_i + FP_i} \Rightarrow FP_i = \frac{Acc_i}{IoU_i} \times num_i - num_i \tag{18}$$

where ΔFP_i is the increased false positive from baseline to our method.

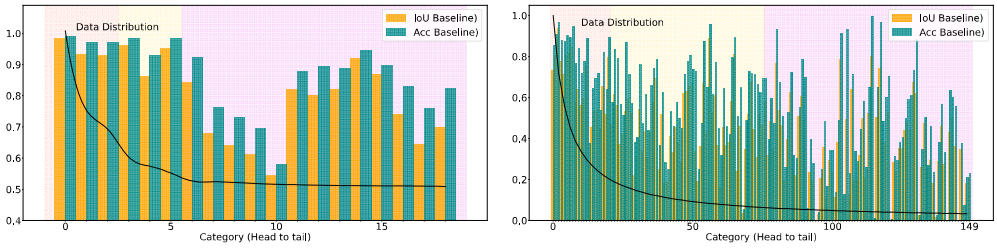
To guarantee the classifier effective and segment more tail categories, it should satisfied:

$$\Delta FP_i = p \times (num_i + FP_i) \ll num_i \tag{19}$$

We observe that the results in Cityscapes benchmark has achieved a high value, *e.g.*IoU_i = 0.8 and Acc_i = 0.85. According to this precondition and Eq. 18, FP_i = 0.0625 $\times$n u m$_i$ and p are both small value, which means ΔFP_i is a minimum value and satisfied Eq. 19. This is just the reason that Acc improved while IoU not decrease is significant for tail categories segmentation.

A.3 Additional proof of the correlation between performance and data distribution

As shown of Figure 6, the baseline method on Cityscapes and ADE20K perform not well on certain categories, and we can clearly see that these categories mainly fall in the tail and body data subsets. This further demonstrates that the long-tailed data distribution limits the overall performance of the baseline method by constraining the accuracy of certain categories.



(a) Data distribution and performance with baseline on Cityscapes.

(b) Data distribution and performance with baseline on ADE20K.

Figure 6: The existence of long-tailed distribution in semantic segmentation, we set baseline as DeepLabv3+ and ResNet-50c by default. (a) and (b) shown the data distributions on Cityscapes and ADE20K are long-tailed, which cause the better performance on the majority categories yet suppress the minority categories. It should be noted that we reordered the categories in Cityscapes according to Pixel Frequency.

A.4 Ablation with Long-tailed methods

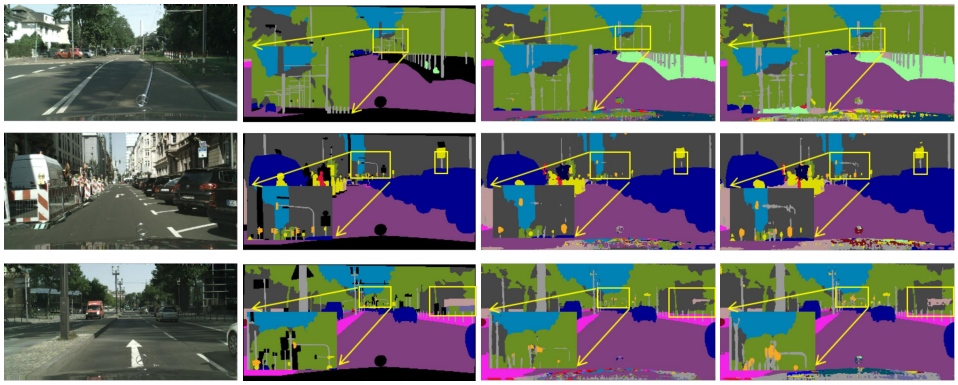
Table 7: Pearson coefficients between categories frequency and accuracy, the lower value means much weaker correlation.

