Adapt Before Comparison: A New Perspective on Cross-Domain Few-Shot Segmentation

Supplementary Material

1. Metrics

We claimed reporting mIoU only does not reflect performance appropriately. The relationship between mIoU, FB-IoU and foreground ratio is derived in the following.

mIoU Mean intersection over union (mIoU) is calculated by

1. Accumulation of intersection areas, union areas over query predictions q Eq. 1
2. Calculating IoU for each semantic class,Eq. 2
3. Averaging the class-wise IoUs, Eq. 3

$$\begin{aligned} I_c &= \sum_q TP_{q,c} \\ U_c &= \sum_q TP_{q,c} + FP_{q,c} + FN_{q,c}, \end{aligned} \tag{1}$$

$$IoU_c = \frac{I_c}{U_c} \tag{2}$$

$$mIoU = \frac{1}{C} \sum_{c=1}^{C} IoU_c \tag{3}$$

with true positives TP counting pixels where both prediction and ground truth label equal c, FP being the number of pixels where c was falsely predicted and FN the amount of ground truth c-labels which were predicted as another class.

Note that the number C of categories in the dataset does not include the background class.

FB-IoU In 1-way segmentation, which we and all previous CD-FSS focus on, the task is binary segmentation: For each episode, a class c is selected, query and support containing c are sampled, c is treated as *f*oreground and everything $\neq c$ is treated as complementary *b*ackground. Foreground background intersection over union is calculated through

1. Accumulating the areas of intersection and union with respect to both c and $\neq c$ - Eqs. (1) and (4)
2. Treating all classes equal by aggregating their metrics to fore-and background - Eq. 5
3. Averaging IoU for foreground and background - Eqs. (6) and (7)

We can obtain the background class metrics in the style of Eq. 1 through

$$\begin{aligned} I_{\neq c} &= \sum_q TN_{q,c} \\ U_{\neq c} &= \sum_q TN_{q,c} + FN_{q,c} + FP_{q,c}, \end{aligned} \tag{4}$$

where $TN_{q,c}$ indicates that both prediction and ground truth did not predict c. The foreground and background intersections and unions are then obtained through

$$\begin{aligned} I_f &= \sum_c I_c, \quad & I_b &= \sum_c I_{\neq c} \\ U_f &= \sum_c U_c, \quad & U_b &= \sum_c U_{\neq c} \end{aligned} \tag{5}$$

$$IoU_f = \frac{I_f}{U_f}, \quad IoU_b = \frac{I_b}{U_b} \tag{6}$$

$$FB\text{-}IoU = \frac{1}{2}(IoU_f + IoU_b) \tag{7}$$

To efficiently handle mIoU and FB-IoU in implementation, we and previous work represent I and U in a $2 \times C$-matrix each, in which the first row stores the $I_{\neq c}$ vector and the second row the I_c vector for the intersection matrix, likewise with U for the union matrix.

Problem of mIoU In the main paper, we showed an example where a naive predictor can outperform previous work by simply predicting always foreground. We inspect the expected performances of random prediction behaviour. Its chance to predict a true positive in Eq. 1 is $r_{\hat{y}} \cdot r_y$, where the both terms denote the foreground ratio of the prediction and ground-truth, respectively. The probabilities for false positives and negatives are $p(FP) = r_{\hat{y}} \cdot (1 - r_y)$ and $p(FN) = (1 - r_{\hat{y}}) \cdot r_y$, such that equation 2 will evaluate to:

$$IoU_c = \frac{r_{\hat{y}} r_y}{r_{\hat{y}} r_y + r_y(1 - r_{\hat{y}}) + (1 - r_y) r_{\hat{y}}} \tag{8}$$

for all c, letting us obtain also IoU_f in Eq. 6. The background IoU_b can be equally obtained by substituting all r with $1 - r$ in Eq. 8. From these, we can obtain both $mIoU$ and $FB\text{-}IoU$ through equations Eq. 3 and Eq. 7 respectively.

