CAR: Class-Aware Regularizations for Semantic Segmentation

Ye Huang[1], Di Kang[2], Liang Chen[3], Xuefei Zhe[2], Wenjing Jia[1], Linchao Bao[2], and Xiangjian He[4⋆]

[1] University of Technology Sydney, Australia
[2] Tencent AI Lab
[3] Fujian Normal University
[4] University of Nottingham Ningbo China

Abstract. Recent segmentation methods, such as OCR and CPNet, utilizing "class level" information in addition to pixel features, have achieved notable success for boosting the accuracy of existing network modules. However, the extracted class-level information was simply concatenated to pixel features, without explicitly being exploited for better pixel representation learning. Moreover, these approaches learn soft class centers based on coarse mask prediction, which is prone to error accumulation. In this paper, aiming to use class level information more effectively, we propose a universal Class-Aware Regularization (CAR) approach to optimize the intra-class variance and inter-class distance during feature learning, motivated by the fact that humans can recognize an object by itself no matter which other objects it appears with. Three novel loss functions are proposed. The first loss function encourages more compact class representations within each class, the second directly maximizes the distance between different class centers, and the third further pushes the distance between inter-class centers and pixels. Furthermore, the class center in our approach is directly generated from ground truth instead of from the error-prone coarse prediction. Our method can be easily applied to most existing segmentation models during training, including OCR and CPNet, and can largely improve their accuracy at no additional inference overhead. Extensive experiments and ablation studies conducted on multiple benchmark datasets demonstrate that the proposed CAR can boost the accuracy of all baseline models by up to 2.23% mIOU with superior generalization ability. The complete code is available at https://github.com/edwardyehuang/CAR.

Keywords: Class-aware regularizations, semantic segmentation

1 Introduction

Semantic segmentation, which assigns a class label for each pixel in an image, is a fundamental task in computer vision. It has been widely used in many real-world scenarios that require the results of scene parsing for further processing,

⋆ Corresponding author

e.g., image editing, autopilot, etc. It also benefits many other computer vision tasks such as object detection and depth estimation.

After the early work FCN [15] which used fully convolutional networks to make the dense per-pixel segmentation task more efficient, many works [34,2] have been proposed which have greatly advanced the segmentation accuracy on various benchmark datasets. Among these methods, many of them have focused on better fusing spatial domain context information to obtain more powerful feature representations (termed *pixel features* in this work) for the final per-pixel classification. For example, VGG [20] utilized large square context information by successfully training a very deep network, and DeepLab [2] and PSPNet [34] utilized multi-scale features with the ASPP and PPM modules.

Recently, methods based on dot-product self-attention (SA) have become very popular since they can easily capture the long-range relationship between pixels [25,7,30,33,11,35,6,19,21]. SA aggregates information dynamically (by different attention maps for different inputs) and selectively (using weighted averaging spatial features according to their similarity scores). Using multi-scale and self-attention techniques during spatial information aggregation has worked very well (*e.g.*, 80% mIOU on Cityscapes [16] (single-scale w/o flipping)).

As complements to the above methods, many recent works have proposed various modules to utilize class-level contextual information. The class-level information is often represented by the class center/context prior which are the mean features of each class in the images. OCR [29] and ACFNet [31] extract "soft" class centers according to the predicted coarse segmentation mask by using the weighted sum. CPNet [28] proposed a context prior map/affinity map, which indicates if two spatial locations belong to the same class, and used this predicted context prior map for feature aggregation. However, they [29,31,28] simply concatenated these class-level features with the original pixel features for the final classification.

In this paper, we also focus on utilizing class level information. Instead of focusing on how to better extract class-level features like the existing methods [29,31,28], we use the simple, but accurate, average feature according to the GT mask, and focus on maximizing the inter-class distance during feature learning. This is because it mirrors how humans can robustly recognize an object by itself no matter what other objects it appears with.

Learning more separable features makes the features of a class less dependent upon other classes, resulting in improved generalization ability, especially when the training set contains only limited and biased class combinations (*e.g.*, cows and grass, boats and beach). Fig. 1 illustrates an example of such a problem, where the classification of dog and sheep depends on the classification of grass class, and has been mis-classified as cow. In comparison, networks trained with our proposed CAR successfully generalize to these unseen class combinations.

To better achieve this goal, we propose CAR, a class-aware regularizations module, that optimizes the class center (intra-class) and inter-class dependencies during feature learning. Three loss functions are devised: the first encourages more compact class representations within each class, and the other two di-

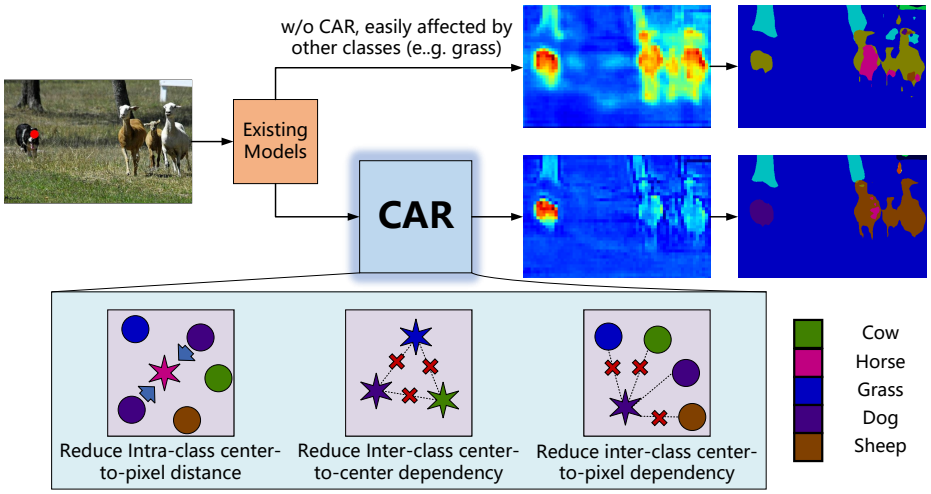


Fig. 1: The concept of the proposed CAR. Our CAR optimizes existing models with three regularization targets: 1) reducing pixels' intra-class distance, 2) reducing inter-class center-to-center dependency, and 3) reducing pixels' inter-class dependency. As highlighted in this example (indicated with a red dot in the image), with our CAR, the grass class does not affect the classification of dog/sheep as much as before, and hence successfully avoids previous (w/o CAR) mis-classification.

