– Supplementary Material –
Learning Few-shot Segmentation from Bounding Box Annotations

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

Tae-Hyun Oh
POSTECH
Pohang, South Korea
taehyun@postech.ac.kr

In this supplementary material, we present additional details and experiments which are not included in the main text due to space constraints. All figures and references in this supplementary material are self-contained.

1. Theoretical analysis of TAM

We provide the theoretical analysis of why our TAM works. For simplicity, we consider the 1-way 1-shot case. Given a reference image with a bounding box including an object of interest, the inside of the box consists of (true) foreground (FG, $\mathcal{C}_{fg}$) and disturbing background (BG, $\mathcal{C}_{bg}$) pixels with the ratio of α:$(1-\alpha)$.

Given a feature vector $\boldsymbol{x}$ in the bounding box of the target image, a 1-step TAB classifies $\boldsymbol{x}$ into $c \in \{fg, bg\}$ by comparing distance with the estimated prototypes $\hat{\boldsymbol{p}}_c$, *i.e.*,

$$\text{if } \|\boldsymbol{x} - \hat{\boldsymbol{p}}_{bg}\| < \|\boldsymbol{x} - \hat{\boldsymbol{p}}_{fg}\| \text{: then } \boldsymbol{x} \in \mathcal{C}_{bg},$$

$$\text{if } \|\boldsymbol{x} - \hat{\boldsymbol{p}}_{bg}\| \geq \|\boldsymbol{x} - \hat{\boldsymbol{p}}_{fg}\| \text{: then } \boldsymbol{x} \in \mathcal{C}_{fg}.$$

in the Maximum Likelihood manner. Later, if $\boldsymbol{x}$ is closer to $\hat{\boldsymbol{p}}_{bg}$ and inside a bounding box, $\boldsymbol{x}$ is regarded as Gray zone and excluded from estimating prototypes and loss. We show this simple rule with estimates $\hat{\boldsymbol{p}}_c$'s is analogous to the rule with true $\boldsymbol{p}_c$'s, and its proof sketch as follows.

Before any refinement, we estimate the initial prototypes for $\{fg, bg\}$ from a reference image. When we have a sufficient number of background pixels outside of the bounding box, we can reasonably assume our estimated background prototype $\hat{\boldsymbol{p}}_{bg}$ converges to $\boldsymbol{p}_{bg}$, because we compute $\hat{\boldsymbol{p}}_{bg}$ from the outside of the bounding box. Also, given that the bounding box contains enough pixels, our estimated foreground prototype $\hat{\boldsymbol{p}}_{fg}$ converges to the linear combination of $\hat{\boldsymbol{p}}_{fg}$ and $\hat{\boldsymbol{p}}_{bg}$, *i.e.*,

$$\hat{\boldsymbol{p}}_{fg} \rightarrow \alpha \boldsymbol{p}_{fg} + (1-\alpha)\boldsymbol{p}_{bg},$$

by Law of large numbers.

Then, for $\boldsymbol{x}$ to hold $\|\boldsymbol{x} - \hat{\boldsymbol{p}}_{bg}\| < \|\boldsymbol{x} - \hat{\boldsymbol{p}}_{fg}\|$, we have:

$$\|\boldsymbol{x} - \hat{\boldsymbol{p}}_{bg}\| = \|\boldsymbol{x} - \boldsymbol{p}_{bg}\| < \|\boldsymbol{x} - \hat{\boldsymbol{p}}_{fg}\| \leq \alpha\|\boldsymbol{x} - \boldsymbol{p}_{fg}\| + (1-\alpha)\|\boldsymbol{x} - \boldsymbol{p}_{bg}\|,$$

by the triangle inequality. By reorganizing, we have $\|\boldsymbol{x} - \boldsymbol{p}_{bg}\| < \|\boldsymbol{x} - \boldsymbol{p}_{fg}\|$, which concludes the proof.

Note that this can be generalized to N-way K-shot settings. This implies, even with noisy weak labels, the gray zone and prototypes can be updated reliably with TAM.

