Learning Few-shot Segmentation from Bounding Box Annotations

Byeolyi Han*
Georgia Tech
Atlanta, Georgia, USA
bhan67@gatech.edu

Tae-Hyun Oh
Dept. of EE, POSTECH
Pohang, Korea
taehyun@postech.ac.kr

Abstract

We present a new weakly-supervised few-shot semantic segmentation setting and a meta-learning method for tackling the new challenge. Different from existing settings, we leverage bounding box annotations as weak supervision signals during the meta-training phase, i.e.*, more label-efficient. Bounding box provides a cheaper label representation than segmentation mask but contains both an object of interest and a disturbing background. We first show that meta-training with bounding boxes degrades recent few-shot semantic segmentation methods, which are typically meta-trained with full semantic segmentation supervisions. We postulate that this challenge is originated from the impure information of bounding box representation. We propose a pseudo trimap estimator and trimap-attention based prototype learning to extract clearer supervision signals from bounding boxes. These developments robustify and generalize our method well to noisy support masks at test time. We empirically show that our method consistently improves performance. Our method gains 1.4% and 3.6% mean-IoU over the competing one in full and weak test supervision cases, respectively, in the 1-way 5-shot setting on Pascal-5^i.*

1. Introduction

The semantic segmentation task aims to cluster pixel regions within an image according to semantic similarity. It is a fundamental visual scene understanding technique in computer vision and its applications [12]. By virtue of the advance of convolutional neural networks, the performance of semantic segmentation has been significantly improved against hand-crafted designs [23]. Nonetheless, there are two remaining challenges toward ultimate generic intelligence for scene understanding. First, the neural network is data-hungry [3]. Furthermore, obtaining high-quality segmentation labeling is far more costly than that of image-level annotations.[1] Second, the standard semantic segmentation task deals with pre-defined classes only, *i.e.*, a closed-set problem. However, there are lots of unseen or uncertain object classes in the real-world scenario, and these may more critically affect the success of systems subsequent to the scene understanding. Naïvely increasing the diversity of classes with high-quality segmentation labels is not a workaround and simply impossible due to an unlimited number of semantic classes in the real world [30].

The advance of few-shot learning (FSL) can deal with these challenges. FSL strives to train or adapt a model to target tasks, *e.g.*, classification and segmentation, with only a few samples. To generalize to the few-shot test with novel classes, few-shot learners are typically meta-trained by solving synthesized few-shot test episodes, *i.e.*, episodic learning [32]. Many few-shot segmentations [8, 22, 34, 36] also follow the same scheme. In the previous works, an episode is composed of a support set and query set with those segmentation annotations. Then, few-shot segmentation methods are trained to segment the query set given the support set. However, the phrase "a few" might be misleading in the annotation-efficiency perspective. While it is true that few-shot segmentation requires a few {image, segmentation mask} pairs during test time, the same level of large-scale full segmentation annotations are still required during meta-training to mimic the test time episodes. This hardly reduces the necessity of costly annotations.

Based on the aforementioned observations, in this work, we present a new weakly meta-training method for *few-shot semantic segmentation from bounding box annotations*, which has been under-explored before. Recently, different weakly-supervised few-shot semantic segmentation tasks have been proposed [27, 30, 34, 39]. They utilize weak labels during the inference phase, but a large number of segmentation masks are still used during the meta-training stage. On the contrary, we focus on addressing the overloaded la-

*This work has been done when she was a visiting researcher at POSTECH.

[1] While image-level labeling takes less than a second per image by non-expert subjects, semantic segmentation labeling takes more than 1.5 hours per image by trained experts even with an efficient polygon-based annotation tool [7].

beling cost during *meta-training* with a large-scale weakly supervised dataset. The meta-training stage requires a much higher number of segmentation masks than the inference stage. Since the segmentation labels are particularly costly to annotate, replacing the segmentation annotation with the weak one during meta-training may reduce the significant amount of annotation cost. This, therefore, enables low-cost learning than the prior arts in terms of annotation load.

In particular, we leverage bounding boxes as weak supervision during meta-training. In the weakly-supervised field, the commonly used weak labels are image-level labels. However, recent research [6] pointed out the ill-posedness of weak-supervision-based localization problems. Without localization information, if the class information is more correlated with background information than the object of interest, neural networks are likely to focus on background information, consequently leading to failed localization. That is, image-level labels may not be sufficient for obtaining sufficient supervision signals, especially in our challenging few-shot learning setup. On the contrary, bounding boxes require much less effort for annotating than segmentation masks, and contain necessary localization information for semantic segmentation [15]; thus, a good compromise between image-level and segmentation labels. [2]

However, directly leveraging bounding boxes perturbs learning few-shot segmentation. We experimentally show the segmentation performance on novel classes is degraded in a prototype learning scenario with bounding boxes during meta-training. We posit the cause of performance degradation stems from the background pixels included in bounding boxes. The contaminated information propagates through both support prototypes and query labels, which results in worse performances. Hence, we propose the pseudo trimap estimator and the trimap-attention based prototype learning, which exclude the uncertain regions within bounding boxes from learning, to deal with the noise injected by bounding boxes during meta-training. With extensive experiments, we found our method consistently enhances the few-shot semantic segmentation performance in various settings. This demonstrates our method effectively purifies bounding boxes and learns more accurate prototypes during meta-training. Furthermore, since our method suggests a weakly-supervised meta-training scheme, our model can be adapted to both fully- and weakly-supervised testing settings.

