Beyond Prototypes: Semantic Anchor Regularization for Better Representation Learning

Yanqi Ge[1*], Qiang Nie[2*], Ye Huang[1], Yong Liu[2], Chengjie Wang[2,3], Feng Zheng[4†], Wen Li[1], Lixin Duan[1,5†]

[1] Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China
[2] Tencent Youtu Lab
[3] Shanghai Jiao tong University
[4] Southern University of Science and Technology
[5] Sichuan Provincial People's Hospital
geyanqiqi@gmail.com

Abstract

One of the ultimate goals of representation learning is to achieve compactness within a class and well-separability between classes. Many outstanding metric-based and prototype-based methods following the Expectation-Maximization paradigm, have been proposed for this objective. However, they inevitably introduce biases into the learning process, particularly with long-tail distributed training data. In this paper, we reveal that the class prototype is not necessarily to be derived from training features and propose a novel perspective to use pre-defined class anchors serving as feature centroid to unidirectionally guide feature learning. However, the pre-defined anchors may have a large semantic distance from the pixel features, which prevents them from being directly applied. To address this issue and generate feature centroid independent from feature learning, a simple yet effective Semantic Anchor Regularization (SAR) is proposed. SAR ensures the inter-class separability of semantic anchors in the semantic space by employing a classifier-aware auxiliary cross-entropy loss during training via disentanglement learning. By pulling the learned features to these semantic anchors, several advantages can be attained: 1) the intra-class compactness and naturally inter-class separability, 2) induced bias or errors from feature learning can be avoided, and 3) robustness to the long-tailed problem. The proposed SAR can be used in a plug-and-play manner in the existing models. Extensive experiments demonstrate that the SAR performs better than previous sophisticated prototype-based methods. The implementation is available at *https://github.com/geyanqi/SAR*.

1 Introduction

Classification, either at the image level or at the pixel level (semantic segmentation), is a foundation computer vision task with a wide range of applications, including but not limited to autonomous agent tasks such as scene understanding, augmented reality, and autonomous driving. Many efforts have been made in this problem and great progress has been achieved in recent years, especially after deep learning methods (Perronnin, Sánchez, and Mensink 2010; He et al. 2016; Krizhevsky, Sutskever, and Hinton 2017; Chen, Fan, and Panda 2021; Long, Shelhamer, and Darrell 2015; Chen et al. 2017a; Wang et al. 2020) being introduced. However, no matter what kind of methods are utilized or what kind of network structures are designed, the ultimate goal is to learn representations of data that are compact within a class and separable between classes in the semantic space. To achieve this, many methods have been proposed, such as metric learning and prototype-based learning.

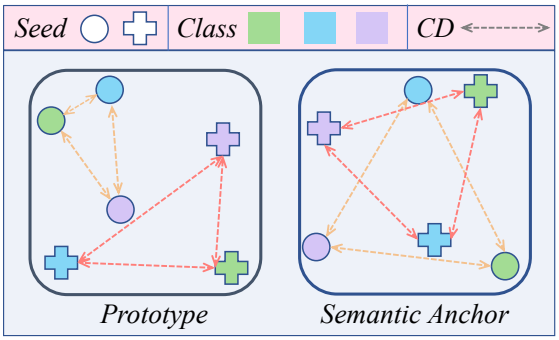

Figure 1: The difference between prototypes and semantic anchors in feature space (UMAP-Based). We train HRNet with two different seeds on Cityscapes to get these prototypes and semantic anchors. Shapes, colors, and CD represent random seeds, classes, and class dependencies, respectively. The generation of semantic anchors is independent of the main task, and it achieves more consistent and weaker inter-class dependencies on imbalanced data.

Metric learning is to pull together the intra-class samples and push away the samples of different categories by designing a distance metric. A lot of distance metrics have been widely utilized and benefit the representation learning, such as the contrastive loss (He et al. 2020; Oord, Li, and Vinyals 2018; Wu et al. 2018; Huang et al. 2019; Wang and Isola 2020; Wang and Liu 2021; Yang et al. 2022) and the triplet loss (Schroff, Kalenichenko, and Philbin 2015; Ge 2018). These losses are utilized to learn effective image representations for downstream tasks by explicitly selecting positive

*These authors contributed equally.
†Corresponding author.

data pairs and negative data pairs. Wang and Isola (2020) revealed that the contrastive representation benefits from the alignment of features of positive pairs and uniformity of the induced feature distribution. However, the contrastive representation relies on the construction of positive and negative sample pairs, which might induce bias in the feature learning process.

Prototype-based deep learning has been attracting increasing interest recently due to its exemplar-driven nature and intuitive interpretation, which also can be deemed as only using one or several hyper-positive samples. By aligning samples with the most similar prototype in the semantic space, prototype-based methods have attained remarkable results in few-shot learning (Wang et al. 2019; Kwon et al. 2021), unsupervised learning (Xu et al. 2020), supervised learning (Zhou et al. 2022; Wang et al. 2021), and domain adaptation (Jiang et al. 2022; Lu et al. 2022), especially for long-tailed problems, *e.g*, semantic segmentation. ProtoAttend (Arik and Pfister 2020) shows that prototype learning is more robust when handling out-of-distribution samples, which should be attributed to the more compact data representation within the class. While CNN tends to learn non-discriminative features with high activations for different classes (Nguyen and Todorovic 2019), i.e., the low inter-class distance. Similarly, learning more separable prototype relationships reduces the interdependence of class features, leading to enhanced generalization capabilities, especially when the training set follows a long-tailed distribution. Recently, RegionContrast and ContrastSeg (Hu, Cui, and Wang 2021; Wang et al. 2021) propose to explore the "global" context of the training set by leveraging contrastive loss between pixel features and prototypes. CAR (Huang et al. 2022) and SASM (Hong et al. 2022) propose directly optimizing inter-class and intra-class prototype relationships by Euclidean distance. ProtoSeg (Zhou et al. 2022) proposes a non-learnable classifier using online clustering to match learned prototypes.

However, the methods mentioned above are all via the Expectation-Maximization paradigm (Moon 1996), which estimates prototype assignments given learned features and updates learned features with updated prototype assignments. Compared to these sophisticated prototype learning methods, one realistic but seldom mentioned fact is that the relative relationships among prototypes undergo an evident drift with distinct random seeds, even though the training set and structure of the network are fixed (see Fig. 1). Especially in long-tailed problems like segmentation, the prototype of the rare class appears a strong bias towards certain classes. This phenomenon demonstrates that the traditional prototype calculations are sub-optimal since they are heavily bound to the feature learning process and distributions of training data, which can potentially result in the learning collapse for tail-end classes.