Fig. 1 visualizes the expected values for both metrics as a function of the foreground ratios. The fact that a higher

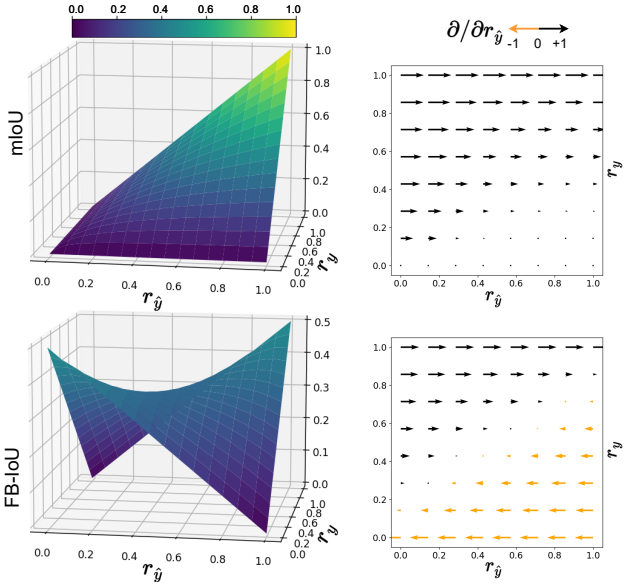

Figure 1. Results of a random mask predictor in 1-way FSS as a function of *Bernoulli-sampled predicted foreground probability* $r_{\hat{y}}$ and *dataset ground truth foreground ratio* r_y (left) and its gradients with respect to the chosen predicted foreground ratio (right). For mIoU, the gradient is always positive, meaning one can get an increase in mIoU by increasing foreground prediction ratio, while for FB-IoU such overprediction is punished.

predicted foreground ratio leads to higher mIoU is reflected by its non-negative derivative $w.r.t.\ r_{\hat{y}}$:

$$\frac{\partial(mIoU)}{\partial r_{\hat{y}}} = \frac{r_y^2}{(r_y r_{\hat{y}} - r_y - r_{\hat{y}})^2} \tag{9}$$

In contrast, the derivative of the FB-IoU

$$\frac{\partial(FB\text{-}IoU)}{\partial r_{\hat{y}}} = \frac{r_y^2}{(r_{\hat{y}} + r_y - r_{\hat{y}} r_y)^2} - \frac{(r_y - 1)^2}{(1 - r_{\hat{y}} r_y)^2} \tag{10}$$

can be negative and is zero at $r_y = r_{\hat{y}} = \frac{1}{2}$. Compare Fig. 1.

Discussion We showed that mIoU performance can be boosted by increasing the foreground prediction ratio in 1-way FSS by the example of a random predictor. In reality, the prediction has some confidence and suppressing almost-sure background naturally decreases the union area in the denominator and hence increases mIoU. Exploiting the remaining uncertainty in a foreground-biased manner still boosts mIoU, which contrasts the intuition that the maximum performance should be reached when predicted and ground truth foreground areas match. In standard semantic segmentation, this is less an issue, since the categories in the dataset C typically equals the number of possible labels to be assigned. However, in 1-way FSS, and in particular CD-FSS, where the uncertainty is still high, the phenomena we highlighted warrants careful consideration. Note that the problem cannot be fixed by including the background as a semantic class for mIoU calculation, since it will still have minor contribution for large C. Moreover, simply adding the background class is not semantically meaningful because the background is not a consistent class across episodes. In 1-way episodes, there is one class selected as the foreground class, and others are treated as background. As a consequence, background objects in one episode can be foreground objects in another. As an alternative, we showed FB-IoU is a metric to reveal overprediction behaviour.

mIoU has been preferred over FB-IoU in previous work because it is considered to give better judgment about the generalizability of the model [8]. This can be understood in the sense that mIoU punishes bad predictions on single classes and underrepresented classes in comparison with FB-IoU. We agree, hence the mIoU measure should not be replaced, but complemented with the foreground ratio sensitive FB-IoU.