rectly maximize the distance between different classes. Specifically, an intra-class center-to-pixel loss (termed as "intra-c2p", Eq. (3)) is first devised to produce more compact representation within a class by minimizing the distance between all pixels and their class center. In our work, a class center is calculated as the averaged feature of all pixels belonging to the same class according to the GT mask. More compact intra-class representations leave a relatively large margin between classes, thus contributing to more separable representations. Then, an inter-class center-to-center loss ("inter-c2c", Eq. (6)) is devised to maximize the distance between any two different class centers. This inter-class center-to-center loss alone does not necessarily produce separable representations for every individual pixels. Therefore, a third inter-class center-to-pixel loss ("inter-c2p", Eq. (13)) is proposed to enlarge the distance between every class center and all pixels that do not belong to the class.

In summary, our contributions in this work are:

1. We propose a universal class-aware regularization module that can be integrated into various segmentation models to largely improve the accuracy.
2. We devise three novel regularization terms to achieve more separable and less class-dependent feature representations by minimizing the intra-class variance and maximizing the inter-class distance.
3. We calculate the class centers directly from ground truth during training, thus avoiding the error accumulation issue of the existing methods and introducing no computational overhead during inference.

4. We provide image-level feature-similarity heatmaps to visualize the learned inter-class features with our CAR are indeed less related to each other.

2 Related Work

Self-Attention. Dot-product self-attention proposed in [25,22] has been widely used in semantic segmentation [7,30,33,35]. Specifically, self-attention determines the similarity between a pixel with every other pixel in the feature map by calculating their dot product, followed by softmax normalization. With this attention map, the feature representation of a given pixel is enhanced by aggregating features from the whole feature map weighted by the aforementioned attention value, thus easily taking long-range relationship into consideration and yielding boosted performance. In self-attention, in order to achieve correct pixel classification, the representation of pixels belonging to the same class should be similar to gain greater weights in the final representation augmentation.

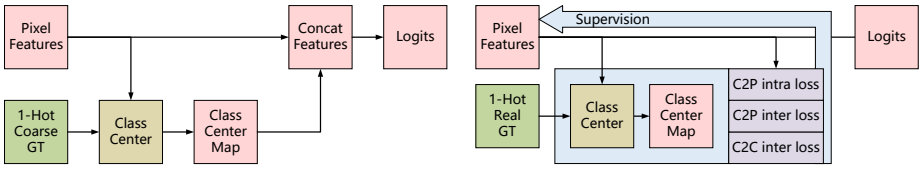
Class Center. In 2019 [31,29], the concept of *class center* was introduced to describe the overall representation of each class from the categorical context perspective. In these approaches, the center representation of each class was determined by calculating the dot product of the feature map and the coarse prediction (*i.e.*, weighted average) from an auxiliary task branch, supervised by the ground truth [34]. After that, those intra-class centers are assigned to the corresponding pixels on feature map. Furthermore, in 2020 [28], a learnable kernel and one-hot ground truth were used to separate the intra-class center from inter-class center, and then concatenated with the original feature representation.

All of these works [29,31,28] have focused on extracting the intra (inter) class centers, but they then simply concatenated the resultant class centers with the original pixel representations to perform the final logits. We argue that the categorical context information can be utilized in a more effective way so as to reduce the inter-class dependency.

To this end, we propose a CAR approach, where the extracted class center is used to directly regularize the feature extraction process so as to boost the differentiability of the learned feature representations (see Fig. 1) and reduce their dependency on other classes. Fig. 2 contrasts the two different designs. More details of the proposed CAR are provided in Sect. 3.

Inter-Class Reasoning. Recently, [5,13] studied the class dependency as a dataset prior and demonstrated that inter-class reasoning could improve the classification performance. For example, a car usually does not appear in the sky, and therefore the classification of sky can help reduce the chance of misclassifying an object in the sky as a car. However, due to the limited training data, such class-dependency prior may also contain bias, especially when the desired class relation rarely appears in the training set.

Fig. 1 shows such an example. In the training set, cow and grass are dependent on each other. However, as shown in this example, when there is a dog or sheep standing on the grass, the class dependency learned from the limited training data may result in errors and predict the target into a class that appears



(a) Design of OCR, ACFNet and CPNet (b) Our CAR

Fig. 2: The difference between the proposed CAR and previous methods that use class-level information. Previous models focus on extracting class center while using simple concatenation of the original pixel feature and the class/context feature for later classification. In contrast, our CAR uses direct supervision related to class center as regularization during training, resulting in small intra-class variance and low inter-class dependency. See Fig. 1 and Sec. 3 for details.

more often in the training data, *i.e.*, cow in this case. In our CAR, we design inter-class and intra-class loss functions to reduce such inter-class dependency and achieve more robust segmentation results.