Dataset	$\rho_{X,Y}(\%)$
CIFAR-100	75.9
ADE20K-pixel	36.8
Cityscapes-pixel	58.9

In this section, we first calculate the Pearson correlation coefficient to show the correlation between category accuracy and category frequency (image category frequency on CIFAR-100) in Table 7. The weak correlation between category accuracy and pixel level frequency on Cityscapes and ADE20K causes challenge to re-weighting in semantic segmentation, which can be demonstrated in Table 8. We compared our work with traditional long-tailed classification methods to further explore the contribution of our work. We adopt the re-weighting method to modify the loss function and put larger weights on tail categories. Despite improving mAcc, the re-weighting method caused a lot of decrease in mIoU. For re-sampling, the contextual information is corrupted resulting in segmentation metrics at a very low level. In general, the current Pixel level re-balance approaches can not work well with the long tail distribution on semantic segmentation.

Table 8: Ablation with our method and pixel level re-weighting and re-sampling, all networks adopt ResNet-50c backbone.

Benchmarks	Methods	mIoU	mAcc
Cityscapes	Baseline	80.37	86.68
	Reweighting-FocalLoss (Lin et al., 2017)	76.23(−4.14)	85.00(−1.68)
	Reweighting-LDAMLoss (Cao et al., 2019a)	77.52(−2.85)	85.15(−1.43)
	Reweighting-SeesawLoss (Wang et al., 2021)	67.58(−12.79)	74.35(−12.33)
	Re-sampling	66.79(−13.58)	75.21(−11.47)
	Baseline+MED	80.20(−0.17)	90.04(+3.36)
ADE20K	Baseline	42.11	54.13
	Reweighting-FocalLoss (Lin et al., 2017)	37.61(−4.50)	55.29(+1.16)
	Reweighting-LDAMLoss (Cao et al., 2019a)	40.39(−1.72)	48.78(−5.35)
	Reweighting-SeasawLoss (Wang et al., 2021)	33.87(−8.24)	40.19(−13.94)
	Resampling	-	-
	Baseline+MED	43.82(+1.71)	60.02(+5.89)



(a) Image (b) Groud Truth (c) DeepLabv3+ (d) Ours

Figure 7: Qualitative Visualization results on the validation set of Cityscapes with ResNet-50c as backbone. All the models here are trained under the same setting.

A.5 Analysis of the performance gap between Cityscapes and ADE20K

According to the experiments' results of Sec4, our method has achieved impressive performance in both mIoU and mAcc on ADE20K dataset. However, there seems to be a gap between the performance on Cityscapes and ADE20K. We analyze the causes: 1) Compared to ADE20K, there are fewer categories in Cityscapes and a more pronounced long-tail distribution (higher proportion of head category instances), so it is easier to fall into local optimum when training body and tail experts, resulting in the overfitting of these categories. Finally, it caused that the increase of FN_{ht} and FP_{ht} on the overall datasets. FN_{ij} and FP_{ij} denote to the pixel instance which the ground truth belongs to category i but the prediction is j. 2) The performance of baseline methods on Cityscapes is at a high level. According to the above two reasons, our method improves the overall TP on Cityscapes, but it will increase FN_{ht} of the head categories and FP_{ht} of the tail categories, respectively, resulting in $mIoU$ being at a relatively stable value.

From the macro level of image visualization, as shown in Figure 7, our method segments the surrounding part of the head categories into tail categories while segmenting the tail categories. We believe that such head accuracy decreases are acceptable and meaningful in real-world scenarios.