2. Deepglobe Issue

In the paper we argued the benchmark's[7] Deepglobe [3] dataset is not appropriate due to annotation issues. Deepglobe is an established and widely used dataset - the problem

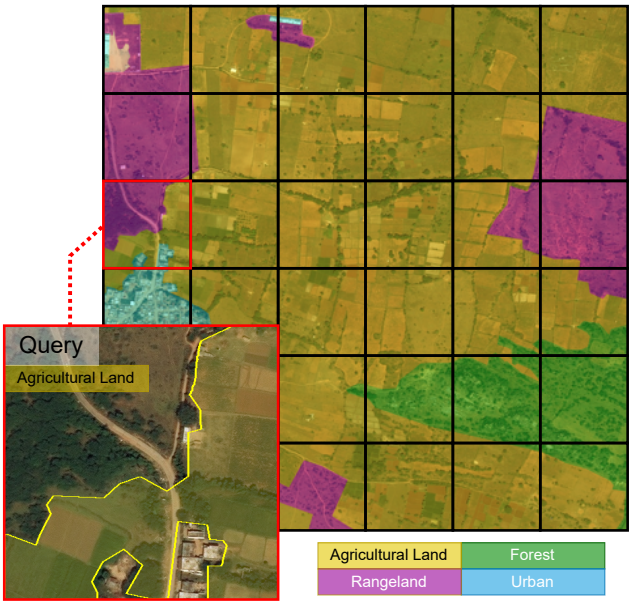

Figure 2. Cause of the Deepglobe Issue. The image from the *Agricultural Land* episode we inspected in the main paper is a crop (red cell) from the here shown larger original[3] image (2448×2448). Cropping is done following the CD-FSS benchmark[7]. While in the scale of the original image the inaccuracies are minor, at the zoom level of the cells it becomes intolerable. We suggest the benchmark should be adjusted accordingly. Note that also the upper left region in the query is actually *Forest*, not *Rangeland*.

only emerges because of heavy cropping applied in the pre-processing for the benchmark. Its creators claim that cropping has little effect because objects in sattelite images have no regular shape, but from Fig. 2 it becomes evident that the actual problem is that, at a higher zoom level, small spatial inaccuracies have large impact, such that almost half of the shown image is annotated wrongly. Another example with with the same issue can be viewed in the first row of Fig. 6.

Table 1. Table from main paper in full. Intra- (FG↔FG) and inter- (FG↔BG) class similarities in the embedding space of (L)ow, (M)iddle and (H)igh-level feature maps. Measure represents averaged cosine similarities of pixel pairs from same and opposite classes, respectively. A higher delta represents higher discriminability. The intra-image statistic measures similarity within the support, across its pixel pairs which match the (FG↔FG)/(FG↔BG) criterion. The inter-image statistic measures similarity between query and support, across query-support pixel pairs. In case of overfitting, the intra-support discriminability would rise without bringing improvement for the inter query-support measure. The latter we argue has direct positive impact on our query-support cross-attention module, as well as the hypercorrelations in [7, 8] and dense affinity matrices in [2].

		Deepglobe			ISIC			Chest			FSS			SUIM		
	Metric	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
Before Task-Adaption	**Intra-Support**															
	FG↔FG ✓	0.65	0.52	0.64	0.73	0.62	0.73	0.68	0.57	0.68	0.56	0.46	0.64	0.60	0.54	0.69
	FG↔BG ×	0.63	0.47	0.58	0.70	0.57	0.63	0.63	0.48	0.59	0.51	0.39	0.56	0.56	0.45	0.60
	delta Δ	0.02	0.05	0.06	0.03	0.05	0.10	0.04	0.09	0.10	0.05	0.07	0.07	0.05	0.09	0.09
	Inter-Query-Support															
	FG↔FG ✓	0.63	0.49	0.59	0.69	0.56	0.63	0.67	0.55	0.67	0.53	0.41	0.60	0.51	0.42	0.58
	FG↔BG ×	0.62	0.46	0.57	0.68	0.55	0.63	0.63	0.48	0.59	0.50	0.39	0.56	0.50	0.41	0.57
	delta Δ	0.01	0.03	0.02	0.01	0.01	0.01	0.04	0.07	0.08	0.03	0.03	0.04	0.01	0.02	0.01
After Task-Adaption	**Intra-Support**															
	FG↔FG ✓	0.12	0.26	0.34	0.19	0.40	0.53	0.20	0.36	0.39	0.29	0.42	0.47	0.32	0.45	0.50
	FG↔BG ×	-0.04	-0.11	-0.14	-0.06	-0.13	-0.16	-0.08	-0.14	-0.15	-0.11	-0.16	-0.16	-0.10	-0.14	-0.14
	delta Δ	0.17	0.37	0.48	0.25	0.53	0.69	0.28	0.50	0.54	0.41	0.58	0.63	0.42	0.59	0.65
	Inter-Query-Support															
	FG↔FG ✓	0.03	0.05	0.05	0.06	0.13	0.19	0.17	0.31	0.33	0.18	0.28	0.33	0.14	0.20	0.16
	FG↔BG ×	-0.01	-0.02	-0.01	-0.02	-0.04	-0.06	-0.06	-0.13	-0.12	-0.06	-0.10	-0.11	-0.04	-0.04	-0.03
	delta Δ	0.05	0.07	0.06	0.07	0.18	0.25	0.23	0.44	0.45	0.25	0.38	0.44	0.18	0.24	0.18