3 Methodology

3.1 Extracting Class Centers from Ground Truth

Denote a feature map and its corresponding resized one-hot encoded ground-truth mask as $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$[5] and $\mathbf{Y} \in \mathbb{R}^{H\times W\times N_{\text{class}}}$, respectively. We first get the spatially flattened class mask $\mathbf{Y}_{\text{flat}} \in \mathbb{R}^{HW\times N_{\text{class}}}$ and flattened feature map $\mathbf{X}_{\text{flat}} \in \mathbb{R}^{HW\times C}$. Then, the class center[6], which is the average features of all pixel features of a class, can be calculated by:

$$\boldsymbol{\mu}_{image} = \frac{\mathbf{Y}_{\text{flat}}^{T} \cdot \mathbf{X}_{\text{flat}}}{\mathbf{N}_{\text{non-zero}}} \in \mathbb{R}^{N_{\text{class}}\times C}, \tag{1}$$

where $\mathbf{N}_{\text{non-zero}}$ denotes the number of non-zero values in the corresponding map of the ground-truth mask $\mathbf{Y}$. In our experiments, to alleviate the negative impact of noisy images, we calculate the class centers using all the training images in a batch, and denote them as $\boldsymbol{\mu}_{\text{batch}}$[7].

3.2 Reducing Intra-Class Feature Variance

3.2.1 Motivation. More compact intra-class representation can lead to a relatively larger margin between classes, and therefore result in more separable features. In order to reduce the intra-class feature variance, existing works [25,7,35,28,11,30] usually use self-attention to calculate the dot-product

[5] H, W and C denote images' height and width, and number of channels, respectively.
[6] It is termed as *class center* in [31] and *object region representations* in [29].
[7] We use $\boldsymbol{\mu}$ and omit the subscript *batch* for clarity.

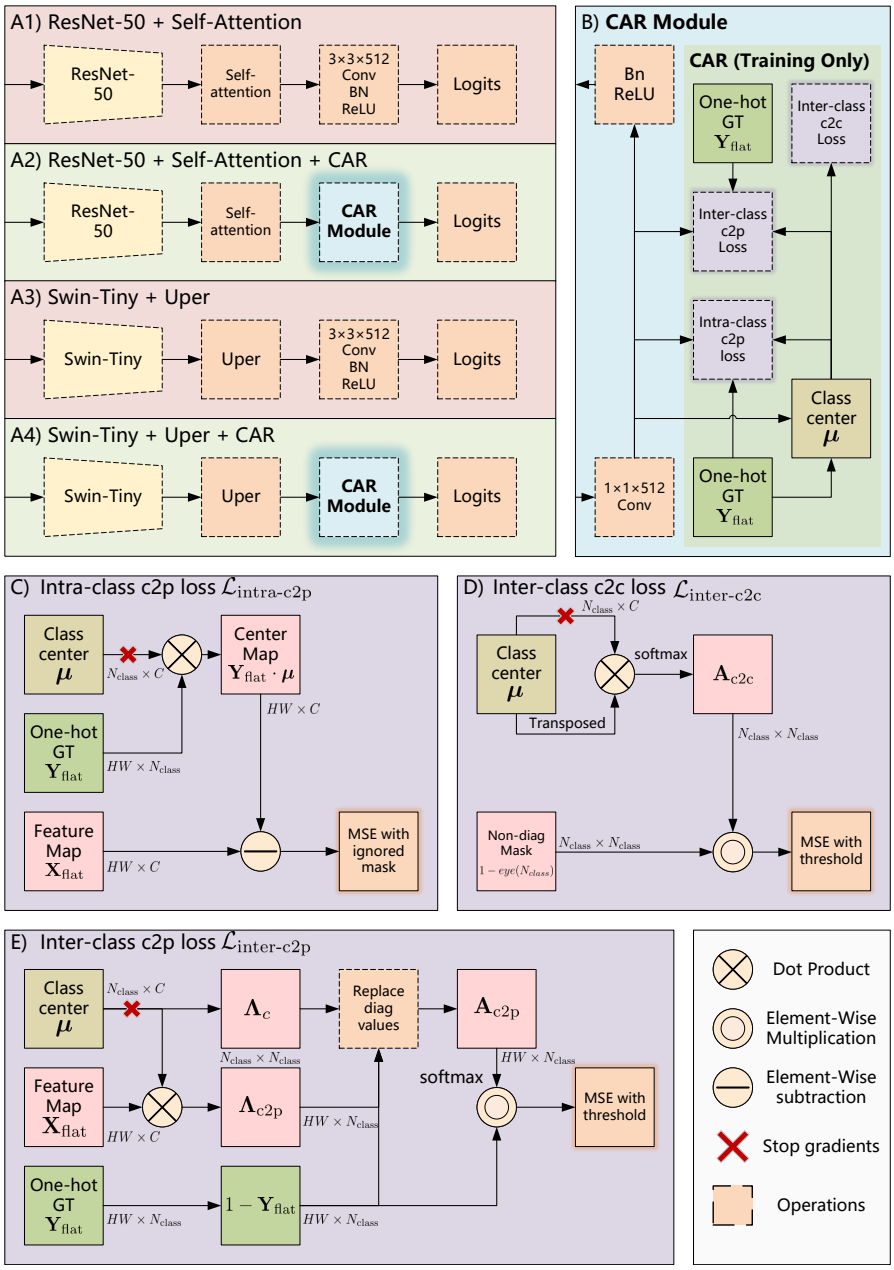


Fig. 3: **The proposed CAR approach.** CAR can be inserted into various segmentation models, right before the logit prediction module (A1-A4). CAR contains three regularization terms, including (C) intra-class center-to-center loss $\mathcal{L}_{\text{intra-c2p}}$ (Sec. 3.2.2), (D) inter-class center-to-center loss $\mathcal{L}_{\text{inter-c2c}}$ (Sec. 3.3.2), and (E) inter-class center-to-pixel loss $\mathcal{L}_{\text{inter-c2p}}$ (Sec. 3.3.3).

similarity in spatial space to encourage similar pixels to have a compact distance implicitly. For example, the self-attention in [25] implicitly pushed the feature representation of pixels belonging to the same class to be more similar to each other than those of pixels belonging to other classes. In our work, we devise a simple *intra-class center-to-pixel loss* to guide the training, which can achieve this goal very effectively and produce improved accuracy.