A.6 Quantitative Visualization Comparisons on Cityscapes and ADE20K

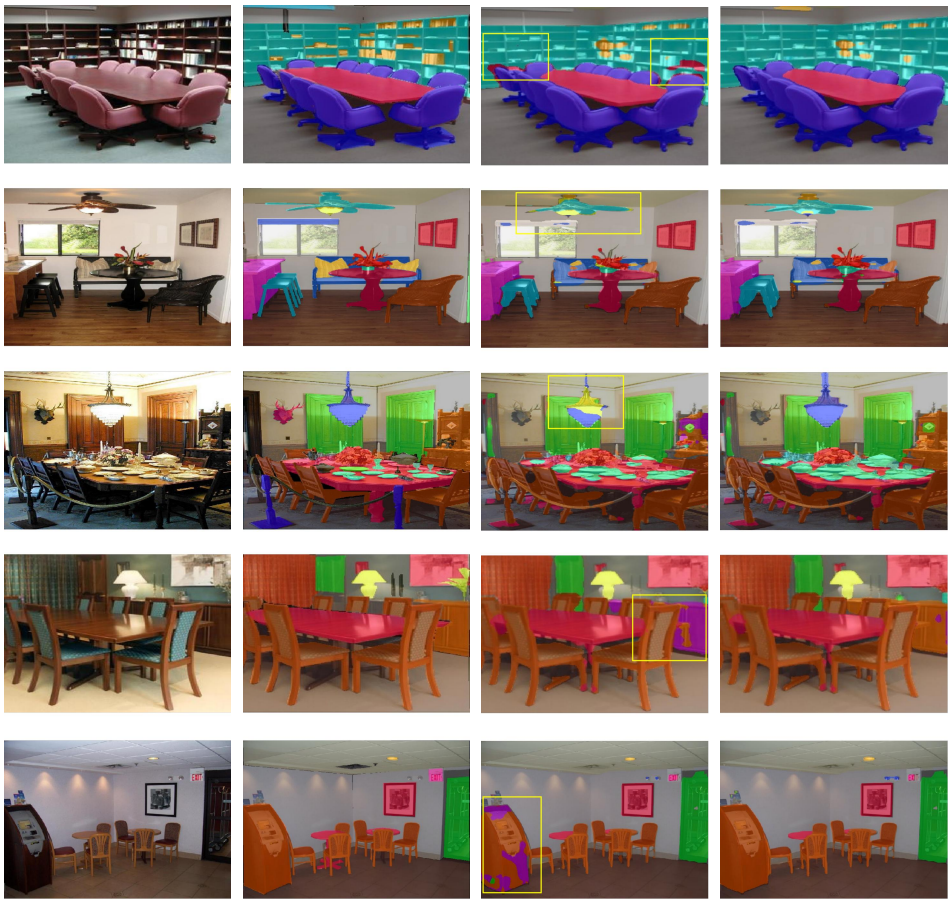
In this section, we demonstrate the better performance of MEDOE framework with quantitative visualization on Cityscapes and ADE20K shown in Figure 7 and 8. We adopt ResNet-50c as backbone and all models trained under the same setting. In most semantic scenarios, our MEDOE method can achieve better performance in segmenting tail categories.

A.7 Additional explanations

A.7.1 Contextual module

Contextual module, as an important module in semantic segmentation, refers to the extraction and aggregation of contextual information for pixels through a series of operations (*i.e.*, feature pyramids, atrous convolution, large-scale convolutional, attention mechanisms, and global pooling). Existing contextual modules include: ASPP (Chen et al., 2018), PSP (Zhao et al., 2017), or non-local (Wang et al., 2017).

Contextual information means the relationship between this pixel and the surrounding pixels and global information is regarded as contextual information. **The reasons why segmentation needs contextual module.** When solving semantic segmentation tasks if each pixel considers only its deep features, such as texture and color, it will be difficult to classify into the correct category (*i.e.*the



(a) Image (b) Groud Truth (c) DeepLabv3+ (d) Ours

Figure 8: Qualitative Visualization on the validation set of ADE20K with ResNet-50c as backbone. All the models here are trained under the same setting.

deep features of leaves and green grass will be very similar). At the same time, as stated in the work, semantic segmentation believes that each pixel is not I.I.D., but is related to the surrounding pixels. The correlation information can better help the segmentation task (*i.e.*the leaves are surrounded by branches, and the grass is likely to be surrounded by roads).

