Scribble-Supervised Semantic Segmentation with Prototype-based Feature Augmentation

Guiyang Chan [1 2] **Pengcheng Zhang** [1 2] **Hai Dong** [3] **Shunhui Ji** [1 2] **Bainian Chen** [1 2]

Abstract

Scribble-supervised semantic segmentation presents a cost-effective training method that utilizes annotations generated through scribbling. It is valued in attaining high performance while minimizing annotation costs, which has made it highly regarded among researchers. Scribble supervision propagates information from labeled pixels to the surrounding unlabeled pixels, enabling semantic segmentation for the entire image. However, existing methods often ignore the features of classified pixels during feature propagation. To address these limitations, this paper proposes a prototype-based feature augmentation method that leverages feature prototypes to augment scribble supervision. Experimental results demonstrate that our approach achieves state-of-the-art performance on the PASCAL VOC 2012 dataset in scribble-supervised semantic segmentation tasks. The code is available at https://github.com/TranquilChan/PFA.

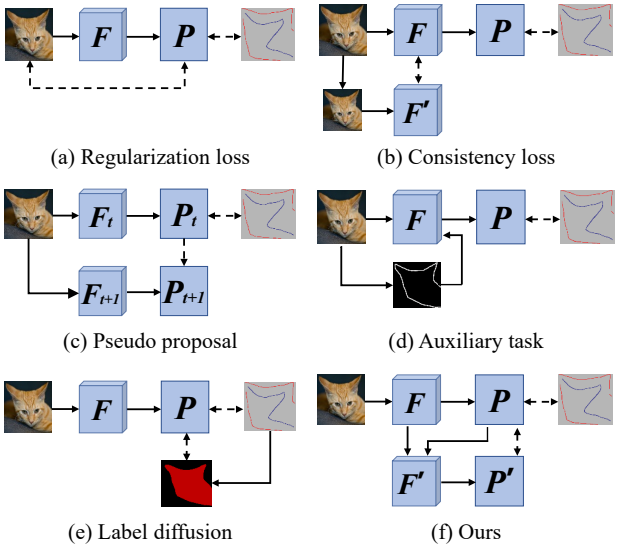


Figure 1. Illustration of existing scribble-supervised semantic segmentation approaches. ***F*** represents the feature map, and ***P*** represents the prediction map.

1. Introduction

In recent years, the rapid progress of deep learning techniques has propelled deep neural networks to achieve significant advancements in image semantic segmentation tasks. These networks play a pivotal role in aiding human comprehension of image content, providing precise pixel-level segmentation. As one of the most intricate tasks in the field of computer vision, training semantic segmentation models typically requires a large number of high-quality annotated samples. However, annotating samples at the pixel level demands a substantial amount of manpower and time, and the annotation process can be tedious. Therefore, in scenarios where data dependency is strong, researchers are increasingly focusing on methods that utilize scribble labels for supervised learning. Training with scribble labels falls under weakly supervised learning, allowing annotators to mark regions in the image using simple lines or sketches and assign corresponding labels to those regions. Compared to pixel-level annotation, using scribble labels can significantly reduce the workload of annotation and improve efficiency. Additionally, compared to point (Bearman et al., 2016; Lee et al., 2021), bounding box (Kulharia et al., 2020; Zhang et al., 2021), and image-level (Zhang et al., 2020; Du et al., 2022) labels, scribble labels provide more crucial semantic information to the models, helping them better learn the semantic structure of the images.

As shown in Figure 1, existing methods mainly rely on *regularization loss* (Tang et al., 2018a;b; Obukhov et al., 2019; Marin et al., 2019; Liang et al., 2022), *consistency loss* (Ke et al., 2021; Pan et al., 2021), *pseudo proposal* (Lin et al., 2016; Zhang et al., 2021; Xu et al., 2021), *auxiliary tasks* (Wang et al., 2019; Pan et al., 2021), and *label diffu-*

[1]Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, China [2]College of Computer Science and Software Engineering, Hohai University, Nanjing, China [3]School of Computing Technologies, RMIT University, Melbourne, Australia. Correspondence to: Pengcheng Zhang <pchzhang@hhu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

sion (Wu et al., 2023; Zhang et al., 2024). However, these methods exhibit certain drawbacks. Regularization methods often overlook leveraging high-level semantic information. Consistency loss does not provide direct supervision at the category level. Pseudo-labeling methods require multi-stage training, which are time-consuming. Auxiliary tasks introduce additional data and predictive errors of introduced data can influence the final outcomes. Label diffusion primarily relies on local information and fails to utilize global information to leverage the features of correctly classified pixels. In fact, the features of pixels that have been correctly classified can play a pivotal role in guiding the classification of pixels in boundary regions. However, many existing methods based on scribble supervision ignore this role.

In the context of semi-supervised classification tasks, FeatMatch (Kuo et al., 2020) learns and extracts feature prototypes from labeled samples, subsequently enhancing the features of unlabeled data with these prototypes to improve the classification of unlabeled samples. Inspired by this, we seek to extend its application to scribble-level weakly supervised semantic segmentation. Nevertheless, in our scenario of scribble-level weakly supervised segmentation, the labels are assigned at the pixel level, unlike the image-level labels in semi-supervised tasks. Our approach entails the initial learning from labeled pixels, extraction of feature prototypes from accurately classified pixels, and the subsequent utilization of these prototypes to guide the classification of remaining pixels.

Specifically, our method initiates with prototype extraction. In contrast to conventional clustering methods employed in semi-supervised and unsupervised approaches, we directly extract feature prototypes from high-confidence regions of initial predictions. To mitigate the potential loss of prototype diversity associated with this extraction method (where clustering methods may yield multiple prototypes for each category), and recognizing the presence of labeled information in each image within a weakly supervised environment, we introduce both local and global prototypes. Local prototypes are extracted from each image, while each category's global prototypes consist of multiple local prototypes. Throughout the training process, local prototypes dynamically update global prototypes, stored using a memory mechanism to minimize additional computation. Combining these two prototype types with prototype-based feature augmenters enhances initial features. The augmented predictions are regularized using a consistency loss. Leveraging feature prototypes enables more effective utilization of information from labeled pixels, enhancing segmentation accuracy by guiding the classification of other pixels.

In summary, our contributions are as follows:

- We propose a prototype-based feature augmentation method for scribble-level weak supervision, extending the application of the method from image-level to pixel-level and significantly enhancing the efficacy of scribble supervision.
- We propose a dynamic augmentation strategy employing local and global prototypes. This synergistic approach maximizes the utilization of information in scribble-supervised semantic segmentation and mitigates the challenge of limited prototype representation at various training stages.
- We validate the components of our method through experiments and report the state-of-the-art performance on the PASCAL VOC 2012 dataset.

2. Related Work

2.1. Scribble-Supervised Semantic Segmentation

Scribble-supervised semantic segmentation, a form of weakly supervised semantic segmentation, employs scribbles as annotations to label image regions. This approach presents a cost-effective alternative to fully supervised labeling. Compared to other weakly supervised annotation methods, including point, bounding-box, and image-level labels, scribble supervision imparts more comprehensive and detailed information, resulting in better performance.