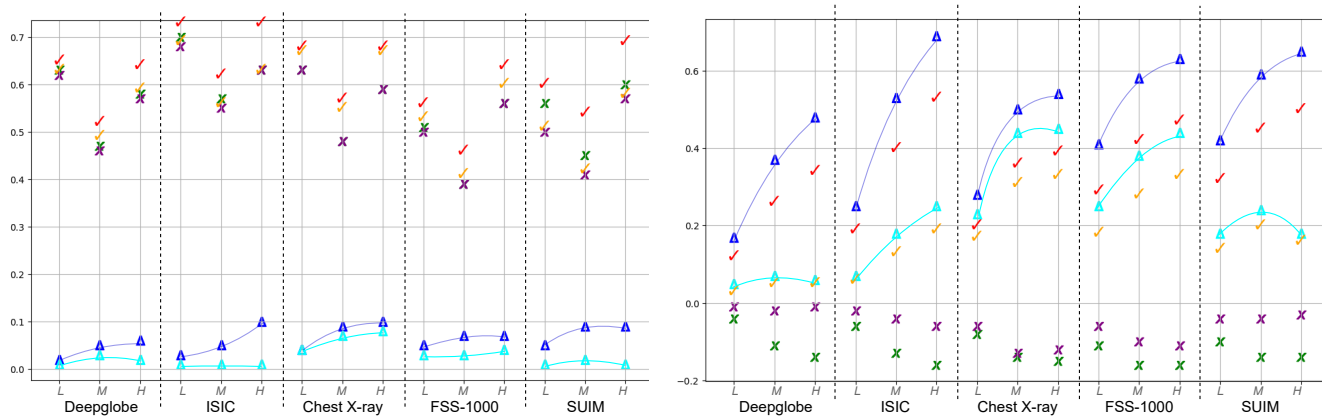

Figure 3. Visualization of Tab. 1. Left: Before *T*ask *A*daption, right: After TA. Checkmarks represent average same-class similarity, crosses average opposite-class similarity. The most important measure for the success of the query segmentation is our discriminability measure *delta* in cyan, representing the distance between check- and crossmarks. An overfitting to the support set could be interpreted as the vertical distance between blue and cyan in the right diagram. *H*igh level features tend to be more susceptible to this (see cyan drop on Deepglobe and SUIM), but still provide important semantic information (highest on ISIC and FSS). In the main paper we noted good performance on ChestXray without TA, which is supported by seeing it to have the highest inter-query-support *delta* in the left diagram. Note also the position of 0 on the y-axis in both charts, indicating on the left the cyan *delta* is almost zero for Deepglobe, ISIC and SUIM, whereas on the right TA could pushed opposite-class similarity below zero.

3. Task Adaption and Embedding Space

Tab. 1 reports our measures in the feature spaces after backbone and attached network respectively. We consider this to be useful for researchers to understand the challenges in CD-FSS and our contribution to solve them.