2. Additional Results

2.1. Ablation

We have ablated the effectiveness of our method on the support and query branch respectively in Table S.1. To this end, we build the variants of our method as follows: we apply pseudo trimap estimation of a query label (PTE) and trimap-attention based prototype refinement (TAM) only one at a time during meta-training (denoted as Ours (PTE only) and Ours (TAM only), respectively) and compare with our complete method (denoted as Ours (TAM+PTE)).

We observe that the performance gain is obtained only when both branches are jointly refined by TAM and PTE. While support masks and query labels both have disturbing background pixels in them, TAM outputs refined prototypes and PTE applied to query labels corrects the loss. When only one of them is refined, the few-shot segmentation performance is marginally improved or even degraded compared to the baseline. Because the prototypes and query labels both affect the loss which updates the feature extractor, denoising undesirable artifacts from weak labels only one at a time is thus not effective due to the disturbance by the other. This demonstrates the compactness of our method, *i.e.*, being composed of only necessary parts.

2.2. Qualitative Results

Pascal-5^i. Figure S.1 shows the improved qualitative results of Ours compared to Baseline on Pascal-5^i: including

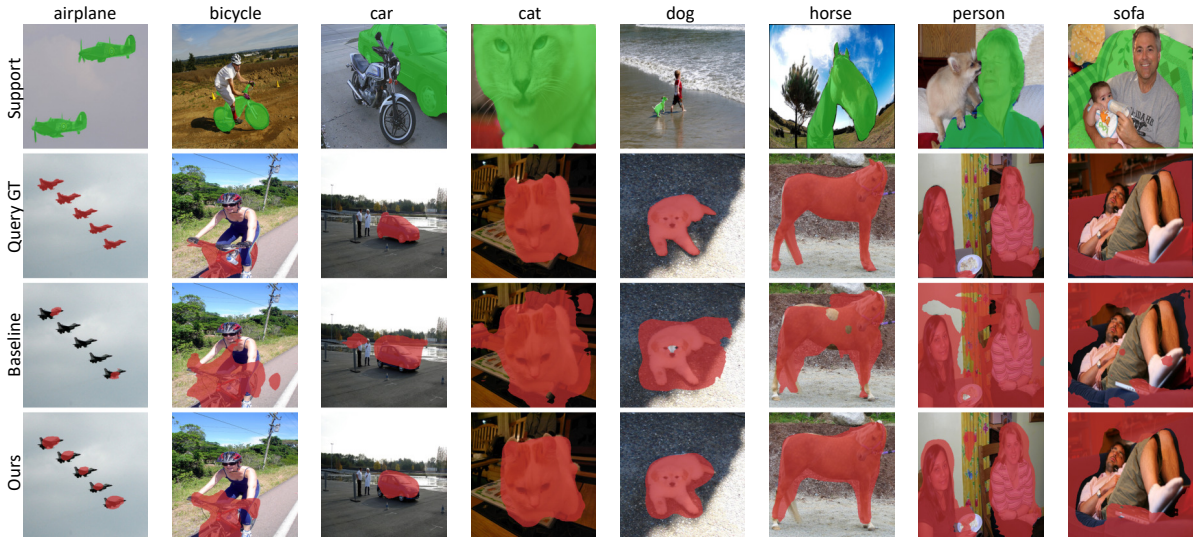

Figure S.1: Qualitative results of our model in the 1-way 1-shot segmentation on Pascal-5^i.

Method	mean-IoU	binary-IoU
PANet (U)*	55.70	70.70
Baseline	50.30	66.72
Ours (PTE only)	46.36	64.93
Ours (TAM only)	49.48	65.24
Ours (TAM+PTE)	**51.66**	**68.09**

Table S.1: Ablations of our methods. Mean-IoU and binary-IoU on the 1-way 5-shot setting on Pascal-5^i are reported. During the test time, segmentation masks are leveraged as support supervision.

thin structures (see *bicycle*), small objects (*dog*), and cluttered scenes (*bicycle* and *car*). Ours distinguishes objects-of-interests from a person around them, while Baseline tends to fail.

FSS-1000. Figure S.2 shows the qualitative result in the 1-way 1-shot setting on FSS-1000. Focusing on *taj mahal*, while the support image has a stone pillar as the background, the baseline fails to distinguish the stone pillar from the temple, which demonstrates the detailed characteristics are well captured from impure supervision with ours.