2. Related Work

Few-shot learning (FSL). FSL aims to learn from few-shot samples. In order to test meta-learning ability, in FSL we test on an unseen domain, *i.e.*, train domain and test domain are disjoint. While there are several categories of FSL such as metric-based approach [17, 31, 32], optimization-based approach [11], and model-based approach [37], we concentrate on metric-based methods and episodic learning, a common learning scheme to train metric-based few-shot learners in this paper. Episodic training reforms the dataset into episodes and feeds episodes to neural networks iteratively. In each episode, the few-shot learner encounters different support classes and is encouraged to perform a task on query images based on a support set. By mimicking the test stage during meta-training, the few-shot learner gets to have meta-learning ability without overfitting. Matching network [32] and prototypical network [34] facilitate episodic learning for FSL. Matching network consists of a feature encoder and a prototype extractor for estimating prototypes, *i.e.* class representatives. In the meta-training stage, support images for each class are fed into the prototype extractor to create a prototype containing information of all examples, while query images are guided to be closer to the corresponding prototype on the feature space. In the test time, the class is predicted as the class with the nearest prototype. Prototypical network uses a feature encoder to both encode features and generate prototypes, specifically, as the average of the features of each class image. Even with a simpler design, it shows improved performances.

Few-shot semantic segmentation. Few-shot semantic segmentation aims to perform segmentation on unseen domains. Matching-based methods [21, 29, 33, 39, 40] guide neural networks to predict pixel-to-pixel correspondence between support and query images so that on novel classes, the pixels with high correspondence values to support foreground regions are assumed to belong to foreground regions. Due to its very dense correspondence estimation, however, these methods typically require heavy computation and consequently are hard to be applied on multi-way multi-shot settings.

Prototype-based methods are in line with Prototypical network [31] in FSL literature. In these methods, prototypes are learned and every pixel is classified based on the distance between prototypes and its feature. Wang *et al.* [34] suggest PANet, a simpler yet effective prototype learning based on late-fusion [26]. Yang *et al.* [36] introduce an expectation-maximization algorithm to extract multiple prototypes per class for detailed class representation. Liu *et al.* [22] develop a graph attention module and apply the superpixel algorithm on an unlabelled set to get part-aware prototypes.

Recently, Cermelli *et al.* [4] suggest an incremental few-shot semantic segmentation (iFSS). iFSS deals with the incremental setting in few-shot semantic segmentation, *i.e.*, it aims to extend a pretrained model with new classes from few annotated images and without access to old training data. Since our work does not need additional training on novel classes, one model trained by our method can perform se-

[2] According to [1], average annotation cost per image from Pascal-VOC dataset is 20, 38.1 and 239.7 seconds for an image-level class, a bounding box and a full segmentation mask, respectively. That says, using bounding boxes instead of segmentation masks reduces the labelling cost 6 times.

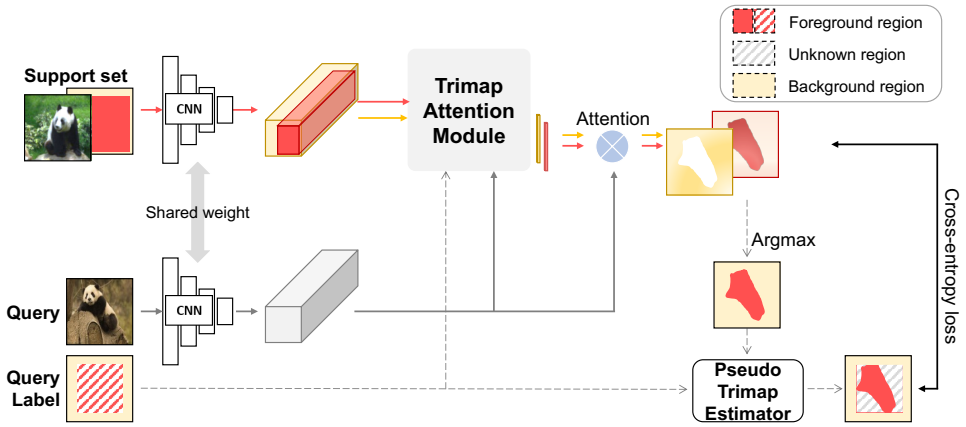

Figure 1: Overview of our model. We aim to learn few-shot semantic segmentation from bounding box annotations. In each train episode, support masks and query labels are all replaced by bounding boxes. We propose a pseudo trimap estimator and a trimap attention module to get a robust prototype from impure information and exclude uncertain regions in loss calculation.

mantic segmentation on both train classes and novel classes. Thus, our work can be seen as extended usage of iFSS with the utilization of bounding box labels during meta-training.

Weakly-supervised few-shot segmentation. Weakly-supervised learning (WSL) aims to learn signals from weak labels. From rough information, it tries to induce strong signals related to the location of objects. In computer vision domains, it usually refers to learning localization from image-level class labels, while bounding boxes, scribbles, and word embeddings can also work as weak labels.