Pixel-to-pixel similarities are measured because they are the basis for dense comparison. We use ResNet-50 and extract the 13-layer feature pyramid following [8, 10]. Measurement is performed independently for each layer, their index l is dropped. Masks are first downsized by bilinear interpolation to match the feature volume size. Intra-support similarities are obtained with the masked feature volumes

$$\begin{aligned} F_f^s &= \{F^s | M^s > 0.5\} \\ F_b^s &= F^s \setminus F_f^s. \end{aligned} \tag{11}$$

Then,

$$sim_{F\leftrightarrow F}^{s\leftrightarrow s} = \frac{1}{|F_f^s|}^2 \sum_{f_i \in F_f^s} \sum_{f_j \in F_f^s} c(f_i, f_j) \tag{12}$$

$$sim_{F\leftrightarrow B}^{s\leftrightarrow s} = \frac{1}{|F_f^s||F_b^s|} \sum_{f_i \in F_f^s} \sum_{f_j \in F_b^s} c(f_i, f_j), \tag{13}$$

with cosine similarity $c(\cdot)$. Equally for inter-query-support similarities, we mask the query features

$$F_f^q = \{F^q | M^q > 0.5\}, \tag{14}$$

$$F_b^q = F^q \setminus F_f^q. \tag{15}$$

Then,

$$sim_{F\leftrightarrow F}^{q\leftrightarrow s} = \frac{1}{|F_f^q||F_f^s|} \sum_{f_i \in F_f^q} \sum_{f_j \in F_f^s} c(f_i, f_j) \tag{16}$$

$$sim_{F\leftrightarrow B}^{q\leftrightarrow s} = \frac{1}{|F_f^q||F_b^s|} \sum_{f_i \in F_f^q} \sum_{f_j \in F_b^s} c(f_i, f_j). \tag{17}$$

Finally, the delta between the intra- and inter-class distances can be interpreted as the discriminability within support

$$delta^{s\leftrightarrow s} = sim_{F\leftrightarrow F}^{s\leftrightarrow s} - sim_{F\leftrightarrow B}^{s\leftrightarrow s} \tag{18}$$

and across (inter) query and support:

$$delta^{q\leftrightarrow s} = sim_{F\leftrightarrow F}^{q\leftrightarrow s} - sim_{F\leftrightarrow B}^{q\leftrightarrow s} \tag{19}$$

The block-wise $L/M/H$ measure is obtained by averaging the measure of layers belonging to a block, as in [8, 10] the $L/M/H$ split for our 13 layers is $(4/6/3)$.

From Tab. 1 dataset-specific characteristics become apparent. Fig. 3 provides an intuitive understanding of the relationship between the measures.

4. On Affinity and Correlation Maps

Fig. 5 visualizes the correlation maps that are the result of the dense comparison from Sec. 3.3 of the main paper. Here we attempt to provide more intuition on their *construction*, subsequent *thresholding* and *refinement*.

Construction of $\hat{q}_{pred_l}$ is similar to [10], but since it is the core comparison mechanism of our approach, we attempt to break it down to make it more understandable why it works. A correlation map is calculated from query features, support features and support mask. The steps are 1) query-support pixel-to-pixel dot product, 2) softmax over the support dimension 3) filtering support foreground class.

$$\hat{q}_{pred_l} = \underbrace{softmax(\overbrace{fl(\hat{F}_l^q) fl(\hat{F}_l^s)^T / \sqrt{d}}^{1)})}_{2)} \underbrace{fl(M_l^s)}_{3)}. \tag{20}$$

1) Query features $\hat{F}_l^q$ and support features $\hat{F}_l^s$ are multiplied. By flattening fl, feature volumes are converted into matrices with spatial dimensions represented in the first axis and channel dimensions in the second. This results in a matrix multiplication between $HW \times C$ and $C \times HW$, yielding a dense pixel-to-pixel affinity map of shape $HW \times HW$. Each element of this map is a dot product of two C-dimensional feature vectors, indicating the similarity between individual query and support pixels. Division by square root of channel dimension d is only scaling.

2) For any given query pixel (specific row), taking the softmax over its similarities to all support pixels (columns) accentuates support pixels with high similarity, pushing their values towards 1.

3) Multiplying a HW-shaped row of the affinity map with the HW-shaped support mask vector filters out support background regions and aggregates the remaining foreground similarities. As a result, $\hat{q}_{pred}$ will highlight query pixels with large similarity to the support foreground.