3.2.2 Intra-class Center-to-pixel Loss. We define a simple but effective intra-class center-to-pixel loss to suppress the intra-class feature variance by penalizing large distance between a pixel feature and its class center. The Intra-class Center-to-pixel Loss $\mathcal{L}_{\text{intra-c2p}}$ is defined by:

$$\mathcal{L}_{\text{intra-c2p}} = f_{\text{mse}}(\mathcal{D}_{\text{intra-c2p}}), \tag{2}$$

where

$$\mathcal{D}_{\text{intra-c2p}} = (1-\sigma)|\mathbf{Y}_{\text{flat}} \cdot \boldsymbol{\mu} - \mathbf{X}_{\text{flat}}|. \tag{3}$$

In Eq. (3), σ is a spatial mask indicating pixels being ignored (*i.e.*, ignore label), $\mathbf{Y}_{\text{flat}} \cdot \boldsymbol{\mu}$ distributes the class centers $\boldsymbol{\mu}$ to the corresponding positions in each image. Thus, our intra-class loss $\mathcal{L}_{\text{intra-c2p}}$ will push the pixel representations to their corresponding class center, using mean squared error (MSE) in Eq. (3).

3.3 Maximizing Inter-class Separation

3.3.1 Motivation. Humans can robustly recognize an object by itself regardless which other objects it appears with. Conversely, if a classifier *heavily* relies on the information from other classes to determine the classification result, it will easily produce wrong classification results when a rather rare class combination appears during inference. Maximizing inter-class separation, or in another words, reducing the inter-class dependency, can therefore help the network generalize better, especially when the training set is small or is biased. As shown in Fig. 1, the dog and sheep are mis-classified as the cow because cow and grass appear together more often in the training set. To improve the robustness of the model, we propose to reduce this inter-class dependency. To this end, the following two loss functions are defined.

3.3.2 Inter-class Center-to-center Loss. The first loss function is to maximize the distance between any two different class centers. Inspired by the center loss used in face recognition [26], we propose to reduce the similarity between class centers $\boldsymbol{\mu}$, which are the averaged features of each class calculated according to the GT mask. The *inter-class relation* is defined by the dot-product similarity [22] between any two classes as:

$$\mathbf{A}_{\text{c2c}} = \text{softmax}(\frac{\boldsymbol{\mu}^T \cdot \boldsymbol{\mu}}{\sqrt{C}}), \quad \mathbf{A}_{\text{c2c}} \in \mathbb{R}^{N_{class} \times N_{class}}. \tag{4}$$

Moreover, since we only need to constrain the inter-class distance, only the non-diagonal elements are retained for the later loss calculation as:

$$\mathbf{D}_{\text{inter-c2c}} = \Big(1 - eye(N_{class})\Big)\mathbf{A}_{\text{c2c}}. \tag{5}$$

We only penalize larger similarity values between any two different classes than a pre-defined threshold $\frac{\epsilon_0}{N_{class}-1}$, *i.e.*,

$$\mathcal{D}_{\text{inter-c2c}} = f_{\text{sum}}\Big(\max(\mathbf{D}_{\text{inter-c2c}} - \frac{\epsilon_0}{N_{class}-1}, 0)\Big). \tag{6}$$

Thus, the Inter-class Center-to-center Loss $\mathcal{L}_{\text{inter-c2c}}$ is defined by:

$$\mathcal{L}_{\text{inter-c2c}} = f_{\text{mse}}(\mathcal{D}_{\text{inter-c2c}}). \tag{7}$$

Here, a small margin is used in consideration of the feature space size and the mislabeled ground truth.

3.3.3 Inter-class Center-to-pixel Loss. Maximizing only the distances between class centers does not necessarily result in separable representation for every individual pixels. We further maximize the distance between a class center and any pixel that does not belong to this class. More concretely, we first compute the center-to-pixel dot product as:

$$\mathbf{\Lambda}_{\text{c2p}} = \boldsymbol{\mu}^T \cdot \mathbf{X}_{\text{flat}}, \quad \mathbf{\Lambda}_{\text{c2p}} \in \mathbb{R}^{HW \times N_{\text{class}}}. \tag{8}$$

Ideally, with the previous loss $\mathcal{L}_{\text{inter-c2c}}$, the features of all pixels belonging to the same class should be equal to that of the class center. Therefore, we replace the intra-class dot product with its ideal value, namely using the class center $\boldsymbol{\mu}$ for calculating the intra-class dot product as:

$$\mathbf{\Lambda}_c = diag(\boldsymbol{\mu}^T \cdot \boldsymbol{\mu}), \tag{9}$$

and the replacement effect is achieved by using masks as:

$$\mathbf{\Lambda}' = \mathbf{\Lambda}_{\text{c2p}}(1 - \mathbf{Y}_{\text{flat}}) + \mathbf{\Lambda}_c. \tag{10}$$

This updated dot product $\mathbf{\Lambda}'$ is then used to calculate similarity across class axis with a softmax as:

$$\mathbf{A}_{\text{c2p}} = \text{softmax}(\mathbf{\Lambda}'), \quad \mathbf{A}_{\text{c2p}} \in \mathbb{R}^{HW \times N_{\text{class}}}. \tag{11}$$