A potentially better solution could be to directly guide feature learning using well-separated and fixed class anchors. To explore this assumption, we generate three sets of predefined anchors as feature centroid guiding feature learning, by randomly sampling from three distinct sources: standard normal distributions, random orthogonal matrix, and random matrix with a maximum equiangular separability structure (Papyan, Han, and Donoho 2020). Subsequently, we minimize the Euclidean distance between pixel features and their corresponding anchor features to regularize the model. Amazingly, Tab. 7 shows that although the performance of randomly generated anchors is unstable, they can be beneficial for performance sometimes, and are comparable to the performance achieved by sophisticatedly prototype-based methods. In addition, solely controlling the angular structure of these class anchors did not guarantee inter-class separability and a more noticeable performance improvement. We believe this unstable and limited improvement is due to the significant semantic gap between the randomly generated anchors and learned pixel features.

To align the anchor with features in the semantic space and keep the independence of anchor generation from feature learning, we propose a simple yet effective Semantic Anchor Regularization (SAR) for learning intra-class compact and inter-class separable representations. As shown in Fig. 2, instead of collecting prototypes during feature learning process, these pre-defined class anchors $\boldsymbol{A} \in \mathbb{R}^{C \times D}$ for all categories are projected into the semantic space through a lightweight embedding layer and categorized by the classifier of the main network, where C is the total class number and D denotes the semantic dimension of last feature layer before classification. In addition, we apply two key training strategies, loss reweighting, and exponential moving average (EMA) updates, to ensure that semantic anchors obtained during training are independent of the main task. We will detail these in Sec. 3. In addition to being supervised by GT labels, by aligning features in the main network with semantic anchors, several advantages can be achieved: 1) the intra-class compactness and inter-class separability can be intuitively achieved by pulling the feature of each class to the corresponding semantic anchor, 2) induced bias and errors of the learned prototype which is calculated as the feature center can be avoided, 3) less influenced by the number of training samples and robust in long-tailed problem. The main contributions of this paper are summarized as follows:

- We reveal that prototype representations derived from the learned features are sub-optimal and propose a simple yet effective SAR to gain better class representation.
- SAR can be used in a plug-and-play manner in existing models with a little extra training cost (add 0.3 GFLOPs and 1.56M parameters for HRNet) and no testing cost.
- We evaluate the proposed approach on various challenging semantic segmentation benchmarks. Extensive experiments and visualization examples demonstrate the proposed SAR is capable of promoting intra-class compactness and inter-class separability.

2 Related Work

One of the ultimate goals of learning data representation is to have good intra-class compactness and inter-class separability. In the following, we review some related works that pursue this goal in metric learning and prototype-based deep learning.

2.1 Metric Learning

Metric learning is to pull together samples within a class and push away the samples of different categories by designing a distance metric. Among them, the contrastive loss (He et al. 2020; Oord, Li, and Vinyals 2018; Wu et al. 2018; Huang et al. 2019; Wang and Isola 2020; Wang and Liu 2021), the triplet loss (Schroff, Kalenichenko, and Philbin 2015; Ge 2018), and the n-pair loss (Sohn 2016) are the most widely utilized. These losses are utilized to learn effective image representations for downstream tasks by explicitly selecting positive data pairs and negative data pairs. CPC (Oord, Li, and Vinyals 2018) applied contrastive predictive coding to learn representations from widely different data modalities, images, speech, and natural language. MoCo (He et al. 2020) proposed a momentum contrast method for unsupervised visual representation learning, which allows them to build large and consistent dictionaries. Wang and Isola (2020) revealed that the contrastive representation benefits from the alignment of features from positive pairs and uniformity of the induced feature distribution. However, the contrastive representation relies on the construction of positive or negative sample pairs, which might induce bias in this process. DCL (Chuang et al. 2020) proposed a debiased contrastive learning method to reduce false negative samples without human annotations. After all, the ideal unbiased contrastive learning is unachievable in practice since calculating all pairwise comparisons on a large dataset is impossible.

2.2 Prototype-based Deep Learning

Prototype-based learning can be deemed as special metric learning which only considers the hyper-positive samples. Previously, prototype-based learning was combined with nearest neighbors rule (Cover and Hart 1967) for classification tasks. Recently, a lot of work has combined prototype learning with deep neural networks and achieved remarkable results in many areas. ProtoAttend (Arik and Pfister 2020) shows that prototype learning is more robust when handling out-of-distribution samples. DPCL (Kwon et al. 2021) addresses the few-shot semantic segmentation problem by learning more discriminative prototypes that have larger inter-class distance and small intra-class distance in feature space. APN (Xu et al. 2020) utilized an attribute prototype network to transfer knowledge from known to unknown classes. To tackle the bias in calculating prototypes, BiSMAP (Lu et al. 2022) proposed multiple anisotropic prototypes. ProCA (Jiang et al. 2022) proposed a prototypical contrast adaptation method for domain adaptive segmentation, which incorporates more inter-class information into class-wise prototypes. CAR (Huang et al. 2022) proposed optimizing representation distance from inter-class and intra-class representation relationships. ProtoSeg (Zhou et al. 2022) directly selects sub-cluster centers of embedded pixels as prototypes and implements segmentation via nonparametric nearest prototype retrieving. Unlike these previous methods that via EM paradigm to optimize representation relationships, SAR introduces some anchors in the semantic space to serve as feature centroids and employs them to unidirectionally guide feature learning. By generating feature centroids independently of feature learning, SAR is more consistent across the learning process and robust to long-tailed distribution.

3 Method

3.1 Recap of Prototype-based Deep Learning

In the setting of semantic segmentation, each pixel i of an image I has to be assigned to a class $c \in C$. Specifically, let model $S_{\phi,\theta}$ comprises a feature extractor f_ϕ parameterized by ϕ and a classifier g_θ parameterized by θ, i.e., $S_{\phi,\theta}(x) = g_\theta(f_\phi(x))$. Denote a 2D dense feature map for I and its corresponding semantic feature as $\boldsymbol{F} = f_\phi(I) \in \mathbb{R}^{HW \times D}$ and the ground truth label as $\boldsymbol{Y} \in \mathbb{R}^{HW \times C}$. H, W, and D denote I's height and width, and number of channels, respectively. Existing methods typically obtain the prototype by using the average features of all pixels of a certain class during training. Specifically, prototype $\boldsymbol{P}^c$ of a class c in an image[1] can be formulated as follows,

$$\boldsymbol{P}^c = \frac{\sum_{i=1}^{HW}[\boldsymbol{Y}_i == c]\cdot\boldsymbol{F}}{\sum_{i=1}^{HW}[\boldsymbol{Y}_i == c]} \in \mathbb{R}^D, \tag{1}$$

where $[\cdot]$ denotes the Iverson bracket. To improve the representation relationship between and within classes, many metric strategies $\mathcal{D}(\cdot,\cdot)$ have been proposed and can be grouped into two types: intra-class compactness loss and inter-class dispersion loss. The training loss with prototype regularization can be expressed as (here we take the intra-class pixel-to-prototype compactness loss as an example for illustration (Huang et al. 2022)):

$$\mathcal{L}_{seg} = \mathcal{L}_{ce}(S_{\phi,\theta}(I), \boldsymbol{Y}) + \lambda\mathcal{D}_{intra-p2p}(\boldsymbol{Y}\cdot\boldsymbol{P}, \boldsymbol{F}), \tag{2}$$

where λ is the trade-off that balances the cross-entropy loss $\mathcal{L}_{ce}$ and regularization loss $\mathcal{D}_{intra-c2p}$ which aims to reduce the distance between prototypes and class features. $\boldsymbol{Y}\cdot\boldsymbol{P}$ distributes prototypes to corresponding positions in each image. Similarly, inter-class pixel-to-prototype loss can be expressed as pushing two different classes of pixel features and prototypes apart.