There are several works which extract initial masks from bounding boxes by GrabCut [28] and refine from them to enable weakly-supervised segmentation from bounding boxes. For example, Kulharia *et al.* [18] predict the per-class attention map to focus on foreground pixels and refine boundaries. Ji *et al.* [16] train the class-wise CNN to better capture the class-wise shapes across all bounding boxes from the same class. Adapting those directly to few-shot segmentation is not straightforward due to the nature of the few-shot literature, which aims to remove class dependencies to avoid overfitting to any set of (train) classes and generalize well to unseen classes. Different than [16, 18], we suggest another way to make our method cooperate with GrabCut algorithm (see Table 5). For more details on WSL, refer to [14].

Given the potential of WSL, weakly-supervised few-shot semantic segmentation tasks have been proposed [27, 30, 34, 39]. In these tasks, weak labels including image-level class labels [27], bounding boxes [34, 39], scribbles [34], and word embeddings [30] of novel classes are used as supervision during *the test stage*. However, a large number of segmentation masks are still used during meta-training. Our work differs from them in that no segmentation mask is used during meta-training.

Concurrent to few-shot segmentation and weakly-supervised segmentation works, partially-supervised instance segmentation (PSIS) has been introduced [2, 10]. Also, in [35], BoxCaseg, a solution for box-supervised class-agnostic instance segmentation has been proposed. Both focus on solving instance segmentation by benefiting from bounding boxes, while they differ from our setting in three folds: (1) the problem scope; that instance and semantic segmentation have different challenges and therefore, require different network architectures and performance metrics, (2) that none of them solely focuses on few-shot settings, and (3) that they deal with different data regimes from ours, where we specifically focus on the scarce data regime that just few weak-supervised data is only available.

Relationship to our setting. To our knowledge, our work is the first attempt to enhance label efficiency by utilizing only weak labels during meta-training in the few-shot segmentation works. Since our method uses weak supervision during meta-training, we treat the performance when meta-trained in a fully supervised manner as the upper bound of ours.

Our method consists of an iterative scheme to refine bounding box labels and generate more accurate prototypes and update the neural network based on refined labels. While different iterative schemes have been proposed in [34, 36, 39], their motivations are completely different. CANet [39] and PMM [36] conduct iterative refinement of feature maps and iterative clustering for part prototype estimation, respectively, but they made no interaction between query and support sides. PANet [34] is not an iterative method, but it interacts with prototypes and features of query and support samples for symmetric regularization in both query and support branches. Stemming from different motivation, our refinement scheme is novel compared to the prior few-shot segmentation methods.

3. Method

We aim to learn semantic segmentation from a few annotated samples at test time. To this end, we meta-train our segmenter on classes $\mathcal{C}_{train}$ and evaluate on unseen classes $\mathcal{C}_{test}$. For convenience, we suppose to use two disjoint and non-overlapping datasets, $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$, which have annotated samples from $\mathcal{C}_{train}$ and $\mathcal{C}_{test}$, respectively.

To learn a few-shot segmenter generalizable to unseen scenarios, we follow the commonly used episodic-style mini-batch configuration, *i.e.*, episodic learning [29, 32]. An episode is synthesized to mimic test-time tasks so that the few-shot learner can be meta-trained to adapt to new tasks during test time. A C-way K-shot episode consists of a support set $\mathcal{S}$ and a query set $\mathcal{Q}$ as $\mathcal{E} = \{\mathcal{S}, \mathcal{Q}\}$, where the prior weakly-supervised few-shot methods [29] compose

$$\mathcal{S} = \bigcup_{c=1}^{C} \{ (I_{c,k}, M_{c,k}) \,|\, k = \{1, \cdots, K\} \}, \;\; \mathcal{Q} = \{ (I_q, M_q) \},$$

$I_{c,k}$ denotes the k-th support image of class c and $M_{c,k}$ its corresponding label. Each episode defines a few-shot segmentation task for a certain class combination that targets to classify every pixel in the query image as a semantic class $c \in \{0, \cdots, C\}$ well using the information given in its support set, where $c = 0$ defines the background class.

Our problem setting differs from the previous weakly-supervised few-shot segmentation work. Previously, full supervision of segmentation masks are provided as annotations, $M_{c,k}$ and M_q, for meta-training in both support and query sets. On the contrary, we use cheaper bounding box annotations as weak supervision signals in both support and query sets; thus, in our problem, $M_{c,k}$ and M_q now contain simple rectangular masks of class-of-interest instead of free-form segmentation, which makes our setting annotation efficient but more challenging.

During meta-training, at each step i, a training episode $\mathcal{E}^i_{train}$ is composed of randomly sampled class combination $\mathcal{C}_i \subset \mathcal{C}_{train}$. Throughout training steps, our model experiences various combinations of classes from $\mathcal{C}_{train}$ so that our neural network learns a general example-based segmentation strategy without class dependencies. This learning strategy is known to encourage preventing overfitting and yielding better generalization to unseen classes.

During testing, test episodes are composed similarly. A test episode $\mathcal{E}^j_{test} = \{\mathcal{S}_j, \mathcal{Q}_j\}$ is sampled from $\mathcal{D}_{test}$ at each test step j. Given the support set $\mathcal{S}_j$, segmentation masks of query images in $\mathcal{Q}_j$ are inferred in an episodic manner. Our model is evaluated based on segmentation performance over these test episodes.