Similar to the calculation of $\mathcal{L}_{\text{inter-c2c}}$ in the previous subsection, we have

$$\mathbf{D}_{\text{inter-c2p}} = (1 - \mathbf{Y}_{\text{flat}})\mathbf{A}_{\text{c2p}}, \tag{12}$$

$$\mathcal{D}_{\text{inter-c2p}} = f_{\text{sum}}\Big(\max(\mathbf{D}_{\text{inter-c2p}} - \frac{\epsilon_1}{N_{\text{class}}-1}, 0)\Big). \tag{13}$$

Thus, the Inter-class Center-to-pixel Loss $\mathcal{L}_{\text{inter-c2p}}$ is defined by:

$$\mathcal{L}_{\text{inter-c2p}} = f_{\text{mse}}(\mathcal{D}_{\text{inter-c2p}}). \tag{14}$$

3.4 Differences with OCR, ACFNet, CPNet, and CIPC

Methods that are closely related to ours are OCR [29], ACFNet [31] and CPNet [28], which all focus on better utilizing class-level features and differ on how to extract the class centers and context features. However, they all use a **simple concatenation** to fuse the original pixel feature and the complementary context feature. For example, OCR and ACFNet first produce a coarse segmentation, which is supervised by the GT mask with a categorical cross-entropy loss, and then use this predicted coarse mask to generate the (soft) class centers by weighted summing all the pixel features. OCR then aggregates these class centers according to their similarity to the original pixel feature termed as "pixel-region relation", resulting in a "contextual feature". Slightly differently from OCR, ACFNet directly uses the probability (from the predicted coarse mask) to aggregate class center, obtaining a similar context feature termed as "attentional class feature". CPNet defines an affinity map, which is a binary map indicating if two spatial locations belong to the same class. Then, they use a sub-network to predict their ideal affinity map and use the soft version affinity map termed as "Context Prior Map" for feature aggregation, obtaining a class feature (center) and a context feature. Note that CPNet concatenates class feature, which is the updated pixel feature, and the context feature.

We also propose to utilize class-level contextual features. Instead of extracting and fusing pixel features with sub-networks, we propose three loss functions to directly regularize training and encourage the learned features to maintain certain desired properties. The approach is simple but more effective thanks to the direct supervision (validated in Tab. 2). Moreover, our class center estimate is more accurate because we use the GT mask. This strategy largely reduces the complexity of the network and introduces no computational overhead during inference. Furthermore, it is compatible with all existing methods, including OCR, ACFNet and CPNet, demonstrating great generalization capability.

We also notice that Cross-Image Pixel Contrast (CIPC) [24] shares a similar high-level goal as our CAR, *i.e.*, learning more similar representations for pixels belonging to the same class than to a different class. However, the approaches of achieving this goal are very different. CIPC is motivated by contrastive learning while our CAR is motivated by the compositionality of the scene, for better generalization in the cases of rare class combinations. Therefore, CIPC adopts the *one-vs-rest* style InfoNCE loss, including the typical pixel-to-pixel loss and a special pixel-to-center loss. In contrast, **(1)** we propose an additional *center-to-center* loss to regularize the inter-class dependency explicitly and effectively (see Table 1); **(2)** we use *one-vs-one* style inter-class losses while CIPC uses *one-vs-rest* style NCE loss. Compared to our *one-vs-one* loss, using *one-vs-rest* loss for training does not necessarily result in small inter-class similarity between the current class and every individual "other" classes and may increase the inter-class similarity among those "other" classes. **(3)** we leave margins to prevent CAR regularizations, which is not the primary task of pixel classification, from dominating the learning process.

Table 1: Ablation studies of adding CAR to different methods on Pascal Context dataset. All results are obtained with single scale test without flip. "A" means replacing the 3×3 conv with 1×1 conv (detailed in Sec. 4.2.1). CAR improves the performance of different types of backbones (CNN & Transformer) and head blocks (SA & Uper), showing the generalizability of the proposed CAR.

	Methods	$\mathcal{L}_{\text{intra-c2p}}$	$\mathcal{L}_{\text{inter-c2c}}$	$\mathcal{L}_{\text{inter-c2p}}$	A	mIOU (%)
R1	ResNet-50 + Self-Attention [25]	-	-			48.32
R2					✓	48.56
R3	+ CAR	✓				49.17
R4		✓	✓			49.79
R5		✓	✓	✓		50.01
R6		✓			✓	49.62
R7		✓	✓		✓	50.00
R8		✓	✓	✓	✓	**50.50**
S1	Swin-Tiny + UperNet [27]	-	-			49.62
S2					✓	49.82
S3	+ CAR	✓				49.01
S4		✓	✓			50.63
S5		✓	✓	✓		50.26
S6		✓			✓	49.62
S7		✓	✓		✓	50.58
S8		✓	✓	✓	✓	**50.78**

4 Experiments

4.1 Implementation

Training Settings. For both CAR and baselines, we apply the settings common to most works [32,33,11,10,35], including SyncBatchNorm, batch size = 16, weight decay (0.001), 0.01 initial LR, and poly learning decay with SGD during training. In addition, for the CNN backbones (*e.g.*, ResNet), we set *output stride* = 8 (see [3]). Training iteration is set to 30k iterations unless otherwise specified. For the thresholds in Eq. 6 and Eq. 13, we set $\epsilon_0 = 0.5$ and $\epsilon_1 = 0.25$.