In the setting of classification, class prototypes can be calculated in batch data,

$$\boldsymbol{P}^c = \frac{\sum_{i=1}^{N}[\boldsymbol{Y}_i == c]\cdot\boldsymbol{F}}{\sum_{i=1}^{N}[\boldsymbol{Y}_i == c]} \in \mathbb{R}^D, \tag{3}$$

where N denote the batch size.

3.2 Motivation

Although previous prototype-based methods have achieved significant results, the following two problems still exist: *1*) Feature entanglement. Conventionally, the prototype is generated from the learned feature and updated with consideration of the previous state (*i.e.*, prototypes in memory bank) (Zhou et al. 2022; Wang et al. 2021) during training. As a result, some errors and induced biases accumulate during the whole training process. For example, the bias caused by

[1]To obtain more robust prototypes, previous methods typically calculate the class centers using all the training images in a batch.

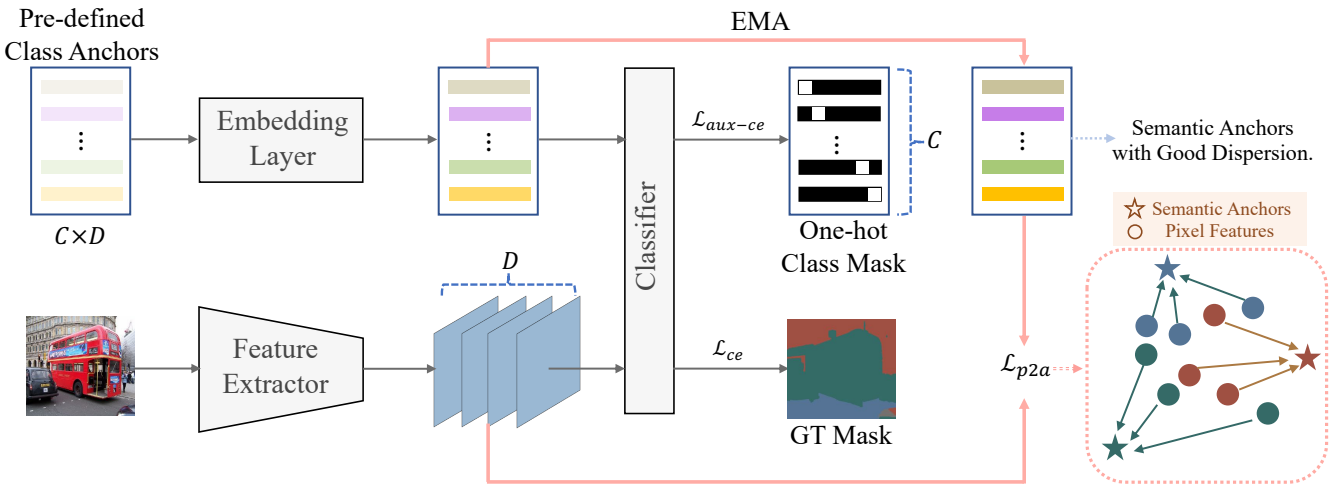

Figure 2: Framework of the proposed method which consists of a main stream (lower stream) for segmentation/classification and an auxiliary stream (the upper stream) for SAR. Pre-defined class anchors are first embedded into the semantic space to mitigate the semantic gap and then categorized by the classifier of the mainstream. The embedded anchors are ensembled into semantic anchors in an EMA manner. The learned feature with dimension is pulled to the corresponding semantic anchor for better intra-class compactness and inter-class separability. Bold pink lines highlight the proposed SAR.

the long-tailed problem, where there are numerous features learned from red cars but very few from green cars in the training set, leads to an overemphasis on color attributes for the car's prototype. 2) Classifier imperceptible. Although a large number of metric functions have been proposed for optimizing inter-class distance in the semantic space, they are not directly perceptible to the model classifier which predicts probabilities.

To address issue 1, we propose Class-anchor Regularization (CR) to decouple feature centriod generation from feature learning, by pulling pixel features for each class to pre-defined class anchors with good angle relationships. Our motivation stems from the fact that in the training paradigm of empirical risk minimization, class representations are not only bound to the data but also guided by the objective function. As seen in Fig.1, class prototypes can be any feature vector in the semantic space as long as they are separable. In this sense, if we explicitly guide class representations towards some pre-defined anchors that are independent of feature learning and well-separated, we can attain more consistent and discriminative class representations. In other words, the prototype is predetermined and consistently maintains good inter-class relationships, as opposed to being estimated from the learned representations through the Expectation-Maximization (EM) paradigm. Errors and biases caused by long-tailed distributions can be effectively minimized compared to EM estimation.

However, as shown in Tab. 7, CR cannot steadily improve performance since suffers from the issue 2. The semantic gap between learned features and class anchors greatly inhibits the effect of class anchors. To solve these problems simultaneously, we further propose the classifier-aware Semantic Anchor Regularization.

3.3 Semantic Anchor Regularization

Semantic Anchor Regularization (SAR) introduces classifier-aware semantic anchors by projecting the pre-defined class anchors into the semantic space and sorting them through the classifier, to address issue 2. As shown in Fig. 2, SAR learns in the fashion of multi-task learning (Caruana 1997) by introducing a simple auxiliary steam (the upper steam) to classify the embedded anchors. The lower stream is the main task stream to perform segmentation/classification based on existing models. The C-way classifier is shared between the auxiliary and main streams. In training, we randomly generate pre-defined class anchors $\boldsymbol{A} \in \mathbb{R}^{C\times D}$ and fix them, and project them into the semantic space through a trainable embedding layer h_ψ, getting namely embedded anchors $h_\psi(\boldsymbol{A})$, and utilizing them update semantic anchors $\hat{\boldsymbol{A}}$ by Exponential Moving Average (EMA) strategy. In this manner, the separability of semantic anchors is guaranteed according to the classifier's decision directly in the semantic space. Hence, shifting class representations toward corresponding semantic anchors can get intra-class compact embedding space and naturally achieve inter-class separability. Specifically, the proposed SAR is a pixel-to-anchor compactness loss by directly minimizing the distance between data representations and corresponding semantic anchors,

$$\mathcal{L}_{p2a} = \mathcal{D}_{mse}(\boldsymbol{F},\ \boldsymbol{Y}\cdot\hat{\boldsymbol{A}}) \tag{4}$$

The next problem that needs to be addressed is how to train embedding layers in a way that disentangles them from the main task.