2.3. Behavior Analysis of TAM

To understand the behavior characteristics of TAM during meta-training, we qualitatively visualize our estimated trimaps during meta-training in Figure S.3, where estimated pseudo trimaps of query and support images are presented with ground-truth segments. The gray zone in the bounding box weak label implies the unconfident region other than the foreground zone. In particular, the gray zone in support pseudo trimaps is excluded for prototype learning, while

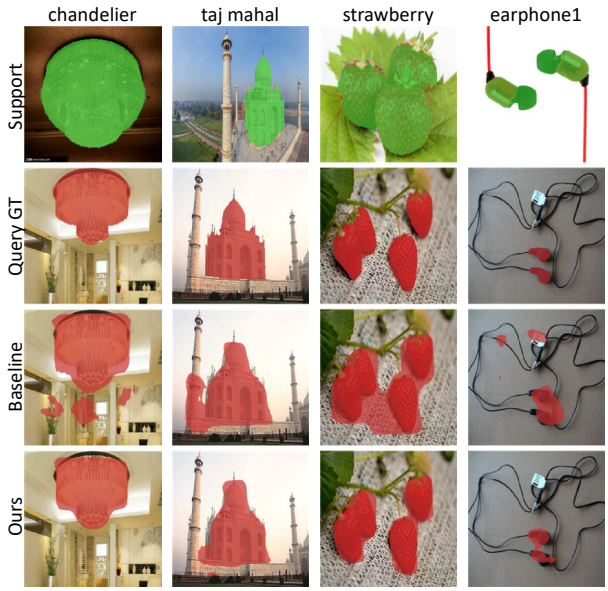

Figure S.2: Qualitative results of our model in the 1-way 1-shot segmentation on FSS-1000.

the gray zone of query pseudo trimaps is excluded for loss calculation. As shown in *sofa*, when even the ground truth labels of support and query images are inaccurate (see the middle *dog doll* region and *table* region are labeled as *sofa* in the GT label), our method effectively removes those regions in prototype learning and loss calculation.

2.4. Combining with interactive segmentation

We utilize an interactive segmentation method, GrabCut [28] to form stronger baselines. While GrabCut and other interactive segmentation methods require manual human labeling interactively, from the initial bounding box, we