Determinism and Reproducibility Our implementations are based on the latest NVIDIA deterministic framework (2022), which means exactly the same results can be always reproduced with the same hardware and same training settings (including random seed). To demonstrate the effectiveness of our CAR with equal comparisons, we reproduced all the baselines that we compare, all conducted with exactly the same settings unless otherwise specified.

4.2 Experiments on Pascal Context

The Pascal Context [18] dataset is split into 4,998/5,105 for training/test set. We use its 59 semantic classes following the common practice [29,33]. Unless otherwise specified, both baselines and CAR are trained on the training set with 30k iterations. The ablation studies are presented in Sect. 4.2.1.

4.2.1 Ablation Studies on Pascal Context

CAR on ResNet-50 + Self-Attention. We firstly test the CAR with "ResNet-50 + Self-Attention" (w/o image-level block in [33]) to verify the effectiveness of the proposed loss functions, *i.e.*, $\mathcal{L}_{\text{intra-c2p}}$, $\mathcal{L}_{\text{inter-c2c}}$, and $\mathcal{L}_{\text{inter-c2p}}$.

As shown in Tab. 1, using $\mathcal{L}_{\text{intra-c2p}}$ directly improves 1.30 mIOU (48.32 vs 49.62); Introducing $\mathcal{L}_{\text{inter-c2c}}$ and $\mathcal{L}_{\text{inter-c2p}}$ further improves 0.38 mIOU and 0.50 mIOU; Finally, with all three loss functions, the proposed CAR improves 2.18 mIOU from the regular ResNet-50 + Self-attention (48.32 vs 50.50).

CAR on Swin-Tiny + Uper. "Swin-Tiny + Uper" is a totally different architecture from "ResNet-50 + Self-Attention [25]". Swin [14] is a recent Transformer-based backbone network. Uper [27] is based on the pyramid pooling modules (PPM) [34] and FPN [12], focusing on extracting multi-scale context information. Similarly, as shown in Tab. 1, after adding CAR, the performance of Swin-Tiny + Uper also increases by 1.16, which shows our CAR can generalize to different architectures well.

The Devil is In the Architecture's Detail. We find it important to replace the leading 3×3 conv (in the original method) with 1×1 conv (Fig. 3B). For example, $\mathcal{L}_{\text{intra-c2p}}$ and $\mathcal{L}_{\text{inter-c2p}}$ did not improve the performance in Swin-Tiny + Uper (Row S3 vs S1, and S5 vs S4 in Tab. 1). A possible reason is that the network is trained to maximize the separation between different classes. However, if the two pixels lie on different sides of the segmentation boundary, a 3×3 conv will merge the pixel representations from different classes, making the proposed losses harder to optimize.

To keep simplicity and maximize generalization, we use the same network configurations in our **all** experiments. However, performance may be further improved with some minor dedicated modifications for each baseline when deploying our CAR. For example, decreasing the filter number to 256 for the last conv layer of ResNet-50 + Self-Attention + CAR results in a further improvement to 51.00 mIOU (from 50.50). Replacing the conv layer after PPM (inside Uper block, A3 in Fig. 3) from 3×3 to 1×1 in Swin-Tiny + UperNet boosts Swin (tiny & large) + UperNet + CAR by an extra 0.5-1.0 mIOU. We intentionally did *not* exhaustively search these variants and *not* report these results in any table since they did not generalize.

CAR on Different Baselines. After we have verified the effectiveness of each part of the proposed CAR, we then tested CAR on multiple well-known baselines. All of the baselines were reproduced under similar conditions (see Sect. 4.1). Experimental results shown in Tab. 2 demonstrate the generalizability of our CAR on different backbones and methods.

Visualization of Class Dependency Maps. In Fig. 4, we present the class dependency maps calculated on the complete Pascal Context *test* set, where every pixel stores the dot-product similarities between every two class centers. The maps indicate the inter-class dependency obtained with the standard ResNet-50 + Self-Attention and Swin-Tiny + UperNet, and the effect of applying our CAR.

Table 2: Ablation studies of adding CAR to different baselines on Pascal Context [18] and COCOStuff-10K [1]. We deterministically reproduced all the baselines with the same settings. All results are single-scale without flipping. CAR works very well in most existing methods. **§** means reducing the class-level threshold to 0.25 from 0.5. We found it is sensitive for some model variants to handle a large number of class. Affinity loss [28] and Auxiliary loss [34] are applied on CPNet and OCR, respectively, since they highly rely on those losses.