In scribble-supervised semantic segmentation, ScribbleSup (Lin et al., 2016) first introduced the concept of using scribble labels for semantic segmentation and provided the ScribbleSup dataset. They propagated the scribble label information to surrounding pixels using superpixels and designed corresponding loss functions for supervision. Tang et al. proposed methods based on regularization losses, such as Normalized Cut loss (Tang et al., 2018a) and Kernel Cut loss (Tang et al., 2018b), for model training. Gated CRF (Obukhov et al., 2019) augmented the efficiency by introducing gating operations on top of the Kernel Cut loss. RAWKS (Vernaza & Chandraker, 2017) and BPG (Wang et al., 2019) utilized boundary detectors to assist the models in achieving better results. URSS (Pan et al., 2021) reduced uncertainty through neural representation and self-supervision in the neural feature space. SPML (Ke et al., 2021) improved performance by using metric learning methods and introducing a contour detector as additional supervision. PSI (Xu et al., 2021) utilized latent contextual dependency to enhance and refine segmentation results from partially known seeds. A2GNN (Zhang et al., 2021) introduced a graph neural network approach to generate pseudo-labels and applied multi-level supervision. TEL (Liang et al., 2022) proposed a novel tree energy loss method that provides semantic guidance to unlabeled pixels. AGMM (Wu et al., 2023) implemented supervision by constructing Gaussian mixture models based on the feature distribution

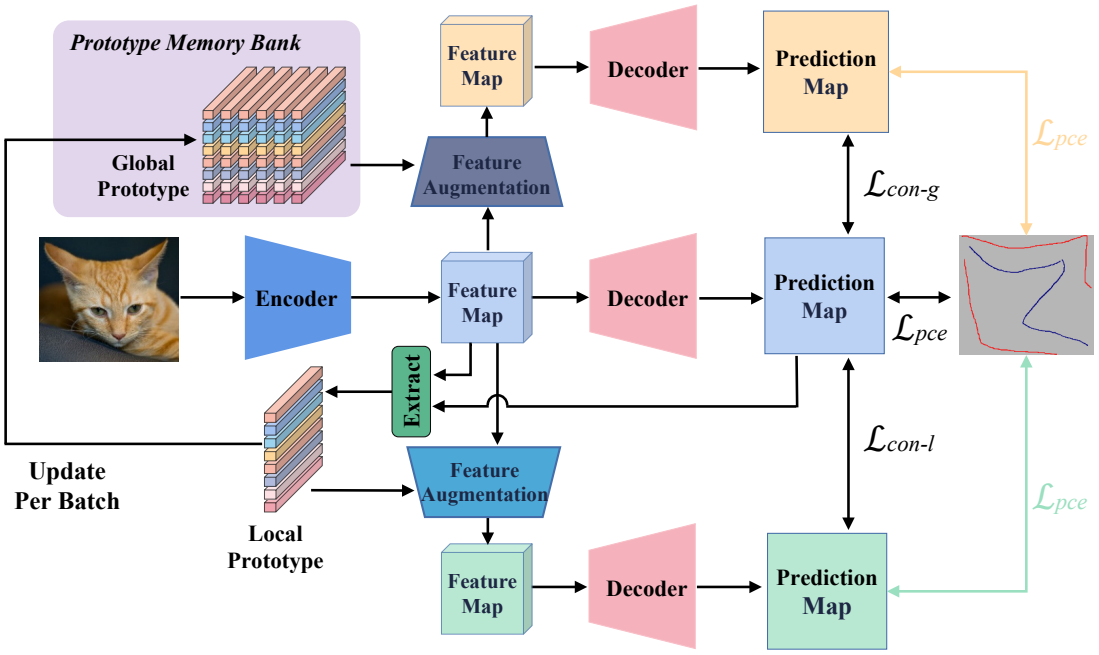


Figure 2. The overall framework of our approach. Initially, the image undergoes encoding to produce a feature map. Subsequently, this feature map is fed into the decoder to generate a semantic segmentation prediction map. Scribble labels are employed to impose constraints using partial cross-entropy loss. Next, local prototypes are extracted from the initial prediction map and the feature map, while global prototypes are updated throughout the training iterations. The initial feature map is augmented separately using these two types of prototypes, and predictions are generated using the decoder. Consistency loss is used to constrain between two predicted maps and the initial predicted map. During the warm-up phase, partial cross-entropy is black. It turns green when global prototypes are inactive, and yellow when global prototypes are in use.

of labeled pixels and unlabeled pixels sharing similar feature distributions. CDSP (Zhang et al., 2024) utilized pseudo-labels supervised by image-level class labels and global semantics.

2.2. Prototype-based Method

Feature prototypes can be seen as 'exemplars' of different categories in the feature space. Their essence lies in being feature vectors widely used in computer vision tasks to augment the model's recognition capability of different types of features. In fully supervised semantic segmentation tasks, OCRNet (Yuan et al., 2020), ACFNet (Zhang et al., 2019), and CondNet (Yu et al., 2021) aggregate category feature embeddings by considering the initial segmented regions. Mask2former (Cheng et al., 2022) and CFT (Tang et al., 2023) focus on category features through masking. The main idea of these methods is to augment features through prototypes, but their methods are relatively common, such as convolution, multiplication, and attention. In weakly supervised semantic segmentation, EPS (Yoon et al., 2021) utilizes prototypes to guide the model in learning more accurate feature representations, thereby improving segmentation results. PPC (Du et al., 2022) provides pixel-level supervisory signals by contrasting pixels with prototypes to narrow the gap between classification and segmentation. SIPE (Chen et al., 2022) proposes an image-specific prototype exploration method to capture complete regional information in the image. They delve into the use of feature prototypes, but they do not use prototypes for feature augmentation, or their employed fusion approaches are relatively simple, thereby failing to harness the guiding role of prototype. FeatMatch (Kuo et al., 2020) augments features by fusing them with prototypes, but it is a semi-supervised classification task. Xu et al. further introduces prototype-augmented methods into the field of semantic segmentation (Xu et al., 2022), but it remains a semi-supervised task. The goal of our work is to incorporate this idea into weakly supervised semantic segmentation, introducing prototype augmentation at the pixel level.

3. Method

3.1. Overview

Figure 2 illustrates the overall framework of our approach, leveraging a Vision Transformer (Xie et al., 2021) as the backbone for extracting initial feature maps. These feature maps are then input to a decoder for generating semantic segmentation prediction maps. Details of the encoder and decoder are discussed in Section 4.2. Supervised by scribble labels through partial cross-entropy loss, the semantic

segmentation prediction maps are refined. Subsequently, high-confidence regions from the initial prediction maps are identified as accurately predicted pixels, and corresponding feature vectors in the initial feature maps are extracted. Local prototypes are formed by weighting and averaging these feature vectors based on predicted values. Throughout training iterations, these local prototypes update global prototypes, elucidated in Section 3.2. Both the global prototypes and local prototypes are used to augment the initial features through prototype-based feature augmenters. The augmented feature maps are then passed through the decoder to generate augmented prediction maps, with the weights of the three decoders shared during this process. The consistency loss is applied between the two augmented prediction maps and the initial prediction maps for supervision, detailed in Section 3.3.

3.2. Prototype Extraction and Update

3.2.1. Setting of the prototype

Augmenting features via prototypes depends heavily on precisely defined prototypes. The creation of these prototypes is vital. We classify prototypes into two types: local prototypes and global prototypes. Local prototypes are extracted from image features within each batch during training iterations and are specifically for augmenting features within that batch. In contrast, global prototypes encompass more comprehensive information and can be dynamically updated globally. Global prototypes augment information diversity by updating with local prototypes, rendering them more suitable for feature augmentation.

3.2.2. Extraction of the prototype

Here, we outline the process of extracting prototypes from features, namely generating local prototypes. Currently, methods for extracting prototypes from image features are mostly employed in semi-supervised and unsupervised semantic segmentation techniques. Many of these methods (Kuo et al., 2020; Xu et al., 2022) utilize clustering as an effective approach in scenarios where labels are either scarce or lacking. However, the primary goal of prototype extraction is to derive highly representative features from the data as the basis for prototype generation. Using clustering methods often fails to effectively align with the actual categories. In a weakly supervised setting, we believe that relying solely on clustering methods does not fully leverage the semantic information provided by scribble labels. These labels can significantly contribute by offering valuable insights into the representative regions required for prototype extraction.

Therefore, we utilize the initial prediction values, represented as *p*, generated by the model as confidence scores. Assisted by category labels, we compute prototypes from high-confidence pixel-level features of the categories in the current image. Initially, we select the top K confident points for each category. Unlike conventional methods (Du et al., 2022), we exclusively employ features associated with categories present in the current image for prototype computation by utilizing category labels. This approach aids in eliminating prototype interference from irrelevant categories during subsequent feature augmentation. The formula for computing feature prototypes is:

$$\mathbf{c}_t = topk(p_t),\ p_t \in \Omega_p \tag{1}$$

$$\mathbf{v}_t = topk(f_t),\ f_t \in \Omega_f \tag{2}$$

$$fp_t = norm(\frac{\sum_i^K \mathbf{c}_{t,i}\mathbf{v}_{t,i}}{\sum_i^K \mathbf{c}_{t,i}}) \tag{3}$$

Here, t represents one of all categories. The sets $\Omega_p$and Ω_f respectively denote the collections of feature values and prediction values for all categories present in the current image. The feature prototype fp_t is the weighted average of feature embedding, normalized accordingly.