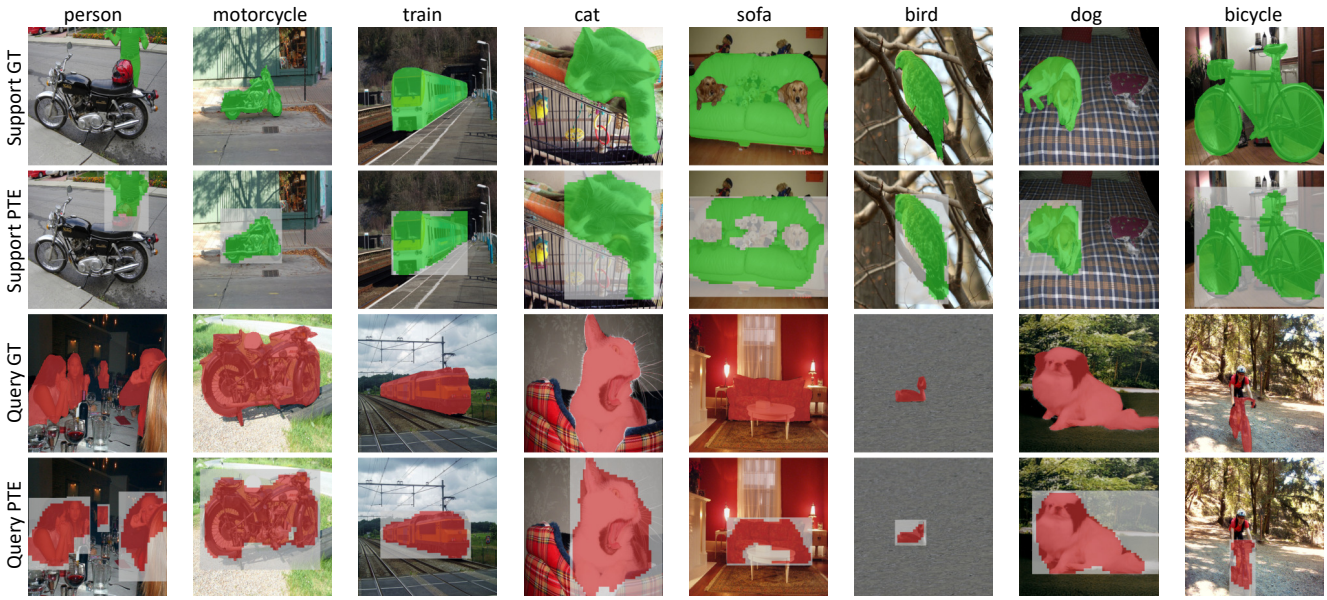

Figure S.3: Visualization of our method during meta-training in the 1-way 1-shot setting on Pascal-5^i. Given the bounding boxes of query and support images, our modules effectively estimate pseudo trimaps of both query and support images. Query ground-truth labels and support ground-truth labels are not accessible during meta-training and are only shown for comparison.

can extract rough foreground masks by those.

To make those baselines more reasonable and compare with our method more fairly, we bring our idea of trimap on those. Specifically, we perform GrabCut on support and query images with bounding box annotations and generate trimap representations during meta-training. Trimap representations are composed of the background zones as outer parts of bounding box regions, the foreground zones extracted from GrabCut, and the gray zones as the inner parts of bounding box regions which do not belong to GrabCut outputted regions.

GrabCut+Baseline and GrabCut+Ours denote that bounding boxes from support and query set are decomposed into the foreground and gray zone by GrabCut beforehand and then inputted to the baseline and ours, respectively. In GrabCut+Ours, gray zones are defined as the union of gray zones obtained from TAM and GrabCut.

We first demonstrate GrabCut can provide a strong baseline denoted as GTBbox+GrabCut in Tables S.2, S.3, S.4, and S.5. While bounding boxes for queries are not given in our setup at inference time, we use groundtruth bounding boxes followed by GrabCut at the inference stage without training for GTBbox+GrabCut. Support set is no longer used during testing since almost accurate object regions on query images are provided. Mean-IoU and binary-IoU performances of GTBbox+GrabCut are in the highest performance that implies GrabCut can be used for a stronger baseline. Furthermore, GTBbox+GrabCut can be regarded as the performance of a straightforward extension of GrabCut to segmentation with detection but with an oracle detector.

Note that the assumption of a perfect detector is very strong since the state-of-the-art fine-tuning based few-shot detector has around 40% and 50% in nAP50 (AP50 on novel classes) in the 1-shot and 5-shot settings on Pascal-VOC, respectively [S1]. This performance compared to GTBbox will lead to lower than half performance from 61.52 mean-IoU and 74.72 binary-IoU in the few-shot testing setup when integrated with a segmentor. Hence, we argue that this could function as a strong upper bound to any few-shot detector + the GrabCut segmentor. If an incremental few-shot detector is provided, our method can also be combined with that to form a realistic baseline and can be expanded to weakly-supervised incremental few-shot instance segmentation as well.

Tables S.2 and S.3 show mean-IoU and Tables S.4 and S.5 show binary-IoU on Pascal-5^i in the 1-way 1-shot and 1-way 5-shot settings, respectively. As we can see, in all the settings, our method achieves improved or comparable (mean, binary)-IoU performances than GrabCut-based methods.