Prototype learning. Our goal is to learn a neural network that extracts feature maps, where features at each location are expected to be separated according to the semantic meaning and to be generalized for new semantic objects. We denote the convolution neural network-based encoder $f_\theta : I \rightarrow F_{I;\theta}$, where $F_{I;\theta} \in \mathbb{R}^{H \times W \times d}$ denotes pixel-wise features with dimension d extracted from an RGB image $I \in \mathbb{R}^{H \times W \times 3}$, H and W the spatial dimensions of I. We follow the standard architecture of semantic segmentation based on fully convolutional layers [23] and dilated convolutions [5, 38] as used in [34]. This facilitates obtaining a pixel-wise feature map with a larger receptive field.

We first compute feature maps for images in the support and query sets, denoting $F_{c,k}{=}F_{I_{c,k};\theta}$, $F_q{=}F_{I_q;\theta}$, respectively. With this feature map, typical segmentation is implemented by measuring correlations between a feature of each pixel and a classification weight. In prototype-based few-shot methods, the classification weight is adaptively predicted from few-shot examples in the support set, called prototype [31]. We estimate the mean of a class distribution, *i.e.* prototypes for each class in the support set. By learning prototypes and neural networks to contrast positive and negative pairs of features and prototypes, the small distance between a class prototype and a feature is induced to have high likelihoods of pixel segmentation. Accordingly, we classify each pixel in a query image to the class with the nearest prototype from its feature. We use cosine distance as a distance metric as suggested in [25, 34].

3.1. Joint prototype and segmentation refinement

In prior arts, a prototype from a support sample is estimated by averaging the positive features, *e.g.*, global average pooling [31] or masked average pooling [26, 34]. However, in our problem setting, we observed that these simple pooling-based methods are detrimental to the quality of the learned feature map and prototype both. This quality degradation stems from impure information inside a bounding box; bounding box representation possesses foreground pixels as well as background ones. Due to this, prototypes estimated conventionally become less discriminative, and this leads to degrading the quality of learned feature maps. This motivates us to develop an alternating refinement method to distill segmentation labels and to denoise foreground prototypes so that robust prototypes and feature maps against the weak label noise can be learned.

For robust meta-training, we propose an expectation-maximization [24] like alternating method to refine both prototypes and segmentation labels. We first estimate an initial support prototype and predict initial segmentation for a given query sample using the initial prototypes. Given these initial estimates of prototype and segmentation, our refinement is applied to jointly improve prototypes and segmentation predictions of both query and support samples.

Specifically, to mitigate noisy supervision by weak bounding box labels in the alternating framework, we also propose a pseudo trimap representation and its simple estimation method. More distinguishably, since both query and support

labels are noisy in our problem setup, we propose a trimap attention (T-Attention) block and alternatively apply it to refine them. More details are as follows.

Pseudo trimap estimator. To take into account the imperfectness of bounding box labels in semantic segmentation, we introduce the pseudo trimap representation and its estimation. The idea is to use only confident regions. While pixels outside of bounding boxes are inarguably background, pixels inside bounding boxes are a mixture of foreground and background. If a pixel within bounding box is inferred as background (false-negative), it is unsure which is the case: (A) the prediction is correct and the pixel is background contained in the bounding box, (B) the prediction is wrong and the pixel is the groundtruth foreground part.

Figure 2: Trimap example

While (B) does not require any special care, if (A) is the case, updating the model with this criterion would confuse our model. Therefore, we exclude uncertain false-negative regions for loss calculation and prototype estimation. To implement this, we use the trimap representation $T_{c,k}$ as shown in Figure 2, where the white and black regions are confident foreground and background, respectively, and the gray regions are uncertain regions according to the case (A). The gray region $\mathcal{G}_c$ for class c is determined as: $\mathcal{G}_c = \{(h,w)|M(h,w)=c, \hat{M}(h,w)\neq c\}$, where h and w indicate indices along spatial axes, and the gray region is computed by the logical subtraction of a predicted segmentation $\hat{M}$ from a box mask M of class c. This strategy helps our model learn robust to label noise.

Trimap attention module (TAM). The proposed trimap attention module, TAM, refines both prototypes and segmentation predictions of query and support samples. The module is illustrated in Figure 3, and it consists of the trimap attention (T-Attention) blocks. The same block is applied recurrently by alternating the roles of query and support. This alternation is for improving a target prediction by refining reference information, where the reference and target can be respective query and support or vice versa. Additional theoretical analysis of TAM is provided in the Supplementary.

The T-Attention block is inputted the target feature and the reference label, feature and previous segmentation prediction, and outputs the improved target segmentation prediction. As shown in Figure 3, the block consists of two steps: 1) the reference prototype refinement and 2) the target segmentation prediction by attention. Since the TAM begins with T-Attention block applied to the support as the target with the query information as the reference as depicted in Figure 3, we describe the process of the T-Attention block from this case for simplicity, because the inputs of the even steps are analogous with flipping the query and support roles as reference and target, respectively.