3.2.3. Definition and Update of Global Prototypes

When extracting prototypes solely from each batch, it limits the consideration of diversity within prototypes of the same category. This restriction confines the analysis to the current batch's image features, failing to effectively aid the model in comprehending new features. Hence, we introduced the concepts of global and local prototypes to distinguish prototypes at different stages. In contrast to local prototypes, global prototypes encompass prototypes for each category. Within each category, there exists a set of equally-sized local prototype vectors, managed using *Prototype Memory Bank*. The global prototypes start as empty. Throughout the training, local prototypes continually update the global prototypes. The update strategy is as follows: searching for the global prototype corresponding to the same category as the local prototype. If the global prototype is not full, it's directly assigned; if the current category's global prototype is full, the cosine similarity between the current local prototype and n prototypes in the global prototypes is computed, and the most similar one is updated. The update formula is as follows:

$$fp_{old} = selectmin(\frac{fp_t \cdot fp_i}{||fp_\mathbf{t}|| \cdot ||fp_i||}), fp_i \in \Omega_{fp_{global}} \tag{4}$$

$$fp_{new} = \alpha \cdot fp_{old} + (1-\alpha) \cdot fp_t, \alpha \in [\mathbf{0}, \mathbf{1}] \tag{5}$$

where fp_t represents the current local prototype, $\Omega_{fp_{global}}$ is the collection of global prototypes, $||\cdot||$ denotes the norm of a tensor, $\cdot$ represents the tensor dot product operation, $selectmin()$ selects the computed minimum value among fp_i as the prototype to be updated fp_{old}, and α is the hyperparameter for update speed.

3.3. Prototype-based Feature Augmentation

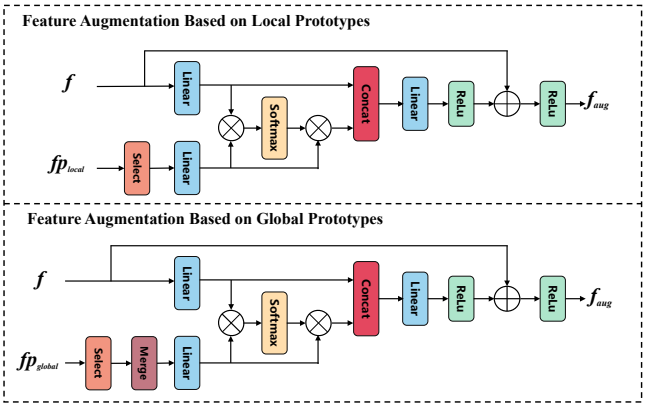


Figure 3. The structural diagram of the prototype-based feature augmenter. When the global prototypes are not fully updated, only the local prototypes are used to augment the features. Once the global prototypes are fully updated, both the local and global prototypes are used to augment the features.

3.3.1. Feature augmentation based on local prototypes

Inspired by the feature prototype method *FeatMatch* (Kuo et al., 2020) used in semi-supervised learning classification, we have introduced a feature-prototype-based classification approach into scribble-supervised semantic segmentation tasks. In semi-supervised classification tasks, prototypes are derived through clustering. During augmentation, prototypes for each category are involved, aiming to extract valuable information from labeled data and propagate it to unlabeled data. However, for our task, prototypes are extracted from high-confidence regions in the current image features. Our goal is to propagate valuable information from correctly classified pixels to those yet to be classified. Therefore, we believe other category prototypes aren't necessary for this augmentation process.

Feature augmentation based on local prototypes occurs within each batch. Therefore, when augmenting using local prototypes, it is essential to select the prototypes corresponding to the categories of the current features for reinforcement. In the aforementioned extraction phase, prototypes unrelated to categories present in the current image were not extracted, resulting in these category prototypes being zero-valued. To select the prototypes corresponding to categories within the current batch, we initially filter the local prototypes fp by defining $fp^* = select(fp)$, thereby removing irrelevant zero-value prototypes. Subsequently, for each feature f within the current batch and its corresponding prototype fp, we project them into an embedding space, obtaining $\mathbf{e}_f = linear(f)$ and $\mathbf{e}_{fp} = linear(fp)$. Then, we compute their attention weight ω, derived from the matrix multiplication of $\mathbf{e}_f$ and $\mathbf{e}_{fp}$, normalized using $softmax$. The formula is expressed as:

$$\omega = softmax(matmul(\mathbf{e}_f^T, \mathbf{e}_{fp})) \tag{6}$$

Then, weighting $\mathbf{e}_{fp}$ using attention weight ω . Next, concatenating the weighted prototype with image features and subjecting them to a linear layer transformation is performed. This process aids in propagating information from the prototype to the image features, thereby enhancing the features:

$$\mathbf{e}_f^* = ReLu(linear(concat(matmul(\omega, \mathbf{e}_{fp}), \mathbf{e}_f)) \tag{7}$$

Finally, the last feature is obtained by using residual connections between the initial features and the augmented features:

$$f_{aug} = ReLu(f + linear(\mathbf{e}_f^*)). \tag{8}$$

3.3.2. Feature augmentation based on global prototypes

As shown in the Figure 3, the overall process of feature augmentation based on global prototypes is similar to that based on local prototypes, with the main difference lying in the initial handling of prototypes. However, compared to local prototypes, global prototypes contain n ordinary prototypes for each category. Hence, before augmentation, it is necessary to perform a merge process on the global prototypes to facilitate subsequent computations.

3.3.3. The setting for augmentation

Considering that our prototypes are extracted from features, the quality of the extracted features might not be assured during the early stages of training. Consequently, relying solely on prototypes for feature augmentation might not yield optimal results. Therefore, we have established a warm-up period during which we refrain from utilizing prototypes for feature augmentation, instead employing basic loss constraints for training. Once this warm-up period concludes, we extract local prototypes from the features. These local prototypes contribute to enhancing the features of the current batch. Meanwhile, the global prototypes might still contain empty values due to ongoing updates. Hence, we have set a condition where the global prototypes do not partake in feature augmentation until they are completely filled. It is only after the global prototypes have been fully updated that we utilize them for augmentation.

3.4. Loss Function

For the design of the loss function, as shown in Figure 2, the overall loss function consists of two parts: partial cross-entropy loss $\mathcal{L}_{pce}$ and consistency loss $\mathcal{L}_{con}$. The consistency loss further includes the consistency loss between the initial predicted values and the predicted values augmented using local prototypes $\mathcal{L}_{total}$, as well as the consistency loss

between the initial predicted values and the predicted values augmented using global prototypes $\mathcal{L}_{global}$.

Partial cross-entropy loss is obtained between the predicted values and the scribble labels, and its expression is as follows (Tang et al., 2018b; Obukhov et al., 2019; Pan et al., 2021; Liang et al., 2022):

$$\mathcal{L}_{pce} = -\frac{1}{|\Omega_L|} \sum_{i \in \Omega_L} \sum_{c \in C} y_i^c log(p_i^c) \tag{9}$$

where Ω_L represents the set of all labeled pixels, C represents all the categories, y_i^c and p_i^c represent the ground truth and predicted values of category c at pixel i, respectively.

Consistency loss is employed to constrain the feature prediction values before and after augmentation. We utilize mean squared error (MSE) (Bauer & Kohavi, 1999) as the consistency loss function to restrict the relationship between these two prediction values, expressed as:

$$\mathcal{L}_{con} = -\sum_{i=1}^{h \times w} p_i log(p_i^{aug}) \tag{10}$$

Here, h and w represent the height and width of the predicted values, where p_i and p_i^{aug} denote the initial predicted value and the augmented predicted value at the same position, respectively.