A.8 Ablation of different experts

To measure the goodness and distinctiveness, We adopt the mAcc and per-category bias in Table 9 as the metrics to describe each expert. It seems to demonstrate that the K experts learned at Stage 1 are good and distinctive from each other, rather than simply boosting confidence through multi-expert training.

Furthermore, according to the well-known bias-variance decomposition (Wang et al., 2020b), per-category bias denotes:

$$\text{Error}(x;h) = E\left[(h(x;D) - Y)^2\right] = \text{Bias}(x;h) + \text{Variance}(x;h) + \text{ irreducible error } (x), \quad (20)$$

Table 9: Ablation of each expert with the mean accuracy and per-category bias on Cityscapes with Deeplabv3+ and ResNet-50c. The higher mean accuracy and lower bias are better.

	Many		Medium		Few	
	mAcc	bias	mAcc	bias	mAcc	bias
Overall	0.96	0.10	0.88	0.22	0.81	0.30
Expert 1	0.97	0.09	0.86	0.24	0.75	0.36
Expert 2	-	-	0.95	0.09	0.86	0.19
Expert 3	-	-	-	-	0.91	0.13

A.9 Long-tailed semantic segmentation

To the best of our knowledge, there is a contemporaneous (Cui et al., 2022) work with our paper. Although, we almost simultaneously focus on the long-tailed distribution as an important reason for constraining semantic segmentation performance, there are some differences between our work and Region rebalance (Cui et al., 2022): **Perspective difference:** Region rebalance was concerned about solving the problem of category rebalance, while our work was more focused on improving the recognition of tail and body categories and placed special emphasis on the significance of segmenting the body, and tail categories. **Methods difference:** Region rebalance relieved the categories imbalance with an auxiliary region classification branch by adjusting segmentation boundaries however motivated by the ensembling and grouping methods, we proposed MEDOE framework to encourage different experts to learn more balanced distribution in the feature space, and finally adjust classification boundaries. Compared to Region Rebalance, the motivation of our work was completely different and provided different research directions. **Interpretations of mIoU and mAcc difference:** Region rebalance only explained the cause of mIoU with previous long-tailed methods. Different from Region rebalance, we took a further step and explored the significance of mAcc in segmenting the body and tail categories.

A.10 Significance of body and tail categories in real-world scenarios

We believe in a large number of real-world scenarios, it is more important to be able to identify body and tail categories than accurately segment the edge pixels of head categories. (*i.e.*In the automatic driving scenario, we need to segment some tail categories objects that appear on the driving path, such as "poles" or "fire hydrants", to avoid traffic accidents. Segmenting small lesions on medical images can help doctors detect underlying diseases). Generally, the benefits of this segmentation are far greater than the decrease of certain head categories edges.

A.11 Comparison with traditional multi-experts methods

We compared the differences from existing multi-expert methods, there is three main difference: **Model architecture:** The pipeline of traditional multi-expert methods contain a backbone and multi-classifiers to adjust classifier boundaries and finally ensemble the outputs. 1)We pioneered the **combination of contextual modules and classifiers** to become experts and learn more balanced distribution in the feature space, and finally adjust classification boundaries. 2)Then we provided each expert with soft weight based on the final contextual information and classification results through a learning mechanism to ensemble the outputs. **Training strategies:** Traditional methods often focus on constraining dominating categories and ignoring the confusing categories. Differing from them, we proposed the expert-specific pixel-masking strategy and diverse data distribution-aware loss function to ensure our model architecture focuses on the confusing categories and has better performance. **Training step:** Advanced multi-expert methods in long-tailed classification, such as BBN (Zhou et al., 2020), RIDE (Wang et al., 2020b), LFME (Xiang et al., 2020). They all take multiple steps to train, mainly including 1. training backbone, 2. training classifiers, and 3. Distillation learning (optional). However, our method can update the backbone parameters while training the head expert, so it is one-step training.