Given the bounding box label and initial segmentation prediction of the query, the T-Attention block first estimates the pseudo trimap $T \in \{\texttt{F},\texttt{B},\texttt{G}\}^{H\times W}$, where $\texttt{F},\texttt{B}$ and $\texttt{G}$ denote foreground, background, and gray region indication labels. The pseudo trimap label bootstraps the weak label and the previously predicted label, so that it can filter out uncertain regions. Then, we estimate the prototypes by masked average pooling [40] with the estimated pseudo trimap T^q of the query and the given query feature F_q. The masked average pooling operation $\text{MAP}(B,F) \in \mathbb{R}^d$ is defined as:

$$\text{MAP}(B,F) = \frac{\sum_{h,w} B(h,w) * F(h,w)}{\sum_{h,w} B(h,w)}, \tag{1}$$

where $B \in \{0,1\}^{H\times W}$ is a binary mask and $F \in \mathbb{R}^{H\times W\times d}$ a feature map. The expected query foreground prototype of class c is estimated by the MAP operation with respect to T^q_c, *i.e.*, $\tilde{p}^q_c = \text{MAP}(\mathbf{1}[T^q_c = \texttt{F}], F_q)$ and the background one as $\tilde{p}^q_0 = \text{MAP}(\mathbf{1}[T^q_c = \texttt{B}], F_q)$, where $\mathbf{1}[\cdot]$ denotes the indicator function. As an exceptional case, if there is no foreground intersection in the pseudo trimap T between the expected query foreground region of class c and query bounding box region of class c, we keep this class prototype as is, *i.e.*, $\tilde{p}^q_c = p_c$. The segmentation predictions of the support samples are estimated by

$$\hat{M}_{c,k}(h,w) = \arg\min_{j\in\{0,\dots,C\}} d(F_{c,k}(h,w), \tilde{p}^q_j), \tag{2}$$

where $d(\cdot,\cdot)$ is a cosine distance metric. This procedure is illustrated in Figure 3. The above procedure is applicable to the odd steps of the T-Attention block, and for the even steps, we feed the query information as the target with the support information as the reference, and the rests are analogous. We demonstrate that our proposed TAM effectively cancels out background pixels within bounding boxes in the Supplementary material. Note that our TAM does not add any additional learnable parameters. In addition, our usage of the MAP with the pseudo trimap is the extension of the standard MAP with binary segmentation [34, 40] to the weak annotation case. Our MAP improves the chance to make the feature space class-wise separable even when images from unseen domains are encountered.

Initial prototypes and segmentation. The TAM begins with an initial segmentation map w.r.t. the query. For obtaining the initial segmentation map, we first estimate the initial prototype p_c of all class $c \in \{0,\cdots,C\}$ for the supports. The initial foreground prototype of class c is computed by MAP with the support subset $\{(I_{c,k}, M_{c,k})\}$:

$$p_c = \tfrac{1}{K}\textstyle\sum_{k=1}^{K} \text{MAP}(\mathbf{1}[M_{c,k} = c], F_{c,k}), \tag{3}$$

where $c = 1,\cdots,C$. A single background prototype p_0 is computed over the entire support set $\mathcal{S}$ since every image

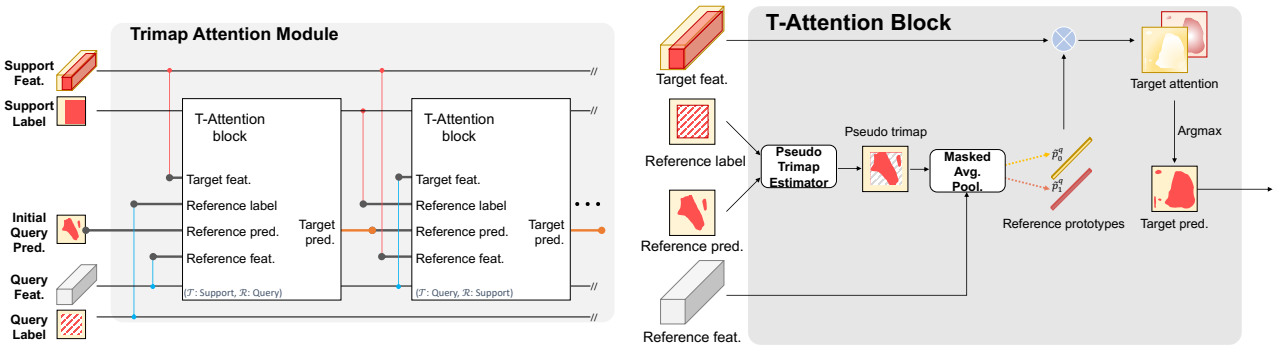

Figure 3: **[Left]** Trimap Attention Module (TAM). TAM is composed of an even number of stacked TABs. Masks are refined by alternatively updating $\mathcal{T}$ and $\mathcal{R}$ and feeding initial masks (starting from bounding boxes) to a TAB. **[Right]** Trimap-Attention Block (TAB). TAB regards the prediction and label as noisy estimators for the groundtruth mask and utilizes them to get refined foreground class information from the reference domain $\mathcal{R}$. Specifically, TAB computes pseudo-trimap representation and denoised foreground prototypes from $\mathcal{R}$ and obtains the prediction of a target domain $\mathcal{T}$.

contains background pixels.

$$p_0 = \frac{1}{CK} \sum_{c=1}^{C} \sum_{k=1}^{K} \text{MAP}(\mathbf{1}[M_{c,k} = 0], F_{c,k}). \quad (4)$$

3.2. Training and inference

Training. During training, we optimize softmax cross-entropy loss between our prediction and the bounding box on query image except for the pseudo trimap region $\mathcal{G}$.