Total loss varies at different stages of training. In the warm-up period, when neither the local prototype nor the global prototype is utilized, the overall loss is represented as: $\mathcal{L}_{total} = \mathcal{L}_{pce}$. During the phase when prototypes start being used but the global prototype hasn't been fully updated, only the local prototype contributes to augmentation. Hence, the overall loss at this stage is: $\mathcal{L}_{total} = \mathcal{L}_{pce} + \lambda_l \mathcal{L}_{con-l}$. Finally, when both the global and local prototypes are engaged in training, the overall loss is defined as: $\mathcal{L}_{total} = \mathcal{L}_{pce} + \lambda_l \mathcal{L}_{con-l} + \lambda_g \mathcal{L}_{con-g}$. Here, λ_l and λ_g represent the loss weights for $\mathcal{L}_{con-l}$ and $\mathcal{L}_{con-g}$ respectively.

4. Experiment

4.1. Dataset and Evaluation Metric

We use the PASCAL-Scribble Dataset, which was initially introduced by Lin et al. in ScribbleSup (Lin et al., 2016). The PASCAL-Scribble Dataset is a dataset with scribble annotations applied to the PASCAL VOC 2012 (Everingham et al., 2010). The PASCAL VOC 2012 dataset consists of 12,031 images with scribble annotations. The training set contains 10,582 images, and the validation set contains 1,449 images. The PASCAL VOC 2012 dataset has 21 categories, including 20 object categories and one background category. Unless otherwise specified, all ablation experiments are conducted on the PASCAL VOC 2012 dataset. The evaluation metric used is the mean Intersection-over-Union (mIoU).

4.2. Implementation Details

Our method consists of five main modules: encoder, decoder, prototype extraction, prototype updating, and feature augmentation. Given the efficient and excellent performance of the vision transformer in semantic segmentation tasks, we adopt the Mix Transformer proposed in Segformer (Xie et al., 2021), which is specifically optimized for semantic segmentation tasks and achieves superior performance compared to vanilla transformers. The backbone parameters of the model are initialized with ImageNet (Deng et al., 2009) pretrained weights, while the remaining parameters are randomly initialized. We utilize the AdamW optimizer with an initial learning rate of 3×10^{-5}. We employ a multi-step scheduler for learning rate decay during iterations, with a decay weight of 0.01. Additionally, the learning rate for other parameters is set to 10 times that of the backbone network. Regarding data preprocessing, all training images undergo random scaling (0.5 to 2), random rotation (-10 to 10 degrees), random flipping, and Gaussian blurring. Finally, the images are cropped to a size of 512×512. In Equation (5), the momentum of prototype updating α, is set to 0.99, and the number of global prototypes is set to 5. The batch size is set to 16. Our experimental code is based on the PyTorch framework (Paszke et al., 2019), and all experiments are conducted on an NVIDIA RTX 3090 GPU.

4.3. Comparison with State-of-the-Art Methods

As shown in Table 1, we compare our method with existing approaches and demonstrate significant improvements in PASCAL VOC 2012 *val* set. To ensure fairness, we chose MiT-B1 as the backbone, which achieved a score of 79.2% on fully supervised data, similar to other methods using ResNet101 (He et al., 2016) as the backbone. MiT-B1 is slightly lower than them. Of course, we can also choose larger MiT series backbones, but this would compromise the fairness of the comparison. Therefore, we will discuss this in detail in Section 4.4.

As shown in Table 1, we compare our method with other state-of-the-art approaches. Our method adopts a single-stage training framework, which does not require the use of additional supervised data during the training process and does not use CRF (Krähenbühl & Koltun, 2011). ScribbleSup (Lin et al., 2016) introduced scribble labels into the semantic segmentation field for the first time and achieved an mIoU of 63.1%. Methods based on the design principle of regularization loss guide pixel classification by extracting pairwise relationships from low-level image information. They also achieve good results, reaching up to 75.0%. BPG (Wang et al., 2019) and SPML (Ke et al., 2021) use

Table 1. Comparison with other state-of-the-art methods on PASCAL VOC 2012 *val* set.

Method	Backbone	Publication	Supervision	Single-stage	Extra Data	CRF	mIoU(%)
(1) DeeplabV2 (Chen et al., 2017)	VGG16	TPAMI'17	Full	√	-	√	71.6
(2) DeeplabV2 (Chen et al., 2017)	ResNet101	TPAMI'17	Full	√	-	√	77.7
(3) DeepLabV3+ (Chen et al., 2018)	ResNet101	ECCV'18	Full	√	-	-	80.2
(4) LTF (Song et al., 2019)	ResNet101	NeurIPS'19	Full	√	-	-	80.9
(5) Segformer (Xie et al., 2021)	MiT-B1	NeurIPS'21	Full	√	-	-	79.2
KernelCut Loss (Tang et al., 2018b)	(2)	ECCV'18	Point	-	-	√	57.0
A2GNN (Zhang et al., 2021)	(3)	TPAMI'21	Point	-	-	√	66.8
Seminar (Chen et al., 2021)	(3)	ICCV'21	Point	-	-	-	72.5
SPML (Ke et al., 2021)	(2)	ICLR'21	Point	√	√	√	73.2
TEL w. Seminar (Liang et al., 2022)	(4)	CVPR'22	Point	-	-	-	74.2
Box2Seg (Kulharia et al., 2020)	UperNet	ECCV'20	Bounding-box	-	-	√	76.4
BAP (Oh et al., 2021)	(2)	CVPR'21	Bounding-box	-	-	√	74.6
ScribbleSup (Lin et al., 2016)	(1)	CVPR'16	Scribble	-	-	√	63.1
NormCut Loss (Tang et al., 2018a)	(2)	CVPR'18	Scribble	-	-	√	74.5
DenseCRF Loss (Tang et al., 2018b)	(2)	ECCV'18	Scribble	-	-	√	75.0
KernelCut Loss (Tang et al., 2018b)	(2)	ECCV'18	Scribble	-	-	√	75.0
GridCRF Loss (Marin et al., 2019)	(2)	ICCV'19	Scribble	-	-	-	72.8
GatedCRF (Obukhov et al., 2019)	(3)	NeurIPS'19	Scribble	√	-	-	75.5
BPG (Wang et al., 2019)	(2)	IJCAI'19	Scribble	√	√	-	76.0
SPML (Ke et al., 2021)	(2)	ICLR'21	Scribble	√	√	√	76.1
URSS (Pan et al., 2021)	(2)	ICCV'21	Scribble	-	-	√	76.1
PSI (Xu et al., 2021)	(3)	ICCV'21	Scribble	√	-	-	74.9
Seminar (Chen et al., 2021)	(3)	ICCV'21	Scribble	-	-	-	76.2
A2GNN (Zhang et al., 2021)	(4)	TPAMI'21	Scribble	-	-	√	76.2
TEL (Liang et al., 2022)	(4)	CVPR'22	Scribble	√	-	-	77.3
AGMM (Wu et al., 2023)	(3)	CVPR'23	Scribble	√	-	-	76.4
CDSP (Zhang et al., 2024)	(3)	AAAI'24	Scribble	√	-	-	75.9
Ours-ResNet101	(2)	ICML'24	Scribble	√	-	-	76.2
Ours-MiT-B1	(5)	ICML'24	Scribble	√	-	-	**77.9**

edge detectors to assist semantic segmentation, but this requires additional data. URSS (Pan et al., 2021) and A2GNN (Zhang et al., 2021) also achieve good results, surpassing 76% mIoU, but they require multi-stage learning. AGMM (Wu et al., 2023) and CDSP (Zhang et al., 2024) leverage label diffusion methodologies to achieve notable performance. However, it is noteworthy that their respective mIoU values have not exceeded the threshold of 77%. TEL (Liang et al., 2022) is the best method in recent years, proposing a tree energy loss to guide the classification of unlabeled pixels, achieving an mIoU of 77.3%. Compared to the current state-of-the-art method TEL, despite our backbone network, MiT-B1, performing slightly weaker than theirs on the fully supervised dataset, our approach still achieves a 0.6% improvement in mIoU.