In the 1-way 1-shot setting, the performances of GrabCut-based methods are worse than the methods without GrabCut. This shows how challenging our problem is. Even though GrabCut algorithm is widely used to extract a rough foreground region from a bounding box and can give a strong baseline in an ideal situation, GrabCut fails to provide more helpful supervision than bounding box labels in the 1-way 1-shot setting, and a trimap based on GrabCut yields worse performance than a bounding box.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
GTBbox+GrabCut	56.02	66.26	62.82	60.99	61.52	56.02	66.26	62.82	60.99	61.52
GrabCut+Baseline	**37.46**	50.37	47.28	25.61	40.18	**47.62**	58.91	56.33	42.13	51.25
GrabCut+Ours	33.76	48.10	44.88	33.59	40.08	45.51	58.79	56.29	41.33	50.48
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 S.2: Mean-IoU of GrabCut-based methods 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.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
GTBbox+GrabCut	56.02	66.26	62.82	60.99	61.52	56.02	66.26	62.82	60.99	61.52
GrabCut+Baseline	34.10	48.65	43.95	25.94	38.16	43.42	56.55	52.62	40.83	48.35
GrabCut+Ours	32.72	48.22	44.53	33.36	39.71	43.72	**58.28**	55.41	40.96	49.59
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 S.3: Mean-IoU of GrabCut-based methods 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.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
GTBbox+GrabCut	72.88	77.53	74.46	74.01	74.72	72.88	77.53	74.46	74.01	74.72
GrabCut+Baseline	60.77	65.62	63.05	46.87	59.08	**67.90**	72.10	69.00	60.09	67.27
GrabCut+Ours	60.76	65.60	63.19	54.63	61.04	67.57	**72.97**	70.44	60.46	67.86
Baseline	62.03	67.22	63.03	**56.27**	62.14	66.82	70.79	69.10	60.17	66.72
Ours	**62.83**	**68.76**	**65.25**	55.22	**63.02**	67.68	72.53	**71.04**	**61.10**	**68.09**

Table S.4: Binary-IoU of GrabCut-based methods 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.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
GTBbox+GrabCut	72.88	77.53	74.46	74.01	74.72	72.88	77.53	74.46	74.01	74.72
GrabCut+Baseline	57.89	63.38	59.46	46.40	56.78	64.24	69.16	64.98	58.30	64.17
GrabCut+Ours	59.43	65.35	62.11	54.83	60.43	**65.90**	**72.08**	**69.00**	**59.79**	**66.69**
Baseline	59.00	64.28	59.73	**55.37**	59.59	62.15	66.81	63.62	57.88	62.62
Ours	**61.10**	**67.78**	**63.71**	55.21	**61.95**	65.64	71.02	68.74	59.71	66.28

Table S.5: Binary-IoU of GrabCut-based methods 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.

Another interesting observation is that GrabCut+Ours outperforms GrabCut+Baseline when bounding boxes are provided as support supervision during testing. This implies our method could remove background noise well and improve the few-shot segmentation performance from more degraded supervision.

2.5. Classwise mean-IoU on Pascal-5^i

Tables S.6 and S.7 show classwise mean-IoU performances on Pascal-5^i when masks are provided as support supervision during testing. In the 1-way 1-shot and 1-way 5-shot settings, our method achieves better performances on 11 and 17 classes out of 20 classes, respectively. Specifically, among 9 classes in which our method has worse performance than the baseline in the 1-way 1-shot setting, ours performs

Method	split-0					split-1					split-2					split-3					mean
	aeroplane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	
Baseline	**51.02**	46.26	39.73	**33.16**	13.55	68.61	**43.24**	65.39	**18.93**	63.30	20.90	62.16	58.43	**59.03**	32.62	**14.48**	**60.03**	**32.59**	**54.26**	23.76	43.07
Ours	48.52	**48.37**	**45.19**	26.84	**15.89**	**69.13**	42.27	**68.53**	16.08	**65.19**	**24.63**	**65.36**	**61.20**	54.94	**39.15**	14.37	59.61	27.42	50.67	**24.09**	**43.37**
Δ	-2.50	**+2.10**	**+5.46**	-6.32	**+2.34**	**+0.52**	-0.97	**+3.14**	-2.85	**+1.89**	**+3.73**	**+3.20**	**+2.77**	-4.09	**+6.53**	-0.12	-0.42	-5.17	-3.60	**+0.33**	**+0.30**

Table S.6: Classwise mean-IoU of our method and baseline on the 1-way 1-shot setting on Pascal-5^i. For each test class, classes from other splits are used during meta-training. During the test time, segmentation masks are leveraged as the support supervision.