$$\tilde{M}_{q;c}(h,w) = \frac{\exp(-\alpha d(F_q(h,w), \tilde{p}_c))}{\sum_{j \in \{0,\dots,C\}} \exp(-\alpha d(F_q(h,w), \tilde{p}_j))}. \quad (5)$$

$$\mathcal{L} = - \sum_{\substack{j \in \{0,\dots,C\}, \\ (h,w) \in I_q \setminus \mathcal{G}_j}} \mathbf{1}[M_q(h,w) = j] \log \tilde{M}_{q;j}(h,w), \quad (6)$$

which is defined for one query image. For multiple query images in an episode, we optimize the averaged softmax cross-entropy. Here, α is initialized to 20 and updated with other network parameters by the stochastic gradient descent.[3]

Inference. In test episodes, our meta-learner is evaluated by comparison of the prediction of our model $\tilde{M}_q$ and groundtruth query segmentation label M_q. The predicted segmentation mask $\hat{M}_q$ is obtained by classifying each pixel into the class of the nearest prototype.

$$\hat{M}_q(h,w) = \arg\min_j d(F_q(h,w), \tilde{p}_j) = \arg\max_j \tilde{M}_{q;j}(h,w). \quad (7)$$

Moreover, the theoretical analysis on our method is provided in the Supplementary.

4. Experiments

We conduct experiments on the Pascal-5^i and FSS-1000 datasets. Also, to evaluate the cross-dataset performance, we conduct a similar experiment in the VOC2COCO setting. Unless mentioned, we follow the standard evaluation setups used in Wang *et al*. [34], that is, we try 5 different random seeds each with 1000 episodes, and report the average of 5 runs for stabilized results. Additional experimental results based on another choice of the network structure and COCO-20^i dataset are provided in the Supplementary.

4.1. Pascal-5^i

Setup. The Pascal-5^i dataset is the parsed version of Pascal-VOC 2012 [9] with SBD [13] augmentation, firstly introduced in [29]. To facilitate the evaluation of few-shot semantic segmentation, Pascal-5^i is composed of 4 splits, in which each split has 5 class labels. Following the few-shot semantic segmentation literature, the performance is measured on a split when the other 3 splits are used for training. For example, performance on Pascal-5^0 indicates the test performance on split-0 when trained on split-1, -2, and -3.

We automatically generate bounding boxes from segmentation masks. Unlike the bounding box generation during the test time in PANet, for a more congruent setting to supervised meta-training, we assume bounding boxes of all instances are given in an image during meta-training. During testing, only the bounding box of one randomly chosen instance per support image is given as the support mask as in PANet so that we can compare with PANet directly.

Metrics. We evaluate our model based on mean-IoU and binary-IoU. Mean-IoU computes the average of Intersection-over-Union (IoU) on all foreground classes. In the binary-IoU computation, the semantic segmentation is treated as pixel-wise binary (foreground-background) classification. Regarding all foreground classes as one foreground class, the binary-IoU is computed by averaging IoU on the foreground class and the background class.

Results. Mean-IoU performances are reported in Table 1

[3] PANet fixes α as 20 and reports learning it yields little performance gain. We found that in our setting, learning α brings improved results for both baselines. We deal with it in Table 4.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
PANet (U)*	42.30	58.00	51.10	41.20	48.10	51.80	64.60	59.80	46.50	55.70
Baseline	36.74	51.89	46.63	**37.03**	43.07	45.83	57.62	56.06	41.70	50.30
Ours	**36.96**	**52.24**	**49.06**	35.23	**43.37**	**46.48**	**58.99**	**58.19**	**42.97**	**51.66**

Table 1: Mean-IoU on the 1-way 1-shot and 1-way 5-shot setting on Pascal-5^i. During the test time, segmentation masks are leveraged as the support supervision. While PANet trained with mask serve as upper bounds, our method effectively gains 0.30% of mean-IoU and 1.36% mean-IoU from lower bound in 1-shot and 5-shot settings compared to the baseline. * refers to quoted results from [34].

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
PANet (U)*	-	-	-	-	45.10	-	-	-	-	52.80
Baseline	34.25	49.69	44.14	**36.17**	41.07	40.74	54.20	50.73	39.95	46.40
Ours	**35.32**	**51.64**	**48.00**	34.77	**42.43**	**44.19**	**57.88**	**56.19**	**41.84**	**50.02**

Table 2: Mean-IoU on the 1-way 1-shot and 1-way 5-shot setting on Pascal-5^i. During the test time, bounding boxes are leveraged as the support supervision. * refers to quoted results from [34].

Method	1-shot	5-shot
PANet (U)*	66.50	70.70
Baseline	62.14	66.72
Ours	**63.02**	**68.09**

(a) Binary-IoU, Pascal-5^i
Mask label @ Test time

Method	1-shot	5-shot
PANet (U)	63.21	68.26
Baseline	59.59	62.62
Ours	**61.95**	**66.28**

(b) Binary-IoU, Pascal-5^i
Box label @ Test time

Method	1-shot	5-shot
PANet (U)	82.40	85.65
Baseline	66.70	70.67
Ours	**69.75**	**72.13**

(c) P-IoU, FSS-1000
Mask label @ Test time

Table 3: Performance comparison on the 1-way 1-shot and 1-way 5-shot settings on Pascal-5^i and FSS-1000 datasets. During the test time, segmentation mask (a,c) or box (b) labels are used as the support supervision.