Table 2. Ablation studies of the components of our proposed method. "basic" means the basic result, "local" means the result augmented with local prototypes, and "global" means the result augmented with global prototypes. The $\mathcal{L}_{pce}$ and $\mathcal{L}_{pce}$ represent different $\mathcal{L}_{pce}$ in Figure 2, while the blue and red represent the top two results.

Method	$\mathcal{L}_{pce}$	$\mathcal{L}_{con-l}$	$\mathcal{L}_{con-g}$	mIoU(%) basic	local	global
Baseline	√			67.5	-	-
Ours	$\mathcal{L}_{pce}$	√		75.7	76.9	-
	$\mathcal{L}_{pce}$		√	76.0	-	77.4
	$\mathcal{L}_{pce}$	√	√	76.4	77.7	77.1
	$\mathcal{L}_{pce}$	√	√	76.8	77.3	77.9

4.4. Ablation Study and Analysis

In this section, we investigate all the operations discussed in Section 3. All training and validation are conducted on the Pascal VOC 2012 dataset.

Effectiveness of each component. First, we investigate the components of our method, which involve the usage of three loss functions. We employ the basic MiT-B1 as the backbone. We designate the use of only partial cross-entropy as the baseline for our method, and then conduct ablation studies on the methods with local prototype augmentation and global prototype augmentation. As shown in Figure 2, our method has three prediction maps, and all three prediction maps can generate results. When using only local prototypes, there are two prediction maps: the basic one and the one augmented with local prototypes.

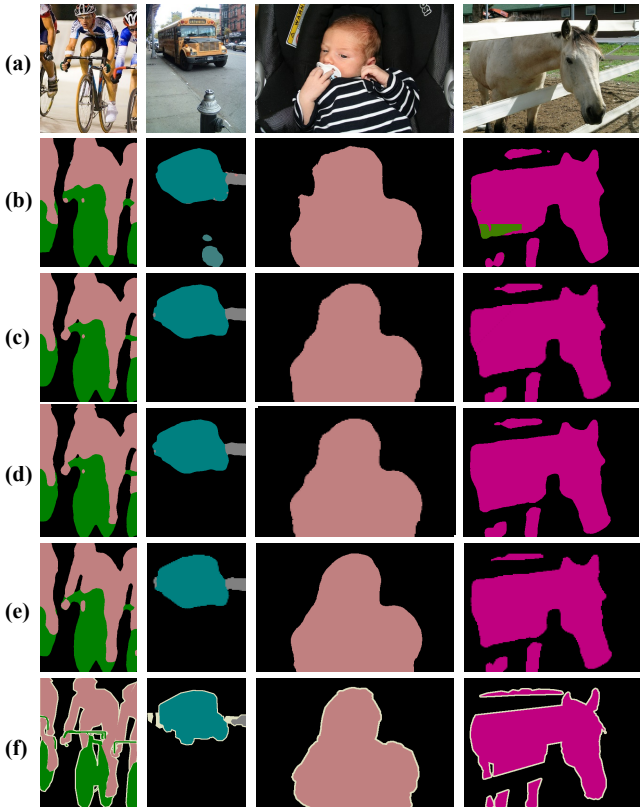


Figure 4. Visualizing the Components of Our Method through Ablation Experiments.(a) Input image. (b) Baseline (Only use partial cross-entropy). (c) Augmented only by local prototypes. (d) Augmented only by global prototypes. (e) Augmented by two types of prototypes. (f) Ground truth.

Similarly, when using global prototype augmentation, there are also two prediction maps: the basic one and the one augmented with global prototypes. As shown in Table 2, when using only local prototype augmentation, our method achieves a 9.4% mIoU improvement compared to the baseline. When using only global prototype augmentation, there is a 9.9% mIoU improvement. When both methods are used together, the best performance is achieved with a 10.4% mIoU improvement. Therefore, we select the simultaneous augmentation of both prototypes as our final method. In Section 3.4, due to the different choices of $\mathcal{L}_{pce}$ at different stages, when both augmentation methods are used, we also conduct ablation experiments on the two choices. The results show that the prediction map augmented with global prototype augmentation, guided by partial cross-entropy loss with scribble labels, achieves the best results. Additionally, we can observe that using both augmentation methods can significantly improve the basic prediction map.

As shown in Figure 4, we present the visualization results for different composition components. Compared to the baseline results, the results augmented with prototypes show significant improvements, especially in regions of other categories, such as the "fire hydrant" in the second image and the green area in the fourth image. Through prototype guidance, the interference from these other categories is weakened or eliminated. There is little difference between using only local prototypes and using only global prototypes, with the global prototypes slightly emphasizing details. Optimal results are achieved when both methods are used, as evidenced by the improvement in the hand region in the first image and the details of the occluded edges in the fourth image. For more detailed information, please refer to the supplementary.

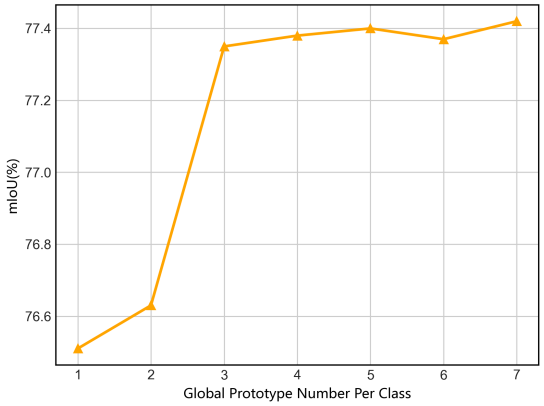


Figure 5. Impact of the number of prototypes contained in each class of global prototypes.

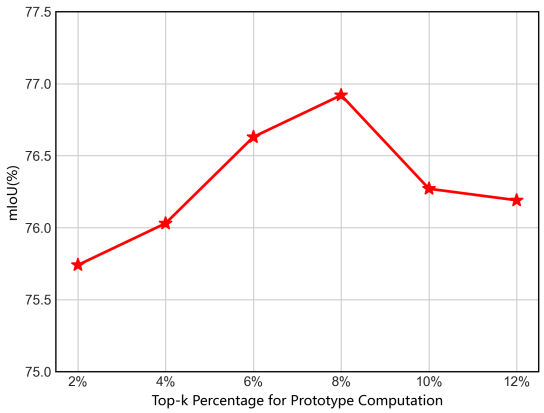


Figure 6. Impact of different *top-k* percentage values on the computation of prototype in Equations (1) and (2).

Setting of prototype. In our approach, the number of global prototypes represents the number of local prototypes that should be included for each class. To assess the impact of increasing the number of global prototypes on the results, we conducted experiments using only global prototypes for augmentation and not utilizing local prototypes. From Figure 5, it can be observed that as the number of prototypes increases to around 5, the increase in mIoU becomes satu-

rated. Therefore, we have set this value as the default in our method.

We also conduct tests on the *top-k* percentage involved in Equations (1) and (2) for prototype extraction. The fewer pixels involved in the computation indicate the use of more confident pixels for prototype extraction. However, if the percentage is too small, we may lose more useful information. Similarly, to evaluate the impact of varying the value of k on the performance, we conduct experiments using only local prototypes. As shown in Figure 6, our method performs better when the k percentage is 8%.

Table 3. Impact of the backbone.