Methods	Backbone	Reference	mIOU(%)	
			Pascal Context	COCO-Stuff10K
FCN [15]	ResNet-50 [8]	CVPR2015	47.72	34.10
FCN + CAR	ResNet-50 [8]		48.40(+0.68)	34.91(+0.81)§
FCN [15]	ResNet-101 [8]	CVPR2015	50.93	35.93
FCN + CAR	ResNet-101 [8]		51.39(+0.49)	36.88(+0.95)§
DeepLabV3 [4]	ResNet-50 [8]	ECCV2018	48.59	34.96
DeepLabV3 + CAR	ResNet-50 [8]		49.53(+0.94)	35.13(+0.17)
DeepLabV3 [4]	ResNet-101 [8]	ECCV2018	51.69	36.92
DeepLabV3 + CAR	ResNet-101 [8]		52.58(+0.89)	37.39(+0.47)
Self-Attention [25]	ResNet-50 [8]	CVPR2018	48.32	34.35
Self-Attention + CAR	ResNet-50 [8]		50.50(+2.18)	36.58(+2.23)§
Self-Attention [25]	ResNet-101 [8]	CVPR2018	51.59	36.53
Self-Attention + CAR	ResNet-101 [8]		52.49(+0.9)	38.15(+1.62)
CCNet [10]	ResNet-50 [8]	ICCV2019	49.15	35.10
CCNet + CAR	ResNet-50 [8]		49.56(+0.41)	36.39(+1.29)
CCNet [10]	ResNet-101 [8]	ICCV2019	51.41	36.88
CCNet + CAR	ResNet-101 [8]		51.97(+0.56)	37.56(+0.68)
DANet [7]	ResNet-101 [8]	CVPR2019	51.45	35.80
DANet + CAR	ResNet-101 [8]		52.57(+1.12)	37.47(+1.67)
CPNet [28]	ResNet-101 [8]	CVPR2020	51.29	36.92
CPNet + CAR	ResNet-101 [8]		51.98(+0.69)	37.12(+0.20)§
OCR [29]	HRNet-W48 [23]	ECCV2020	54.37	38.22
OCR + CAR	HRNet-W48 [23]		54.99(+0.62)	39.53(+1.31)
UperNet [27]	Swin-Tiny [14]	ICCV2021	49.62	36.07
UperNet + CAR	Swin-Tiny [14]		50.78(+1.16)	36.63(+0.56)§
UperNet [27]	Swin-Large [14]	ICCV2021	57.48	44.25
UperNet + CAR	Swin-Large [14]		58.97(+1.49)	44.88(+0.63)
CAA [9]	EfficientNet-B5 [17]	AAAI2022	57.79	43.40
CAA + CAR	EfficientNet-B5 [17]		58.96(+1.17)	43.93(+0.53)

A hotter color means that the class has higher dependency on the corresponding class, and vice versa. According to Fig. 4 a1-a2, we can easily observe that the inter-class dependency has been significantly reduced with CAR on ResNet50 + Self-Attention. Fig. 4 b1-b2 show a similar trend when tested with different backbones and head blocks. This partially explains the reason why baselines with CAR generalize better on rarely seen class combinations (Figs. 1 and 5). Interestingly, we find that the class-dependency issue is more serious in Swin-Tiny + Uper, but our CAR can still reduce its dependency level significantly.

Visualization of Pixel-relation Maps. In Fig. 5, we visualize the pixel-to-pixel relation energy map, based on the dot-product similarity between a red-dot

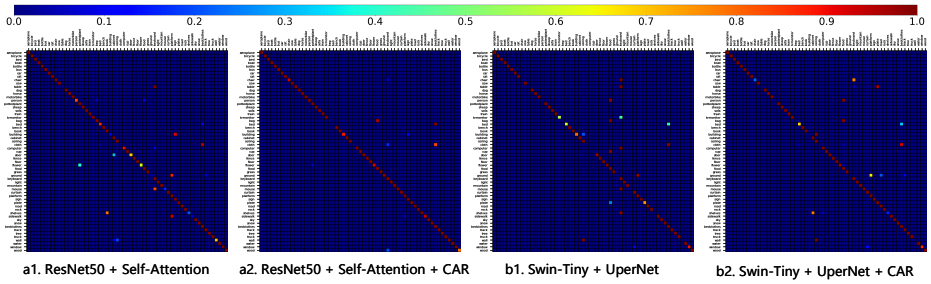


Fig. 4: Class dependency maps generated on Pascal Context test set. One may zoom in to see class names. A hotter color means that the class has higher dependency to the corresponding class, and vice versa. It is obvious that our CAR reduces the inter-class dependency, thus providing better generalizability (see Figs. 1 and 5).

marked pixel and other pixels, as well as the predicted results for different methods, for comparison. Examples are from Pascal Context test set. As we can see, with CAR supervision, the existing models focus better on objects themselves rather than other objects. Therefore, this reduces the possibility of the classification errors because of the class-dependency bias.