Disentanglement Learning. To mitigate biased learning resulting from training drift and long-tailed distributed data, two simple yet effective training strategies are proposed to ensure that the semantic anchor is generated independently

of feature learning. *1*) Reweight. The classifier is required to make correct predictions with high confidence for all embedded anchors instead of the high mean confidence. Specifically, the loss for the auxiliary task can be formulated as a weighted cross-entropy loss in Eq. 5.

$$\mathcal{L}_{aux-ce} = -\sum_{i=1}^{C} w_i \log g_\theta^i(h_\psi(\boldsymbol{A}^i)) \tag{5}$$

where w_c denotes the classification weight of the c-th predefined class anchor. A threshold τ is utilized to filter the high-confidence predictions in Eq. 6 and the w_c can be calculated as Eq. 7. By re-normalizing the w_c after high-confidence suppression, more attention can be put on low-confidence embedded anchors,

$$w_c = \begin{cases} 0, & \text{if } g_\theta^c(h_\psi(\boldsymbol{A}^c)) > \tau \\ g_\theta^c(h_\psi(\boldsymbol{A}^c)), & \text{otherwise} \end{cases} \tag{6}$$

$$w_c = \frac{\log(w_c)}{\sum_{i=1}^{C} \log(w_i)}, \tag{7}$$

The above reweight strategy serves two purposes. First, it can be utilized to correct biases towards common classes the classifier learns under the guidance of the main task. Second, attributed to the Eq. 6, embedded anchors with prediction confidence higher than τ are not changed along with the training, it can accelerate the convergence of the auxiliary task, which is already quite simple (C samples, C-way classification), and avoid too much influence on the main task. In practice, for the 160K training schedule on ADE20K (Zhou et al. 2017), the embedding layer is updated frequently only during the initial 600 steps, and subsequently, it is updated approximately every 25 steps. *2*) Update by exponential moving average. Furthermore, to avoid entangled updates of embedded anchors and main task features, we employ the Exponential Moving Average (EMA) manner to get semantic anchors at each training step t,

$$\hat{\boldsymbol{A}}_t = \alpha \hat{\boldsymbol{A}}_{t-1} + (1-\alpha) h_\psi(\boldsymbol{A})_t, \tag{8}$$

In addition, we only use and update semantic anchors when it is correctly classified with a probability greater than δ for better inter-class separation.

In summary, the above training strategy ensures the independence of learning between semantic anchors and pixel features, even though the main task and auxiliary task share the same classifier, which is inherently different from previous works (Huang et al. 2022; Wang et al. 2021; Wu et al. 2023; Hu, Cui, and Wang 2021) collecting prototypes based on the feature learning process.

Overall. Integrating all components, the overall loss for SAR representation learning is the weighted sum of the presented loss components,

$$\mathcal{L}_{seg} = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{aux-ce} + \lambda_2 \mathcal{L}_{p2a} \tag{9}$$

4 Experiments

4.1 Experimental Settings

Semantic segmentation which is a typical and challenging classification task at the pixel level is adopted as the main downstream task to evaluate the proposed method. In addition, We further apply SAR for image classification exploratory experiment in Appendix Sec. A.

Model	Backbone	mIoU
FCN	ResNet-101	75.1
FCN+**SAR**		**75.9 (+0.8)**
DeepLabV3	ResNet-101	80.2
DeepLabV3+**SAR**		**80.6 (+0.4)**
HRNet	HRNetV2-W48	79.9
HRNet+**SAR**		**81.4 (+1.5)**
OCRNet	HRNetV2-W48	80.7
OCRNet+**SAR**		**81.7 (+1.0)**
SegFormer	MiT-B4	81.9
SegFormer+**SAR**		**82.3 (+0.4)**
UPerNet*	Swin-L	82.7
UperNet+**SAR**		**83.2 (+0.5)**

Table 1: Quantitative results on Cityscapes. * represents based on our reproduction.

Datasets. Our experiments are conducted on three datasets, including Cityscapes (Cordts et al. 2016), ADE20K (Zhou et al. 2017), and Pascal-Context (Mottaghi et al. 2014) Cityscapes contains $5{,}000$ fine-grained annotated European street scenes with $2{,}975/500/1{,}524$ for train/val/test. It contains 19 classes for scene parsing or semantic segmentation evaluation. ADE20K is one of the most challenging large-scale scene parsing datasets due to its complex scene and up to 150 category labels. The dataset is divided into $20{,}210/2{,}000/3{,}352$ images for train/val/test, respectively. Pascal-Context is split into $4{,}998/5{,}105$ for training/test set with 59 semantic classes plus a background class. As a common practice in semantic segmentation tasks, we use its 59 semantic classes for evaluation.

Network Architectures. Our implementation is based on the mmsegmentation framework (Contributors 2020) and follows default model configurations. The embedding layer is designed as a stack of two LinearModule (Linear, Bn, ReLU) and one ConvModule (Conv, Bn, ReLU). All backbones are initialized using corresponding weights pre-trained on ImageNet-1K (Deng et al. 2009).

Implementation Details. The proposed SAR and its baselines use the same image augmentation for fair comparisons, including random resize with ratio $[0.5, 2.0]$, random horizontal flipping, random cropping, and random photometric distortion. We empirically set $\lambda_1 = 1$, $\alpha = 0.999$, $\tau = 0.9$ and $\delta = 0.8$ for our all experiments. We use smaller $\lambda_2 = 0.05$ for DeepLabV3 (Chen et al. 2017a), which has a relatively unstable training process. In addition, to ensure generality, all other models use $\lambda_2 = 0.1$, although customizing hyperparameters for each benchmark can further improve performance. Following previous work (Contributors 2020; Chen et al. 2017a; Liu et al. 2021), we use the stochastic gradient descent (SGD) (Robbins and Monro 1951) optimizer with a learning rate of 0.01, weight decay of 0.0005, and momentum of 0.9 for Convolution-based models. For Transformer-based

Model	Backbone	mIoU
FCN	ResNet-101	39.9
FCN+**SAR**		**40.4 (+0.5)**
DeepLabV3	ResNet-101	45.0
DeepLabV3+**SAR**		**45.3 (+0.3)**
HRNet	HRNetV2-W48	42.0
HRNet+**SAR**		**42.8 (+0.8)**
OCRNet	HRNetV2-W48	43.2
OCRNet+**SAR**		**43.7 (+0.5)**
SegFormer	MiT-B5	49.1
SegFormer+**SAR**		**49.5 (+0.4)**
UPerNet	Swin-L	52.2
UperNet+**SAR**		**52.6 (+0.4)**

Table 2: Quantitative results on ADE20K.

models, we use the AdamW (Loshchilov and Hutter 2017) optimizer with a learning rate of 0.00006 and weight decay of 0.01. The learning rate is scheduled following the polynomial annealing policy. For Cityscapes (Cordts et al. 2016), we train a batch size of 8 with a crop size of $512 \times 1{,}024$ (Transformer-based models trained by $1{,}024 \times 1{,}024$ crop size). For ADE20K and Pascal-Context, we train a batch size of 16 with a crop size of 512×512 and 480×480, respectively. Unless otherwise specified, the models are trained for 80k, 160k, and 40k iterations with 8GPUs (Transformer-based models) or 4GPUs (Convolution-based models) on Cityscapes, ADE20K, and Pascal-Context, respectively.