Model	Backbone	Params	FLOPs	mIOU(%) full	mIOU(%) scribble
DeeplabV2	ResNet101	43.6M	75.2G	77.7	76.2
DeeplabV3+	ResNet101	60.8M	102.3G	80.2	78.1
LTF	ResNet101	91.7M	138.4G	80.9	78.2
Segformer	MiT-B1	13.6M	19.4G	79.2	77.9
Segformer	MiT-B3	44.6M	44.5G	81.9	79.8
Segformer	MiT-B5	82.0M	72.9G	**83.9**	**81.5**

Impact of backbone. In Section 4.3, for a fair comparison, we only compare our method with the state-of-the-art methods using MIT-B1. Here, we investigated the backbone of our method. We select the backbones commonly used in existing methods, such as DeepLabv2 (Chen et al., 2017), DeepLabv3+ (Chen et al., 2018) and LTF (Song et al., 2019) based on ResNet101 (He et al., 2016), as well as Segformer (Xie et al., 2021) based on a larger backbone network that performs better in fully supervised scenarios. As shown in Table 3, we present the results obtained in both fully supervised and scribble supervised settings using the Pascal VOC 2012 dataset, as well as the parameter counts and floating-point operation (FLOPs) of each model during inference. The table demonstrates that MiT-B1 Segformer achieves comparable performance to the ResNet101-based LTF despite its significantly lower parameter counts and computational complexity. And under the premise of using ResNet101 as backbone, our approach is superior to all current approaches. We observed that our method exhibits superior upper bound performance on the Transformer (77.9%/79.2%) compared to ResNet (78.2%/80.9%). This characteristic, coupled with its exceptional efficiency, constitutes a key factor in our selection of Segformer. When utilizing a model with a similar parameter count, MiT-B3 attains a mIoU score of 79.8%. Additionally, opting for MiT-B5 with the highest parameter count yields a peak mIoU score of 81.5%, significantly surpassing existing methods. However, due to the unfair advantage introduced by the backbone, it is not included in Table 1.

5. Conclusion

This paper introduces a prototype-based feature augmentation method for scribble supervision. We extract prototypes from the confident portion of the initial results provided by scribble supervision. By utilizing the extracted prototypes, we augment the initial features and employ different prototype strategies tailored to the specific setting of scribble supervision. The method utilizes generated prototypes from correctly classified pixels to guide the classification of misclassified pixels, resulting in improved prediction performance. Experimental results demonstrate that our method achieves state-of-the-art performance. In the future, we plan to apply our method to other tasks to harness its significant potential and application value.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This work is funded by the National Natural Science Foundation of China under Grant No.62272145 and No.U21B2016.

References

Bauer, E. and Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36:105–139, 1999.

Bearman, A., Russakovsky, O., Ferrari, V., and Fei-Fei, L. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pp. 549–565. Springer, 2016.

Chen, H., Wang, J., Chen, H. C., Zhen, X., Zheng, F., Ji, R., and Shao, L. Seminar learning for click-level weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6920–6929, 2021.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and 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*, 40(4):834–848, 2017.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Chen, Q., Yang, L., Lai, J.-H., and Xie, X. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4288–4298, 2022.

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Du, Y., Fu, Z., Liu, Q., and Wang, Y. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4320–4329, 2022.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338, 2010.

Gao, B.-B. and Zhou, H.-Y. Learning to discover multi-class attentional regions for multi-label image recognition. *IEEE Transactions on Image Processing*, 30:5920–5932, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ke, T.-W., Hwang, J.-J., and Yu, S. X. Universal weakly supervised segmentation by pixel-to-segment contrastive learning. *arXiv preprint arXiv:2105.00957*, 2021.

Krähenbühl, P. and Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

Kulharia, V., Chandra, S., Agrawal, A., Torr, P., and Tyagi, A. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In *European Conference on Computer Vision*, pp. 290–308. Springer, 2020.

Kuo, C.-W., Ma, C.-Y., Huang, J.-B., and Kira, Z. Featmatch: Feature-based augmentation for semi-supervised learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pp. 479–495. Springer, 2020.

Lee, J.-H., Kim, C., and Sull, S. Weakly supervised segmentation of small buildings with point labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7406–7415, 2021.

Liang, Z., Wang, T., Zhang, X., Sun, J., and Shen, J. Tree energy loss: Towards sparsely annotated semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16907–16916, 2022.

Lin, D., Dai, J., Jia, J., He, K., and Sun, J. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3159–3167, 2016.

Liu, S., Zhang, L., Yang, X., Su, H., and Zhu, J. Query2label: A simple transformer way to multi-label classification. *arXiv preprint arXiv:2107.10834*, 2021.

Marin, D., Tang, M., Ayed, I. B., and Boykov, Y. Beyond gradient descent for regularized segmentation losses. In *Computer Vision and Pattern Recognition*, 2019.

Obukhov, A., Georgoulis, S., Dai, D., and Van Gool, L. Gated crf loss for weakly supervised semantic image segmentation. *arXiv preprint arXiv:1906.04651*, 2019.

Oh, Y., Kim, B., and Ham, B. Background-aware pooling and noise-aware loss for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6913–6922, 2021.

Pan, Z., Jiang, P., Wang, Y., Tu, C., and Cohn, A. G. Scribble-supervised semantic segmentation by uncertainty reduction on neural representation and self-supervision on neural eigenspace. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7416–7425, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Song, L., Li, Y., Li, Z., Yu, G., Sun, H., Sun, J., and Zheng, N. Learnable tree filter for structure-preserving feature transform. *Advances in neural information processing systems*, 32, 2019.

Tang, M., Djelouah, A., Perazzi, F., Boykov, Y., and Schroers, C. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1818–1827, 2018a.

Tang, M., Perazzi, F., Djelouah, A., Ben Ayed, I., Schroers, C., and Boykov, Y. On regularized losses for weakly-supervised cnn segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 507–522, 2018b.

Tang, Q., Liu, C., Liu, F., Liu, Y., Jiang, J., Zhang, B., Han, K., and Wang, Y. Category feature transformer for semantic segmentation. *arXiv preprint arXiv:2308.05581*, 2023.

Vernaza, P. and Chandraker, M. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7158–7166, 2017.

Wang, B., Qi, G., Tang, S., Zhang, T., Wei, Y., Li, L., and Zhang, Y. Boundary perception guidance: A scribble-supervised semantic segmentation approach. In *IJCAI International joint conference on artificial intelligence*, 2019.

Wu, L., Zhong, Z., Fang, L., He, X., Liu, Q., Ma, J., and Chen, H. Sparsely annotated semantic segmentation with adaptive gaussian mixtures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15454–15464, 2023.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

Xu, H., Liu, L., Bian, Q., and Yang, Z. Semi-supervised semantic segmentation with prototype-based consistency regularization. *Advances in Neural Information Processing Systems*, 35:26007–26020, 2022.

Xu, J., Zhou, C., Cui, Z., Xu, C., Huang, Y., Shen, P., Li, S., and Yang, J. Scribble-supervised semantic segmentation inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15354–15363, 2021.

Yoon, S.-H., Kweon, H., Jeong, J., Kim, H., Kim, S., and Yoon, K.-J. Exploring pixel-level self-supervision for weakly supervised semantic segmentation. *arXiv preprint arXiv:2112.05351*, 2021.

Yu, C., Shao, Y., Gao, C., and Sang, N. Condnet: Conditional classifier for scene segmentation. *IEEE Signal Processing Letters*, 28:758–762, 2021.

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

Zhang, B., Xiao, J., Wei, Y., Sun, M., and Huang, K. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12765–12772, 2020.

Zhang, B., Xiao, J., Jiao, J., Wei, Y., and Zhao, Y. Affinity attention graph neural network for weakly supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8082–8096, 2021.

Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., and Ding, E. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6798–6807, 2019.

Zhang, X., Zhu, L., He, H., Jin, L., and Lu, Y. Scribble hides class: Promoting scribble-based weakly-supervised semantic segmentation with its class label. *arXiv preprint arXiv:2402.17555*, 2024.

A. More Technical Details

A.1. More Details of Encoder and Decoder

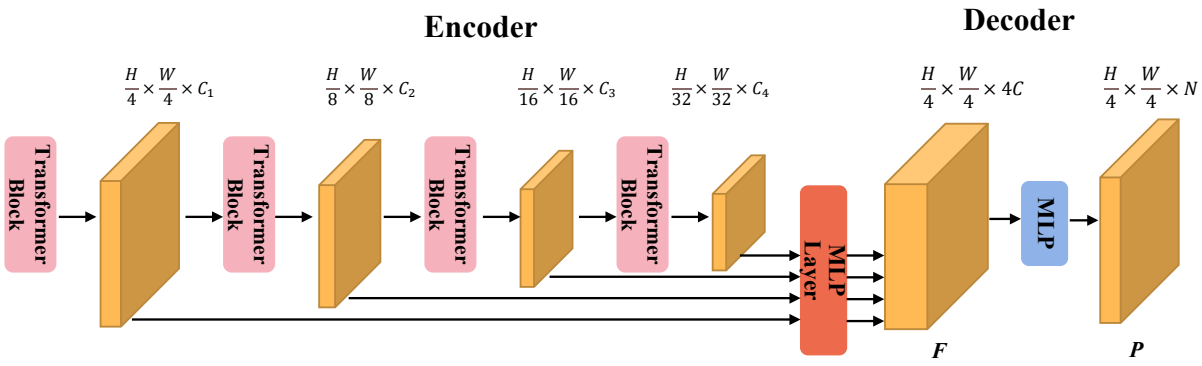


Figure 7. The overall framework of encoder and decoder.