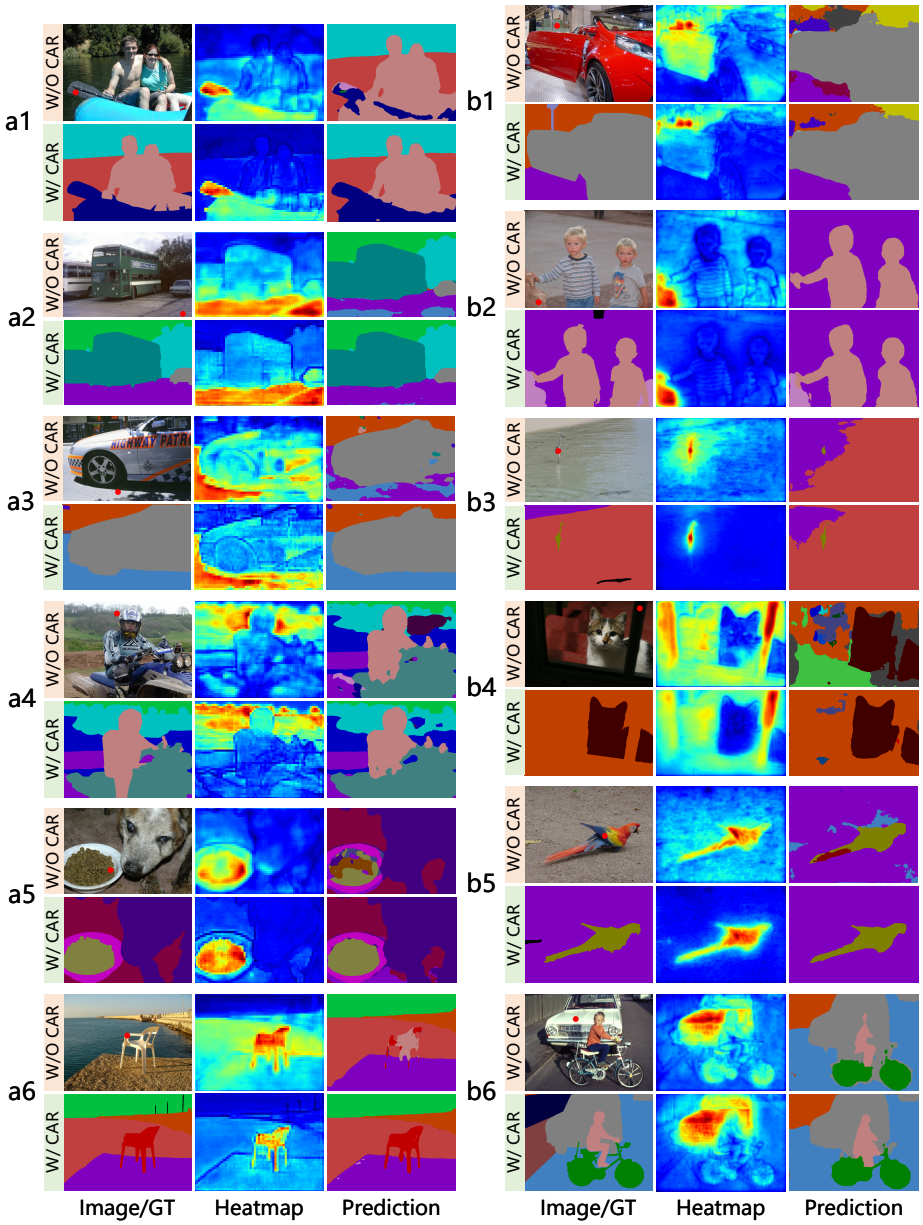
4.3 Experiments on COCOStuff-10K

COCOStuff-10K dataset [1] is widely used for evaluating the robustness of semantic segmentation models [11,29]. The COCOStuff-10k dataset is a very challenging dataset containing 171 labeled classes and 9000/1000 images for training/test. As shown in Tab. 2, all of the tested baselines gain performance boost ranging from 0.17% to 2.23% with our proposed CAR. This demonstrates the generalization ability of our CAR when handling a large number of classes.

5 Conclusions and Future Work

In this paper, we have aimed to make a better use of class level context information. We have proposed a universal class-aware regularizations (CAR) approach to regularize the training process and boost the differentiability of the learned pixel representations. To this end, we have proposed to minimize the intra-class feature variance and maximize the inter-class separation simultaneously. Experiments conducted on benchmark datasets with extensive ablation studies have validated the effectiveness of the proposed CAR approach, which has boosted the existing models' performance by up to 2.18% mIOU on Pascal Context and 2.23% on COCOStuff-10k with no extra inference overhead.

Acknowledgement This research depends on the NVIDIA determinism framework. We appreciate the support from @duncanriach and @reedwm at NVIDIA and TensorFlow team.



(a) ResNet50 + Self-Attention (b) Swin-Tiny + UperNet

Fig. 5: Visualization of the feature similarity between a given pixel (marked with a red dot in the image) and all pixels, as well as the segmentation results on Pascal Context test set. A hotter color denotes larger similarity value. Apparently, our CAR reduces the inter-class dependency and exhibits better generalization ability, where energies are better restrained in the intra-class pixels.

References

1. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: Thing and Stuff Classes in Context. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
3. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation (2017)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision (2018)
5. Choi, S., Kim, J.T., Choo, J.: Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
7. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
9. Huang, Y., Kang, D., Jia, W., He, X., liu, L.: Channelized axial attention - considering channel relation within spatial attention for semantic segmentation. In: AAAI (2022)
10. Huang, Z., Wang, X., Wei, Y., Huang, L., Shi, H., Liu, W., Huang, T.S.: Ccnet: Criss-cross attention for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
11. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: International Conference on Computer Vision (2019)
12. Lin, T.Y., Dollá, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
13. Liu, M., Schonfeld, D., Tang, W.: Exploit visual dependency relations for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)
14. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
16. Marius, C., Mohamed, O., Sebastian, R., Timo, R., Markus, E., Rodrigo, B., Uwe, F., Roth, S., Bernt, S.: The cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
17. Mingxing, T., Quoc, L.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning (2019)

18. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
19. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV (2021)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
21. Sixiao, Z., Jiachen, L., Hengshuang, Z., Xiatian, Z., Zekun, L., Yabiao, W., Yanwei, F., Jianfeng, F., Tao, X., H.S., T.P., Li, Z.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Łukasz Kaiser, Polosukhin, I.: Attention is all you need. In: Conference on Neural Information Processing Systems (2017)
23. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
24. Wang, W., Zhou, T., Yu, F., Dai, J., Konukoglu, E., Van Gool, L.: Exploring cross-image pixel contrast for semantic segmentation. In: ICCV. pp. 7303–7313 (2021)
25. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
26. Wen1, Y., Zhang, K., Li, Z., Qiao, Y.: Discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision (2016)
27. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: European Conference on Computer Vision (2018)
28. Yu, C., Wang, J., Gao, C., Yu, G., Shen, C., Sang, N.: Context prior for scene segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
29. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: European Conference on Computer Vision (2020)
30. Yuan, Y., Huang, L., Guo, J., Zhang, C., Chen, X., Wang, J.: Ocnet: Object context network for scene parsing. International Journal of Computer Vision (2021)
31. Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., Ding, E.: Acfnet: Attentional class feature network for semantic segmentation. In: International Conference on Computer Vision (2019)
32. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
33. Zhang, H., Zhan, H., Wang, C., Xie, J.: Semantic correlation promoted shape-variant context for segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
34. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
35. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: International Conference on Computer Vision (2019)