Evaluation Metric. We report mean Intersection over Union (mIoU) over all classes. For fair comparisons, we do not apply any test-time data augmentation. All results reported in the baseline are derived from MMSegmentation (Contributors 2020).

4.2 Main Results

To verify the effectiveness, SAR is evaluated and compared with other SOTA methods on three segmentation benchmarks using different backbone networks.

Tab. 1 shows the performance on Citysacpes (Cordts et al. 2016) dataset. It can be seen that by integrating SAR with FCN (Long, Shelhamer, and Darrell 2015), DeepLabV3 (Chen et al. 2017b), HRNet (Wang et al. 2020), OCR (Yuan, Chen, and Wang 2020), SegFormer (Xie et al. 2021) and Swin Transformer (Liu et al. 2021), their performance in mIoU are increased by 0.8%, 0.4%, 1.5%, 1.0%, 0.4% and 0.5%, respectively. These improvements are significant compared to these commonly used strong baselines.

The consistent performance improvement can be observed in Tab. 2, which adopts the same baselines as Tab. 1. In addition, we also couple SAR with DisAlign (Zhang et al. 2021) which is a two-stage approach specifically designed to address long-tail segmentation. We report results in Tab. 3, After incorporating DisAlign (DA), we achieved further improvements in the column of mTailIoU (34.5% v.s. 34.3%). This implies that our approach can effectively serve as a complement to methods focused on long-tailed distributions.

	mIoU	mHeadIoU	mBodyIoU	mTailIoU
Stage1				
HRNet	42.0	65.5	46.0	32.8
HRNet+SAR	42.7 (+0.7)	66.2 (+0.7)	45.6 (-0.4)	34.3 (+1.5)
Stage2				
DA+HRNet	42.2 (+0.2)	65.6 (+0.1)	46.0 (+0.0)	33.1 (+0.3)
DA+SAR	**42.9 (+0.9)**	66.1 (+0.6)	46.0 (+0.0)	**34.5 (+1.7)**

Table 3: Incremental improvements for DisAlign (DA) that is focused on long-tail segmentation on ADE20K.

Model	Backbone	mIoU (%)
FCN	ResNet-101	48.4
FCN+**SAR**		**49.7 (+1.3)**
DeepLabV3	ResNet-101	52.6
DeepLabV3+**SAR**		**53.3 (+0.7)**
HRNet	HRNetV2-W48	50.3
HRNet+**SAR**		**51.1 (+0.8)**
OCRNet*	HRNetV2-W48	52.0
OCRNet+**SAR**		**52.4 (+0.4)**

Table 4: Quantitative results on Pascal-Context. * represents based on our reproduction.

To show SAR's capacity for effectively handling tailed classes, we also perform experiments on Pascal-Context which follows serious long-tail distributions. The overall performance is shown in Tab. 4 (MMSeg does not provide available config for Transformer-based methods on this dataset), while for a detailed analysis of specific tail-end classes, please refer to Appendix Sec. B.

4.3 Comparison with Prototype-based Methods

We conduct a fair comparison between SAR and other important prototype-based methods, such as ProtoSeg (Zhou et al. 2022) and CAR (Huang et al. 2022), as these methods employ experimental settings that differ from the MMSeg benchmark. For a performance comparison of the classification task, please refer to Appendix Sec. A.

Method	Resolution	Schedule	mIoU
Model learned on ADE20K			
HRNet	160K	512×512	42.0
SAR	160K	512×512	42.8(+0.8)
ProtoSeg	160K	520×520	43.0(+1.0)
SAR	160K	520×520	**43.3(+1.3)**
Model learned on Cityscapes			
HRNet	80K	1024×512	79.9
HRNet	160K	1024×512	80.6(+0.7)
ProtoSeg	160K	1024×512	81.1(+1.2)
SAR	80K	1024×512	**81.4(+1.5)**

Table 5: Fair comparison of SAR and ProtoSeg based on HRNet as the baseline.

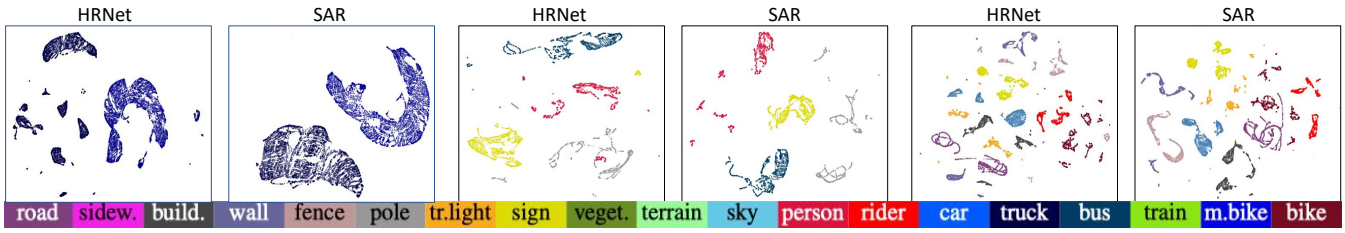

Figure 3: Visualization of the learned features with HRNet and SAR on Cityscapes utilizing UMAP.

Method	mIoU	Method	mIoU	Method	mIoU
DLV3	52.6	HRNet	50.3	OCRNet	52.0
CAR	52.9 (+0.3)	CAR	50.7 (+0.4)	CAR	52.3 (+0.3)
SAR	**53.3** (**+0.7**)	SAR	**51.1** (**+0.7**)	SAR	**52.5** (**+0.5**)

Table 6: Fair comparison with CAR on Pascal-Context using 520×520 training crops. DLV3: DeepLabV3

$\mathcal{L}_{ce}$	$\mathcal{L}_{p2a}$	$\mathcal{L}_{aux-ce}$	EMA	Reweight	mIoU (%)
✓					79.9
✓	ND				79.8∼80.3
✓	OM				79.2∼79.9
✓	MES				79.8∼80.4
✓	N	✓			80.6 (**+0.7**)
✓	N	✓	✓		81.1 (**+1.2**)
✓	N	✓	✓	✓	81.4 (**+1.5**)

Table 7: Ablation studies on the key components of our proposed SAR on Cityscapes. ND: standard Normal Distribution, OM: random Orthogonal Matrix, MES: random matrix with a Maximum Equiangular Separability structure.