As shown in Figure 7, the structure of our encoder and decoder utilizes the Segfomer (Xie et al., 2021) based on MiT-B1, and its efficiency is one of the main reasons why we chose it. We select the features from four levels, which are fused by the MLPLayer, as our feature map F because it can contain information from multiple levels. We use the feature map F and the prediction map P as the basis for prototype extraction. The features augmented through prototype refinement are then passed through a feature augmenter to maintain the same size as F.

A.2. More Details of Feature Augmenter and Inference

During feature extraction, we extract feature prototypes from the feature values corresponding to the top K values of each category contained in the prediction of the current image. Due to the uncertainty of predictions, it is inevitable to extract prototypes from other misclassified categories. In this case, when using only local prototype augmentation, although the prototypes are extracted from the current predicted category, they only reinforce the probability of misclassification for those misclassified images, leading to worse prediction performance. When using global prototypes, we retrieve global prototypes from the *prototype memory bank* for augmentation. Global prototypes include prototypes for each category, and if we do not select the prototypes of the correct category contained in the current image, it will also increase the chances of model misjudgment, resulting in counterproductive effects. Therefore, during the training process, we use category labels to filter the prototypes and select out those irrelevant prototypes to better utilize the effects of prototypes. The scribble labels contain category information, so we can easily obtain category labels from the scribble labels. Therefore, these can be achieved quite well during the training process, but they pose challenges during the inference process.

As depicted in Table 4, employing the correct class labels to guide prototype augmentation yields a mIoU score of 80.4%, surpassing even the predictions of MiT-B1 on the fully supervised dataset. However, this approach violates inference rules, as during inference, we can only access images without direct class label information. To address this, our initial strategy involves filtering preliminary prediction maps of unused prototypes by setting a threshold, because these pixels with small quantities are often misclassified. Experimenting with various thresholds, as shown in Table 4, the optimal result occurs with a threshold of 500, resulting in a mIoU of 75.7%. Nevertheless, this is lower than the 76.8% mIoU achieved without prototype augmentation. Consequently, we incorporate the classification results of established multi-label classification methods, Q2L (Liu et al., 2021) and MCAR (Gao & Zhou, 2021), attaining respective mAP scores of 96.6% and 94.3% on the Pascal VOC 2012 *val* dataset. Utilizing their classification results, we achieve mIoU scores of 77.9% and 76.2%, respectively. This underscores the influence of superior classification results on enhancing prototype augmentation. Conversely, incorrect classification results can misguide the model's understanding of semantic information by employing prototypes from incorrect categories, resulting in inferior performance. Given the potential for improving segmentation results through category labels, our future work will focus on refining the guidance of the prototype and incorporating this approach into multimodal tasks.

Table 4. The impact of class labels during inference.

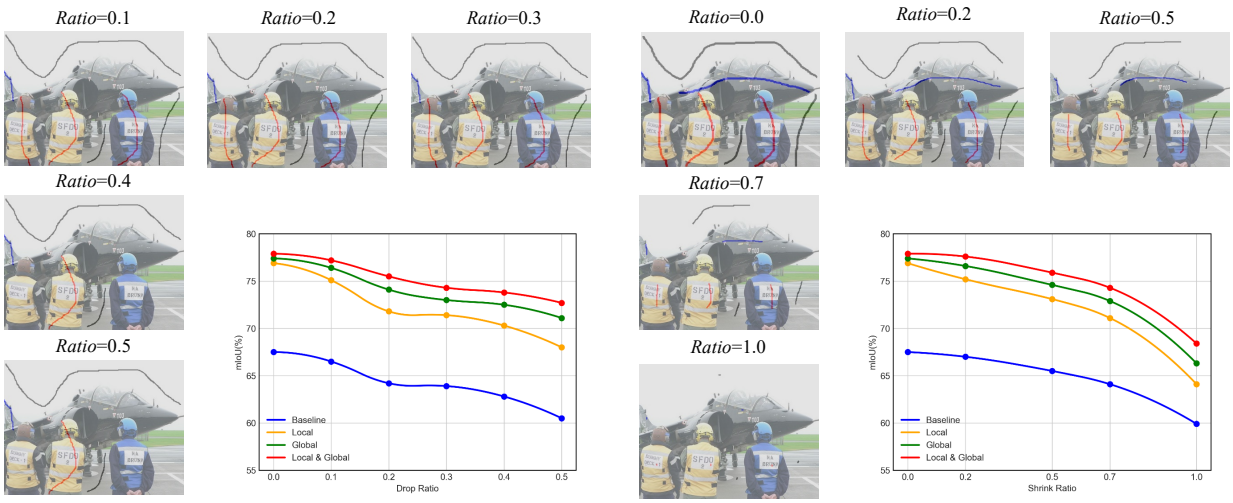
Method	Num	mAP(%)	mIoU(%)
Category Labels	-	-	80.4
Without Augmented	-	-	76.8
Q2L(Liu et al., 2021)	-	96.6	77.9
MCAR(Gao & Zhou, 2021)	-	94.3	76.2
Prediction Map Filtering	0	-	74.1
	100	-	75.2
	500	-	75.7
	1000	-	74.3

Table 5. Impact of the weights of loss terms.

	λ_l	λ_g	mIoU(%)
default	0.01	0.01	**77.9**
	0.005		77.7
	0.02		77.2
	0.05		76.8
		0.005	77.4
		0.02	77.6
		0.05	77.1

B. More Experimental Results

B.1. Ablation Study



(a) scribble-drop (b) scribble-shrink

Figure 8. The experiments on scribble-drop and scribble-shrink dataset with different drop or shrink ratios.

The form of Scribble. Since doodling can vary greatly in style from person to person, it is inherently subjective. Therefore, it becomes imperative to conduct robustness tests on the model with different degrees of drop and shrink ratios. In Figure 8, we present the results of our method's ablation experiments using scribble labels with varying degrees of drop and shrink. The results clearly demonstrate that, as the drop rate and shrink ratio increase, the model's performance declines. The method based on global prototypes exhibits relatively greater stability compared to the method employing local prototypes. However, overall, the model performs well even when the scribble annotations are reduced to mere dots.

Weight of the loss. In Table 5, we conducted an ablation study on the weighting of loss functions. We adjusted the weight ratios between different loss functions. Through this study, we determined the optimal weighting configuration for the loss functions as $\lambda_l = 0.01$, $\lambda_g = 0.01$.

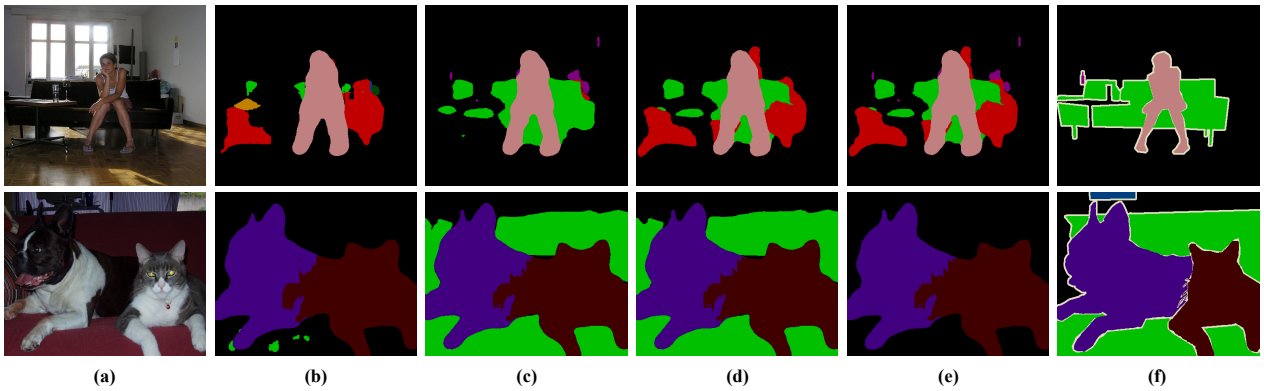


Figure 9. Failure cases of the proposed method on Pascal VOC *val* dataset. (a) Input image. (b) **Baseline (Inference without prototype augmentation).** (c) Inference with using category labels (d) Inference with using Q2L results. (e) Inference with using MCAR results. (f) Ground truth.