Method	split-0					split-1					split-2					split-3					mean
	aeroplane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	
Baseline	59.41	55.49	48.68	**41.11**	**24.45**	74.08	50.29	71.54	**24.02**	68.14	34.21	67.24	63.30	68.20	47.34	17.64	62.48	38.23	60.93	29.22	50.30
Ours	**60.11**	**57.89**	**51.96**	38.84	23.59	**76.40**	**53.11**	**72.54**	22.88	**70.02**	**36.40**	**69.84**	**64.60**	**68.47**	**51.66**	**20.41**	**64.43**	**38.84**	**61.40**	**29.76**	**51.66**
Δ	**+0.69**	**+2.40**	**+3.28**	-2.27	-0.86	**+2.32**	**+2.82**	**+1.00**	-1.14	**+1.88**	**+2.19**	**+2.60**	**+1.30**	**+0.27**	**+4.32**	**+2.76**	**+1.95**	**+0.61**	**+0.46**	**+0.54**	**+1.36**

Table S.7: Classwise mean-IoU of our method and baseline on the 1-way 5-shot setting on Pascal-5^i. For each test class, classes from other splits are used during meta-training. During the test time, segmentation masks are leveraged as the support supervision.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
PMMs (U)*	51.98	67.54	51.54	49.81	55.22	55.03	68.22	52.89	51.11	56.81
PMMs-Baseline	36.23	48.96	34.99	34.94	38.78	36.81	49.68	35.88	35.91	39.57
PMMs-Ours	**41.20**	**52.32**	**40.50**	**37.36**	**42.84**	**40.77**	**54.80**	**41.90**	**40.12**	**44.40**

Table S.8: Mean-IoU on the 1-way K-shot setting on Pascal-5^i ($K = 1, 5$). During the test time, segmentation masks are leveraged as the support supervision. * refers to quoted results from [36].

better than the baseline on 7 classes in the 1-way 5-shot setting. Also, mean-IoU differences from ours and the baseline (denoted as Δ in Tables S.6 and S.7) are increased on 12 classes from the 1-way 1-shot setting to the 1-way 5-shot setting, which means that on 12 classes, our method improves mean-IoU performances from the 1-way 1-shot setting to the 1-way 5-shot setting more than the baseline. This implies that our method can integrate information well from many support samples and infer more accurately than the baseline.

2.6. PMM-based models

TAM can cooperate with more complicated prototype-based models. To this end, we adapt PMMs [36] which extract multiple prototypes per each class and perform clustering on pixel-wise features to classify each pixel. PMMs are powerful since they are an advanced model that contains both prototype-based branch (P-Conv) and matching-based branch (P-Match). When estimating prototypes, PMMs find N prototypes from every semantic class by conducting the iterative EM clustering algorithm on pixel-wise features and finding the mean of each cluster. In PMMs, it considers 1-way setting and denotes the foreground prototypes and background prototypes as $\{\mu^+\}_{n=1}^N$ and $\{\mu^-\}_{n=1}^N$.

PMMs-Baseline is constructed by replacing full supervisions (segmentation masks) with weak supervisions (bounding boxes) during meta-training of PMMs. We observe the performance degradation compared to the fully-supervised model, PMMs (U). That says, even with more complicated structures, utilizing only bounding boxes during meta-training brings a challenge for models to learn class-separable feature spaces.

To resolve this issue, TAM cooperates with PMMs so that based on refined foreground prototypes, P-Conv branch can provide denoised probability maps and P-Match branch can activate query features with more accurate class-wise information. To construct PMMs-Ours, we apply TAM to the prototype estimation part. In PMMs-Ours, the masked average pooling operation in the TAB is replaced by prototype-mixture models (PMMs) to output multiple prototypes by EM clustering for both foreground and background classes. Also, attention is measured over all foreground and background prototypes. The target prediction is pixel-wise classification.

We report the few-shot segmentation performances of PMMs-Ours and PMMs-Baseline on Pascal-5^i in S.8and the additional challenging few-shot benchmark, COCO-20^i in S.9, following the experimental setting from [36]. To follow the experiment setup of [36] to set the result from [36] as the direct upper bound to our model, we have not used the instance-wise information during meta-training, *i.e.*, bounding boxes are generated regarding the semantic mask belongs to one instance, consequently inducing more background noises. This is unlike the experiment on Pascal-5^i from the main paper, which generates bounding boxes from each instance respectively and concatenate all bounding boxes. This is due to the uncoordinated few-shot experiment set-