4.4 Ablation Studies

In Tab. 7, we evaluate the efficacy of each component in the proposed SAR on Cityscapes (Cordts et al. 2016). $\mathcal{L}_{ce}$ means the case only using HRNet as baseline. Without embedding layer (+$\mathcal{L}_{p2a}$), learned features in the segmentation task are directly regularized by the pre-defined anchors A which are randomly sampled from the three sources. As discussed in Sec. 3.2, these random class anchors can improve the performance of the baseline but with strong variations. To reduce semantic gaps between class anchors and semantic space, we embed the pre-defined anchor into semantic space (+embedding layer) and control their separability using the classifier for segmentation (+$\mathcal{L}_{aux-ce}$). In this manner, a stable improvement of 0.7 in mIoU can be obtained. Further, the EMA updating strategy and Reweighting strategy are utilized in disentanglement learning these semantic anchors. Combining all components, SAR can achieve an increment of 1.5 in mIoU compared to the baseline.

Detailed Analyses. More detailed ablation studies can refer to Appendix Sec. C, including independence of semantic anchor, model robustness, hyper-parameters sensitivity, and extra computational and storage burden analyses.

4.5 Qualitative Evaluation on the Segmentation Results

Visualization of learned representations. Fig. 3 visualizes the feature learned with and without the proposed SAR using UMAP (McInnes, Healy, and Melville 2018) analysis. Learning with SAR improves intra-class compactness and inter-class separability. According to the basic assumption proposed in (Oliver et al. 2018), the decision boundary generated by SAR will pass through more sparse regions and have stronger robustness and generalization

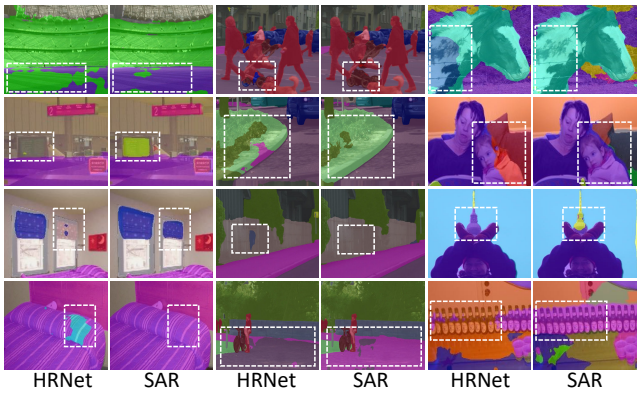

Figure 4: Qualitative results on ADE20K (L. 2 Cols.), Cityscapes (M. 2 Cols.), and Pascal-Context (R. 2 Cols.).

Qualitative results. We present qualitative examples of the segmentation results in Fig. 4. Examples are from ADE20K, Cityscapes, and Pascal-Context, respectively. The results from the HRNet and HRNet training with SAR are included for comparison.

5 Conclusion

In this paper, we present that prototype representations derived from the learned features are sub-optimal since they heavily rely on the data distribution. We proposed a novel perspective to leverage pre-defined class anchors which are decoupled from pixel features to guide representation learning. However, directly using these anchors suffers from the semantic gap between pre-defined anchors and learned features in the semantic space. To address this issue, we proposed semantic anchor regularization (SAR) for improved class representation. SAR adopts a disentangled learning approach to collect these semantic anchors, using them to

unidirectionally guide feature learning. SAR can be applied in a plug-and-play manner to help existing models achieve better performance and address long-tail distributions. Experiments on downstream semantic segmentation with extensive ablation studies have validated the effectiveness of the proposed SAR method. In addition, exploratory experiments in Appendix Sec. A show SAR is promising as a general solution for classification-based tasks. We hope that our proposal can advance future studies of representation learning and imbalanced learning. Limitations and future work are provided in Appendix Sec. D.

6 Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants No. 82121003 and No. 62176047, and the Shenzhen Fundamental Research Program under Grant No. JCYJ20220530164812027.

References

Arik, S. Ö.; and Pfister, T. 2020. Protoattend: Attention-based prototypical learning. *The Journal of Machine Learning Research*, 21(1): 8691–8725.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28: 41–75.

Chen, C.-F. R.; Fan, Q.; and Panda, R. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, 357–366.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4): 834–848.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017b. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chuang, C.-Y.; Robinson, J.; Lin, Y.-C.; Torralba, A.; and Jegelka, S. 2020. Debiased contrastive learning. *Advances in neural information processing systems*, 33: 8765–8775.

Contributors, M. 2020. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation.

Contributors, M. 2023. OpenMMLab's Pre-training Toolbox and Benchmark. https://github.com/open-mmlab/mmpretrain.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 3213–3223.

Cover, T.; and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1): 21–27.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Ge, W. 2018. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 269–285.

Girdhar, R.; El-Nouby, A.; Liu, Z.; Singh, M.; Alwala, K. V.; Joulin, A.; and Misra, I. 2023. ImageBind: One Embedding Space To Bind Them All. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15180–15190.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hong, Y.; Pan, H.; Sun, W.; Yu, X.; and Gao, H. 2022. Representation Separation for Semantic Segmentation with Vision Transformers. *arXiv preprint arXiv:2212.13764*.

Hu, H.; Cui, J.; and Wang, L. 2021. Region-aware contrastive learning for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16291–16301.

Huang, J.; Dong, Q.; Gong, S.; and Zhu, X. 2019. Unsupervised deep learning by neighbourhood discovery. In *International Conference on Machine Learning*, 2849–2858. PMLR.

Huang, Y.; Kang, D.; Chen, L.; Zhe, X.; Jia, W.; Bao, L.; and He, X. 2022. Car: Class-aware regularizations for semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, 518–534. Springer.

Jiang, Z.; Li, Y.; Yang, C.; Gao, P.; Wang, Y.; Tai, Y.; and Wang, C. 2022. Prototypical contrast adaptation for domain adaptive semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, 36–54. Springer.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Kwon, H.; Jeong, S.; Kim, S.; and Sohn, K. 2021. Dual prototypical contrastive learning for few-shot semantic segmentation. *arXiv preprint arXiv:2111.04982*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lu, Y.; Luo, Y.; Zhang, L.; Li, Z.; Yang, Y.; and Xiao, J. 2022. Bidirectional self-training with multiple anisotropic prototypes for domain adaptive semantic segmentation. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1405–1415.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv*.

Moon, T. K. 1996. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6): 47–60.

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; and Yuille, A. 2014. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 891–898.

Nguyen, K.; and Todorovic, S. 2019. Feature weighting and boosting for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 622–631.

Oliver, A.; Odena, A.; Raffel, C. A.; Cubuk, E. D.; and Goodfellow, I. 2018. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Papyan, V.; Han, X.; and Donoho, D. L. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663.