B.2. Failure Cases and Analysis

Figure 9 illustrates failed cases, offering a more intuitive understanding of the impact of prototype augmentation. To deepen the analysis, we present four types of outcomes: results without prototype augmentation, inference based on category labels, and inference utilizing two classification results. In the upper section of Figure 9, without prototype augmentation, the model identifies various semantic categories, including 'person', 'chair', 'sofa', 'table', 'plant', 'boat', among others, with many misclassifications. Given that the category labels only specify 'person', 'sofa', and 'bottle', they direct the model to diminish the representation of other categories in the initial prediction, especially those on the prediction periphery like 'table' and 'plant', which vanish after augmentation. Conversely, the category 'bottle', absent in the initial results, emerges after being guided by the prototype. Notably, although the category label does not contain 'chair', a small portion is retained after prototype augmentation. In the Q2L (Liu et al., 2021) and MCAR (Gao & Zhou, 2021) classification results, 'person', 'chair', 'sofa', and 'person', 'chair', 'sofa', 'bottle' are respectively identified. The segmentation results also align with the classification results, weakening the performance of other categories. In the lower section of Figure 9, the results without prototype augmentation include three categories: 'cat', 'dog', and a minimal representation of 'sofa'. Both category labels and Q2L classification include 'sofa', leading to a significant increase in the sofa's representation after prototype augmentation. However, as MCAR's classification does not involve 'sofa', the remaining small representation of 'sofa' disappears after prototype augmentation. Overall, the prototype augmented the model's understanding and inference capabilities of complex semantics by guiding the identification of edge-class categories using category labels to refine initial results.

B.3. More Quantitative Results

As shown in Table 6, we also present the data comparison of our method with other methods on each category of Pascal VOC 2012 *val* set. Our approach has yielded the best results in most categories.

Table 6. Per-class comparison between our approach and others on PASCAL VOC 2012 *val* dataset.

method	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	Mean
KernelCut (Tang et al., 2018b)	-	86.2	37.3	85.5	69.4	77.8	**91.7**	85.1	**91.2**	38.8	85.1	55.5	85.6	**85.8**	81.7	84.1	61.4	84.3	43.1	81.4	74.2	75.0
BPG (Wang et al., 2019)	93.4	84.8	**38.4**	84.6	65.5	78.8	91.4	**85.9**	89.5	41.0	87.3	58.3	84.1	85.2	**83.7**	83.6	64.9	**88.3**	46.0	86.3	73.9	76.0
SPML (Ke et al., 2021)	-	89.0	**38.4**	86.0	**72.6**	77.9	90.0	83.9	91.0	40.0	**88.3**	57.7	**87.7**	82.8	79.1	**86.5**	57.1	87.4	50.5	81.2	76.9	76.1
Ours-MiT-B1	**93.9**	**89.7**	35.7	**87.6**	69.2	**84.8**	90.3	84.7	89.9	**43.2**	87.9	**60.7**	87.1	82.6	80.6	**86.5**	**71.2**	82.5	**54.3**	**89.0**	**84.3**	**77.9**

B.4. More Qualitative Results

In Figure 10, we present the visualization results of our method, along with the baseline and the SOTA methods. From the images, it can be observed that the SOTA method is able to identify certain regions. However, the identified regions are relatively small. This is where the advantage of our method comes into play. With the guidance of the prototype, our method

can guide the segmentation of other pixels, resulting in superior segmentation results in the boundary regions. For instance, in the first image, the ears of the cow; in the second image, the 'chair' region; in the third image, the 'bottle' region; and in the fourth image, the 'sofa' region. These areas effectively demonstrate the guiding role of the prototype, as they direct the classification of other pixels and augment the classification performance in the boundary regions.

We have provided more visualization results of different composition components in Figure 11.

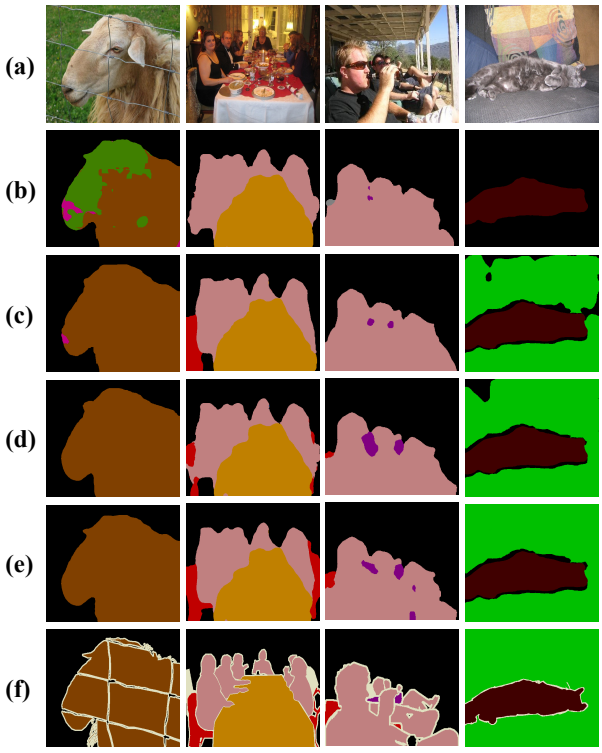


Figure 10. Visualization of our method and SOTA. (a) Input image. (b) Baseline (Inference without prototype augmentation). (c) URSS (Pan et al., 2021). (d) TEL(Wu et al., 2023). (e) Ours. (f) Ground truth.

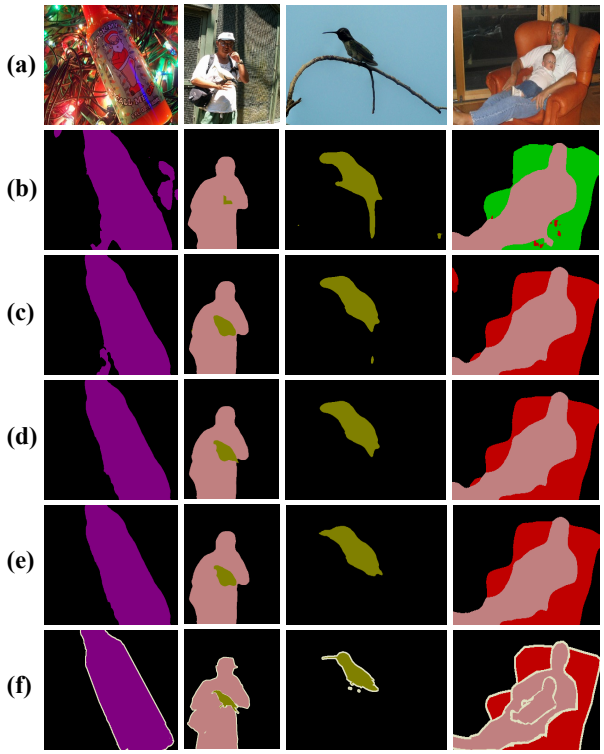


Figure 11. Visualizing the Components of Our Method through Ablation Experiments. (a) Input image. (b) Baseline (Only use partial cross-entropy). (c) Augmented only by local prototypes. (d) Augmented only by global prototypes. (e) Augmented by two types of prototypes. (f) Ground truth.