Method	1-shot					5-shot					10-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
PMMs (U)*	29.28	34.81	27.08	27.27	29.61	33.00	40.55	30.29	33.27	34.28	-	-	-	-	-
PMMs-Baseline	**27.28**	**29.80**	**26.36**	24.74	**27.04**	30.97	33.52	29.55	29.63	30.91	31.10	**36.44**	27.84	29.34	31.18
PMMs-Ours	27.25	28.84	25.49	**24.95**	26.63	**34.94**	**34.94**	**32.05**	**31.87**	**33.45**	**36.11**	36.22	**30.51**	**33.25**	**34.02**

Table S.9: Mean-IoU on the 1-way K-shot setting on COCO-20^i ($K = 1, 5, 10$). During the test time, segmentation masks are leveraged as the support supervision. * refers to quoted results from [36].

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
Baseline+CRF	**38.90**	**55.09**	50.04	**38.87**	**45.72**	**50.75**	61.75	60.82	44.72	54.51
Ours+CRF	37.65	54.05	**51.29**	36.06	44.76	49.67	**61.82**	**62.17**	**45.43**	**54.77**

Table S.10: Mean-IoU on the 1-way K-shot setting on Pascal-5^i ($K = 1, 5$). During the test time, segmentation masks are leveraged as the support supervision.

Method	1-shot					5-shot				
	split-0	split-1	split-2	split-3	mean	split-0	split-1	split-2	split-3	mean
Baseline+CRF	36.63	52.38	47.21	**38.26**	43.62	44.82	57.65	54.52	42.80	49.95
Ours+CRF	**37.35**	**53.76**	**51.02**	36.03	**44.54**	**48.61**	**60.79**	**60.34**	**44.43**	**53.54**

Table S.11: Mean-IoU on the 1-way K-shot setting on Pascal-5^i ($K = 1, 5$). During the test time, bounding boxes are leveraged as the support supervision.

ting. In fact, the splits used for Pascal-5^i and COCO-20^i differ from [36] and [34] although they use the same name. In our work, since we borrow the official code and set the performances from the original papers as the upper bound of our suggested models, it is able to capture how utilizing only weak annotations during meta-training degrades the conventional few-shot segmentation models and how our method retrieves the degraded performances across diverging experiment settings.

In COCO-20^i, we found that in the 1-shot setting, our method obtains comparable results while in 5-shot and 10-shot settings, our method outperforms the baseline by a large margin, 2.54% and 2.84%, respectively.

2.7. CRF-based Post-processing

During the inference time, we have utilized CRF [5] to refine the prediction of neural networks on Pascal-5^i. When we use full masks as test supervisions, in 1-shot setting, the final performance of Ours+CRF is lower than Baseline+CRF. In 5-shot setting, the final performance of Ours+CRF is slightly better than Baseline+CRF. The interesting point is when we utilize bounding boxes as test supervisions. In more challenging and practical setting (when no segmentation masks are available in both meta-training and inference), Ours+CRF tends to outperform Baseline+CRF in both 1-shot and 5-shot settings, which aligns with the result without post-processing. This shows the robustness of our method when combined with post-processing vision algorithms.

3. Implementation details

We empirically observe that the T-Attention blocks alternation in the trimap attention module is quickly saturated; thus, we fix the T-Attention blocks alternation steps as two in our implementation.

In Baseline and Ours, we up-sample the down-sampled activation obtained from the backbone feature extractor, VGG-16, to be the input size by the bilinear interpolation, so that a feature map with the same size of an input image is obtained. Additionally, we utilize GrabCut function from OpenCV library to implement GrabCut-based methods. All codes are implemented based on the official code repository of PANet [34].

For PMMs-Ours and PMMs-Baseline, we used ResNet-50 as backbone CNN for feature extraction. Also, we followed the original setting from PMMs [36] such as evaluation protocol, the number of prototypes per class ($N = 3$), K-shot learning (during meta-training, the models are trained by 1-shot mode; during testing, K support images are sent to the PMMs together to estimate prototypes). For reliable results, we evaluate performances over 5000 randomly-sampled test episodes. All codes are implemented based on the official code repository of PMMs [36].

References

[S1] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In CVPR, 2021.