Perronnin, F.; Sánchez, J.; and Mensink, T. 2010. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, 143–156. Springer.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Sohn, K. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Wang, F.; and Liu, H. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2495–2504.

Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. 2020. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3349–3364.

Wang, K.; Liew, J. H.; Zou, Y.; Zhou, D.; and Feng, J. 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In *proceedings of the IEEE/CVF international conference on computer vision*, 9197–9206.

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 9929–9939. PMLR.

Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; and Van Gool, L. 2021. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7303–7313.

Wu, D.; Guo, Z.; Li, A.; Yu, C.; Gao, C.; and Sang, N. 2023. Semantic Segmentation via Pixel-to-Center Similarity Calculation. *arXiv preprint arXiv:2301.04870*.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3733–3742.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34.

Xu, W.; Xian, Y.; Wang, J.; Schiele, B.; and Akata, Z. 2020. Attribute prototype network for zero-shot learning. *Advances in Neural Information Processing Systems*, 33: 21969–21980.

Yang, F.; Wu, K.; Zhang, S.; Jiang, G.; Liu, Y.; Zheng, F.; Zhang, W.; Wang, C.; and Zeng, L. 2022. Class-aware contrastive semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14421–14430.

Yuan, Y.; Chen, X.; and Wang, J. 2020. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, 173–190. Springer.

Zhang, S.; Li, Z.; Yan, S.; He, X.; and Sun, J. 2021. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2361–2370.

Zhong, Z.; Cui, J.; Liu, S.; and Jia, J. 2021. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16489–16498.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641.

Zhou, T.; Wang, W.; Konukoglu, E.; and Van Gool, L. 2022. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2582–2593.

Appendix

The Appendix is organized as follows. Sec. A further explores the application of our proposed method to the image classification task. Sec B shows the performance of SAR on rare classes. Sec. C provides detailed ablation studies of our proposed methods. Notably, in Sec. C.1, we present two clear pieces of experimental evidence supporting the semantic anchors independent of feature learning. Sec. D discusses the limitations and future work of SAR.

A Application to Image Classification Task

In the exploratory experiment, we further apply SAR to the image-level classification task. We evaluate SAR on image classification performance in normal and long-tail settings.

A.1 Normal Setting

Our experiments are conducted on two datasets, CIFAR-100 (Krizhevsky, Hinton et al. 2009) and CUB-200 (Wah et al. 2011). CIFAR-100 is a subset of the tiny images dataset and consists of $60{,}000$ images. The 100 classes in the CIFAR-100 are grouped into 20 super-classes. There are 600 images per class which are split into 500 training images and 100 testing images per class. CUB-200 is a widely used dataset for fine-grained classification tasks. We use the CUB-200-2011 version, which contains much more pictures than the original CUB-200. It contains $11{,}788$ images of 200 subcategories belonging to birds, $5{,}994$ for training, and $5{,}794$ for testing.

Our implementation is based on the MMPretrain framework (Contributors 2023) and follows default model configs and training schedules. We use ResNet(He et al. 2016) as the baseline model. Compared with the segmentation, the embedding layer is simpler and designed as two LinearModules. All hyper-parameters of SAR are the same as the segmentation task that we reported in the manuscript. We use Top-1 accuracy for evaluation. The reported baseline results are derived from MMPretrain. Tab. 8 and Tab. 9 show the performance on CIFAR-100 and CUB-200 datasets, respectively. Through the use of SAR, our approach has shown a noticeable increase in Top-1 accuracy by 0.57 and 0.25 on both datasets for ResNet-50, respectively. As for ResNet-18, we have achieved a significant improvement in Top-1 accuracy by 0.71 and 1.06 on both datasets, respectively. The results demonstrate the effectiveness and potential of SAR at the image-level classification.

Method	Top-1 Acc
ResNet-18	78.07
ResNet-18+**SAR**	**78.78 (+0.71)**
ResNet-50	79.90
ResNet-50+**SAR**	**80.47 (+0.57)**

Table 8: Quantitative results on CIFAR-100.

Method	Top-1 Acc
ResNet-18	83.10
ResNet-18+**SAR**	**84.16 (+1.06)**
ResNet-50	88.19
ResNet-50+**SAR**	**88.44 (+0.25)**

Table 9: Quantitative results on CUB-200.

A.2 Long-tailed Setting

For long-tailed classification, we employ the CIFAR-100-LT (Cao et al. 2019), which is the long-tailed version of the CIFAR dataset. It is collected by controlling the degrees of data imbalance with an imbalanced factor (IF) $\beta = \frac{N_{max}}{N_{min}}$, where N_{max} and N_{min} are the numbers of training samples for the most and the least frequent classes. We conduct experiments with IF=100.

Our implementation is based on the MiSLAS (Zhong et al. 2021) which is a two-stage approach specifically designed to address long-tailed classification, and follows default model configs (ResNet-32) and training schedules. Furthermore, we reproduce the CAR approach (Huang et al. 2022), an outstanding prototype-based method, in this experiment to compare its outcomes with SAR in long-tailed classification data. These results are summarized in Tab. 10. By integrating SAR, MiSLAS achieves an improvement of 1.0% mIoU over the baseline. However, due to insufficient annotations in the classification task to support CAR in calculating reliable class centers during feature learning, along with the impact of long-tailed distributions, it becomes difficult to apply it to classification tasks.

Stage1		Stage2	
Method	Top-1 Acc	Method	Top-1 Acc
MixUp	39.5	MiSLAS	47.0
MixUp+CAR	38.0 (-1.5)	MiSLAS	45.3 (-1.7)
MixUp+SAR	**40.6(+1.1)**	MiSLAS	**48.0 (+1.0)**

Table 10: Quantitive results on CIFAR-100-LT. The stage 2 model is initialized by the stage 1 model of the same row in the table.

A.3 Summary

The above experiments illustrate the potential of the SAR-based perspective to become a generic component for addressing challenges posed by representation learning and long-tailed distribution problems.