Method	mean-IoU @ Pascal-5^i	Δ
Baseline w/o α-learn	50.30	-
Baseline w/o α-learn + TAM	51.43	+1.13
Baseline + TAM	**51.66**	**+1.36**

† Segmentation masks are used as support labels at the test time.

Table 4: Ablation study on the 1-way 5-shot.

and 2. Table 3a and 3b also show binary-IoU performances. Since Baseline follows the prototype learning procedure of PANet, the results from PANet can serve as our upper bound. Under the same random seed, we quote the corresponding performance of PANet if results in congruent settings are reported in [34], or reproduce PANet with fully-supervised meta-training for capturing the challenge of leveraging bounding box annotations during meta-training. Such upper bounds are denoted as PANet (U). Baseline indicates the model regarding bounding box annotations as segmentation masks, *i.e.*, without any consideration for background noises. Hence, the neural network is guided with noisy information and therefore has poor generalization ability, resulting in degraded performances. In each train episode, it extracts class prototypes from support images and their bounding box labels and trains a neural network by pixel-wise cross-entropy loss with query images and their bounding box labels. Note that the network structure of Baseline and Ours is based on PANet [34].

Specifically, in the 1-way 1-shot and 1-way 5-shot settings, mean-IoU was degraded by 5.03% and 5.40% compared to PANet (U). In this challenging setting, we find that Ours achieves 0.30% and 1.36% compared to Baseline in the 1-way 1-shot and 5-shot settings. We further evaluate our method and the baseline in the weakly-supervised test scenario. Surprisingly, Ours outperforms Baseline by 3.62% in the 1-way 5-shot setting with weak test supervision. The performance gap between Baseline and Ours is increased in harsher settings which shows the robustness of our method. Qualitative results on FSS-1000 are in the Supplementary.

Combining with classical techniques. We also report the performances of when classical techniques are combined with Baseline and Ours in Table 5. An interactive segmentation algorithm, GrabCut [28] could generate pseudo trimaps by extracting the probable foreground from bounding boxes. A post-processing segmentation algorithm, CRF [5] refines the prediction of neural networks. Hence, each method can cooperate with either Baseline or Ours to improve few-shot segmentation performances. However, it is shown that GrabCut sometimes fails to get better pseudo-trimaps, resulting in the worse performances of GrabCut+Ours than Ours.

Furthermore, our method combined with both methods outperforms the corresponding baseline with bounding boxes as test supervisions. This is in line with the result in Table 2, demonstrating the effectiveness of our method especially when no masks are provided from both base and test classes.

Method	Mask@Test		Box@Test	
	1-shot	5-shot	1-shot	5-shot
GrabCut+Baseline	**40.18**	**51.25**	38.16	48.35
GrabCut+Ours	40.08	50.48	**39.71**	**49.59**
Baseline+CRF	**45.72**	54.51	43.62	49.95
Ours+CRF	44.76	**54.77**	**44.54**	**53.54**

Table 5: Mean-IoU on the 1-way 1-shot and 1-way 5-shot settings on Pascal-5^i. Mask@Test and Box@Test denote when either segmentation masks or bounding boxes are leveraged as test supervision, respectively.

4.2. FSS-1000

Recently, FSS-1000, the first large-scale object dataset for few-shot segmentation has been suggested [19]. FSS-1000 contains 1000 classes, especially many of them have not been dealt with in other segmentation datasets. Each category contains 10 {image, segmentation mask} pairs. FSS-1000 is challenging due to the small number of samples per label and much more classes. FSS-1000 also contains synthetic images, which diversify the data distribution.

Setup. In Li *et al*. [19], a train/test set split was proposed considering the hierarchy of the dataset. Following the split configuration, 240 out of 1000 classes are used as test classes while others are used for meta-training. Bounding box annotations are generated from segmentation annotations as in Pascal-5^i. For evaluation, we randomly sample 5000 test episodes and report the average P-IoU.

Metrics. As in [19, 33], we adopt IoU of positive labels in a binary mask (P-IoU). P-IoU is in line with binary-IoU as it assumes a binary classification scenario.

Results. P-IoU performances are reported in Table 3c. Due to the wider breadth and shallower depth of the dataset, when segmentation masks are replaced by bounding boxes, the neural network loses so much information for semantic segmentation. Our method recovers more than 3% of P-IoU on unseen classes in the 1-way 1-shot case. This result implies that our method effectively enables robust meta-training even in the challenging setting and improves the meta-learning performance by a large margin. Qualitative results on FSS-1000 are provided in the Supplementary.

4.3. VOC2COCO Results

Setup. As an extended test for the cross-dataset setup, we additionally suggest the VOC2COCO setup and measure the performance on that. The Pascal-VOC 2012 dataset has 20 classes, and the MS-COCO dataset [20] has 80 classes. MS-COCO is more complicated and challenging than Pascal-VOC. For example, MS-COCO contains 3.5 categories and 7.7 instances per image, while Pascal-VOC has 1.4 categories and 2.3 instances per image on average. Also, the average size of objects of MS-COCO is smaller than that of Pascal-VOC, which makes it harder to recognize them.