B Performance on Rare Classes

Semantic segmentation is inherently a long-tailed problem. To show the SAR that decoupled from feature learning has better robustness in long-tailed problems, we report the results of Top 4 rare classes in ADE20K (Zhou et al. 2017), Cityscapes (Cordts et al. 2016), and Pascal-Context (Mottaghi et al. 2014) datasets, as shown in Tab. 11. SAR improves

Class	HRNet	SAR	Δ	Class	HRNet	SAR	Δ	Class	HRNet	SAR	Δ
Radiator	45.5	54.0	**(+8.5)**	Train	75.5	83.9	**(+8.4)**	Cup	31.1	33.8	**(+2.7)**
Glass	10.9	11.3	**(+0.4)**	Tr.Light	74.9	75.0	**(+0.1)**	Sign	36.3	39.1	**(+2.8)**
Clock	17.3	23.8	**(+6.5)**	Rider	65.4	67.5	**(+2.1)**	Light	39.7	40.5	**(+0.8)**
Flag	30.2	30.3	**(+0.1)**	M.Bike	68.2	68.3	**(+0.1)**	Mouse	34.7	40.3	**(+5.6)**

Table 11: Performance on the Top 4 rare classes of the ADE20K (left), Cityscapes (middle), and Pascal-Context (right), respectively.

significantly in these rare classes. The above performance demonstrates the robustness of the proposed method to long-tailed problems across different datasets. As the generation of class anchors is less affected by feature learning, they are insensitive to the number of samples in different classes. In addition, the reweighting strategy in Eq. 6 ensures that the model can focus more on false predictions, which are usually tailed classes. Therefore, fewer common-case biases will be introduced from semantic anchors when serving as the feature centroid for representation learning. For example, the rarest "Mouse" class in the Pascal-Context dataset accounts for only $7 \times 10^{-3}\%$ of the entire dataset. SAR improves the IoU of "Mouse" by 5.6% to 40.3%.

C Detailed Ablation Studies

C.1 Independence of Semantic Anchors.

In addition to the disentanglement learning analysis mentioned in Sec. 3.3 and the SAR's capacity to address long-tail problems, there are three more straightforward experimental phenomenons that demonstrate the independence of semantic anchors from feature learning.

Segmentation on extremely limited data. As mentioned in Sec. B the "Mouse" class has an extremely rare appearance with $7 \times 10^{-3}\%$ probability in Pascal-Context, we observe that the "Mouse" class was never predicted correctly in the training results of the three DeepLabV3 with different random seeds. Tab. 12 shows the performance of SAR on the "Mouse" class with the same seed. This demonstrates the independence between SAR and learned features, as the learned features do not include the effective recognition features for "Mouse".

Seed	DeepLabV3	SAR
1270964153	0.0	**23.4 (+23.4)**
1024	0.0	**25.6 (+25.6)**
5555	0.0	**35.4 (+35.4)**

Table 12: Quantitative results (IoU) on the "Mouse" class. Our result is based on DeepLabV3 with SAR.

Compare with CAR on the "Mouse" class. An instance is present in Tab. 13, we compared the performance of our method with CAR on the "Mouse" class. CAR is an excellent prototype-based method that calculates prototypes on learned features. However, due to the extremely low frequency of the "Mouse" class, the accumulation of error and bias causes the training of CAR to collapse in this class. On the contrary, since SAR is independent of feature learning, it actually improves the performance of the "Mouse" class.

Model	IoU
HRNet	34.7
CAR	0.0 (-34.7)
SAR	**40.3 (+5.6)**

Table 13: Comparing the IoU of SAR and CAR on the "Mouse" class.

C.2 Robustness to Network Initialization

Our method is correlated with the baseline model and robust to the network initialization. Tab. 14 and Tab. 15 show the performance of multiple seeds on two benchmarks, respectively. As the result shows, our method consistently improves the mIoU over its baseline using different random seeds, which demonstrates the effectiveness and robustness of SAR.

Seed	HRNet	SAR
1270964153	79.9	**81.4 (+1.5)**
1024	79.8	**81.0 (+1.2)**
5555	78.9	**80.1 (+1.2)**

Table 14: Error analyze HRNet on Cityscapes.

Seed	DeepLabV3	SAR
1270964153	52.6	**53.3 (+0.7)**
1024	52.4	**53.4 (+1.0)**
5555	52.6	**53.3 (+0.7)**

Table 15: Error analyze DeepLabV3 on Pascal-Context.

C.3 Hyper-parameter Analysis

We conduct ablation experiments on the hyper-parameters of HRNet on Cityscapes. Tab. 16 and Tab. 17 summarizes the influence of hyper-parameters λ_1 and λ_2 to model performance, respectively. It can be observed that the model performance is robust to the two trade-offs which balance

the effect of the proposed auxiliary cross-entropy loss and pixel-to-anchor loss.

λ_1	0.5	1	2
mIoU	80.9	**81.4**	81.3

Table 16: Sensitivity to λ_1 on Cityscapes.

λ_2	0.05	0.1	0.2
mIoU	81.2	**81.4**	81.0

Table 17: Sensitivity to λ_2 on Cityscapes.

Tab. 18 and Tab. 19 show studies on τ for the auxiliary loss reweighting and δ for class anchors update strategies, respectively. The τ filters class anchors with prediction confidence higher than it and makes the model put more attention on anchors that have lower confidence. However, a low τ leads the model to ignore some embedded anchors with not so high classification confidence, which means their inter-class distance to other class anchors is underoptimized. As a result, anchors are not dispersedly distributed in the semantic space and the inter-class distance between anchors might be not well. With a proper τ, more attention can be put on low-confidence anchors and broadening the inter-class distance between those not well-separated anchors. The δ determines whether an embedded anchor is used as the regularization for feature learning. Similarly, for δ, a high threshold ensures the learned feature is only regularized by those anchors with good inter-class separability. A value of δ less than τ means that class anchors with confidence between δ and τ are continuously optimized and utilized as regularization.

τ	0.5	0.7	0.9	1
mIoU	80.4	81.2	**81.4**	81.0

Table 18: Sensitivity to τ on Cityscapes.

δ	0.5	0.7	0.8	0.9
mIoU	80.3	81.3	**81.4**	81.0

Table 19: Sensitivity to δ on Cityscapes.

C.4 Computational and Storage Burden

SAR requires conducting an auxiliary task during training, which brings additional training parameters. However, in practice, the process only imposes a minor computational and storage burden (See Tab. 20). Compared to the original HRNet, our method only adds 0.03GFLOPs and 1.56M (2.3%) training parameters when input images have a size of 1024×1024.

Model	Flops (GFLOPs)	Δ_1	Params (M)	Δ_2
HRNet	374.34		65.86	
HRNet+SAR	374.37	0.03	67.42	1.56

Table 20: Comparison of the Computation and storage burden on input size as $1{,}024 \times 1{,}024$

D Discussion

D.1 Limitations

In this work, we did not explore the application of SAR to object detection and query-based segmentation methods. We did not use these semantic anchors during the testing phase to ensure speed, but they may be beneficial for the performance during testing.

D.2 Future Work

The key insight of this study is that the features utilized to regularize feature learning do not necessarily come from the task being trained. This enables us to integrate external controlled information to regularize or reinforce the training task, which is in line with the main idea of the now popular multi-modal recognition (Girdhar et al. 2023; Radford et al. 2021). Hence, 1) constructing semantic anchors from a multi-modal perspective to organize embedding space presumably further enhances the representation capability of the model. 2) In addition, using these semantic anchors as additional information during inference through a query-based classification idea. 3) Given the effectiveness of our approach in the classification task, initializing the segmentation model with weights pre-trained under SAR, and training the segmentation model using SAR may lead to a synergistic effect where the whole is greater than the sum of its parts.