Method	1-shot		5-shot	
	Mean-IoU	Binary-IoU	Mean-IoU	Binary-IoU
PANet (U)	21.85	58.66	28.11	60.24
Baseline	19.30	55.90	25.15	58.21
Ours	**20.78**	**56.42**	**25.25**	**58.47**

(a) Mean-IoU and Binary-IoU, VOC2COCO, Mask label @ Test

Method	1-shot		5-shot	
	Mean-IoU	Binary-IoU	Mean-IoU	Binary-IoU
PANet (U)	20.32	56.03	25.60	57.75
Baseline	18.22	53.71	22.99	55.87
Ours	**19.81**	**55.00**	**23.61**	**56.73**

(b) Mean-IoU and Binary-IoU, VOC2COCO, Box label @ Test

Table 6: Performance comparison on the 1-way 1-shot and 1-way 5-shot settings on VOC2COCO. During the test time, segmentation mask (a) or box (b) labels are used as the support supervision.

We utilize all 20 classes of Pascal-VOC for meta-training and test on 60 classes of the COCO 2017 dataset, where the test classes are not overlapped with those of the Pascal-VOC dataset. The meta-training setting and metrics follow Pascal-5^i, while the bounding box annotations in MS-COCO are used for the weakly-supervised test.

Results. Mean-IoU and binary-IoU performances are reported in Table 6. Our method achieves better performances than the baseline on VOC2COCO. It shows that our method improves few-shot segmentation performances on more realistic settings in which test (novel) classes have complicated samples and a larger domain shift from base classes.

5. Conclusion

This work intends to accomplish more annotation-efficient segmentation than the current few-shot segmentation. To this end, we suggest using bounding boxes instead of segmentation masks during meta-training. With the proposed pseudo trimap estimator and trimap-attention based prototype learning, our model enables weakly-supervised meta-learning for semantic segmentation robust to label noise. The favorable performance gain from various settings facilitates tackling a broader semantic class segmentation.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network, 50%; No.2022-0-00124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities, 50%). Also, this work was the result of a study on the "HPC Support" Project, supported by the "Ministry of Science and ICT" and NIPA.

References

[1] Miriam Bellver, Amaia Salvador, Jordi Torres, and Xavier Giro i Nieto. Budget-aware semi-supervised semantic and instance segmentation. In *CVPRW*, 2019.

[2] David Biertimpel, Sindi Shkodrani, Anil S. Baslamisli, and Nora Baka. Prior to segment: Foreground cues for weakly annotated classes in partially supervised instance segmentation. In *ICCV*, 2021.

[3] Lluıs Castrejón, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, 2017.

[4] Fabio Cermelli, Massimiliano Mancini, Yongqin Xian, Zeynep Akata, and Barbara Caputo. Prototype-based incremental few-shot semantic segmentation. In *BMVC*, 2021.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv*, abs/1606.00915, 2016.

[6] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, 2020.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, June 2016.

[8] Nanqing Dong and E. Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018.

[9] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–308, 2010.

[10] Qi Fan, Lei Ke, Wenjie Pei, Chi-Keung Tang, and Yu-Wing Tai. Commonality-parsing network across shape and appearance for partially supervised instance segmentation. In *ECCV*, 2020.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[12] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *arXiv*, abs/1704.06857, 2017.

[13] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[14] Seunghoon Hong, Suha Kwak, and Bohyung Han. Weakly supervised learning with deep convolutional neural networks for semantic segmentation: Understanding semantic layout of images with minimum human supervision. *IEEE Signal Processing Magazine*, 34(6):39–49, 2017.

[15] Cheng-Chun Hsu, Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised instance segmentation using the bounding box tightness prior. In *NeurIPS*, 2019.

[16] Zongliang Ji and Olga Veksler. Weakly supervised semantic segmentation: From box to tag and back. In *BMVC*, 2021.

[17] Gregory R. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICMLW*, 2015.

[18] Viveka Kulharia, Siddhartha Chandra, Amit Agrawal, Philip Torr, and Ambrish Tyagi. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In *ECCV*, 2020.

[19] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *CVPR*, 2020.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[21] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *CVPR*, June 2020.

[22] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, pages 142–158, 2020.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[24] Todd K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[25] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, volume 31. Curran Associates, Inc., 2018.

[26] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A. Efros, and Sergey Levine. Few-shot segmentation propagation with guided networks. *arXiv*, page arXiv:1806.07373, May 2018.

[27] Hasnain Raza, Mahdyar Ravanbakhsh, Tassilo Klein, and Moin Nabi. Weakly supervised one shot segmentation. In *ICCVW*, pages 1401–1406, 2019.

[28] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, page 309–314, 2004.

[29] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017.

[30] Mennatullah Siam, Naren Doraiswamy, Boris N. Oreshkin, Hengshuai Yao, and Martin Jägersand. Weakly supervised few-shot object segmentation using co-attention with visual and semantic embeddings. In *IJCAI*, pages 860–867. ijcai.org, 2020.

[31] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, volume 30. Curran Associates, Inc., 2017.

[32] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, volume 29. Curran Associates, Inc., 2016.

[33] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *ECCV*, pages 730–746, Cham, 2020.

[34] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, October 2019.

[35] Xinggang Wang, Jiapei Feng, Bin Hu, Qi Ding, Longjin Ran, Xiaoxin Chen, and Wenyu Liu. Weakly-supervised instance segmentation via class-agnostic learning with salient images. In *CVPR*, 2021.

[36] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, 2020.

[37] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 7115–7123. PMLR, 09–15 Jun 2019.

[38] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[39] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, pages 5212–5221, 2019.

[40] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S. Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *arXiv*, abs/1810.09091, 2018.