Thresholding. Estimating the correct foreground ratio has been shown [1] to be a primary driver for performance in FSS. We use function $thresh$ to obtain binary $\hat{M}^q$ from $\hat{q}_{fused}$. A simple idea would be to classify every pixel with a score larger than its expected value as foreground. For random features, the expected value of $\hat{q}_{pred}$ and thus also $\hat{q}_{fused}$ equals $mean(M^s)$, i.e. the foreground ratio in the support set, because we obtained $\hat{q}_{pred}$ by $softmax(\cdots)M^s$ in Eq. 20. Fig. 4 shows that the correlation scores (x-axis) are distributed around this $mean(M^s)$, but we can also observe that choosing it as a threshold would lead to overprediction. From the shown samples it becomes apparent why a) separating the foreground cluster through k-means/Otsu's[9] is an efficient strategy, b) we choose $thresh(\hat{m}) = max(mean(\hat{m}), otsus(\hat{m}))$ as the threshold.

We believe the understanding of the distributions is relevant for the future development of models that want to further process correlation maps.

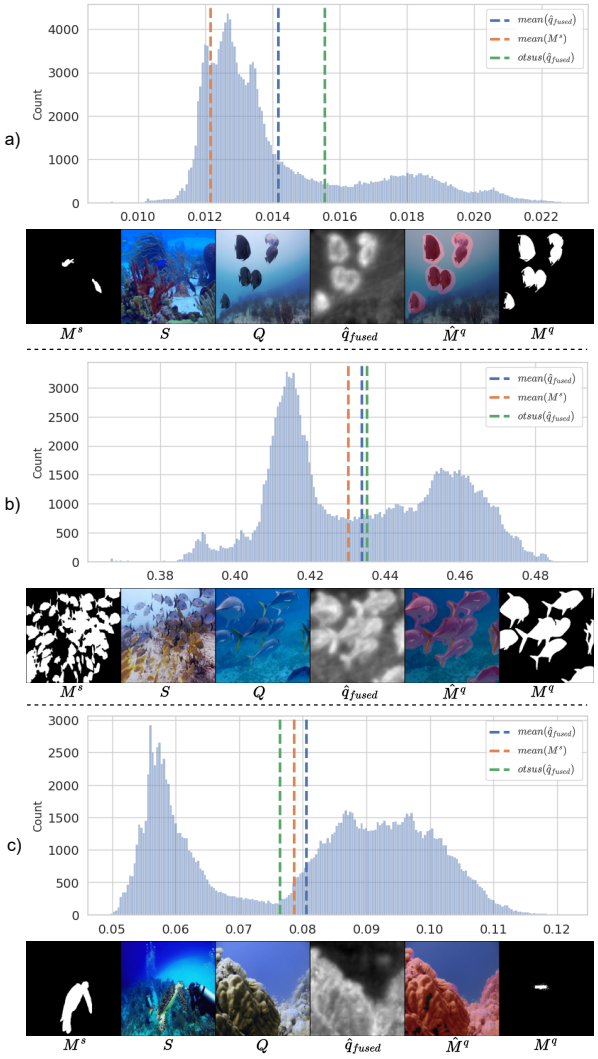

Figure 4. Histograms of correlation/prediction maps $\hat{q}_{fused}$. Cases a) and b) represent success cases where the foreground objects (right cluster in the histogram) are easily segmentable by $otsus$ (green vertical). Case c) also seems to feature two clearly distinct Gaussians, but the threshold would fall below the average prediction score across pixels (blue vertical). The right cluster is too similar to the average score, which indicates the cluster rather represents an "unknown" class which can be distinguished from the support background cluster (left) but is not very similar to the support foreground object. Indeed, we can see that 1) the backgrounds in Q and S are similar (sea), 2) the object highlighted in our $\hat{q}_{fused}$ is not similar to either support foreground (turtle) or background, 3) the actual query ground truth object (tiny hidden fish in Q, M^q) is visually disparate from the support turtle and hidden in unknown background, making it too difficult to segment. In this case, the average $\hat{q}_{fused}$ (blue) serves as the threshold.

Refinement. As a postprocessing step, the prediction mask $\hat{M}^q$ is refined through applying [4, 6]. Not for all domains this is beneficial, and in the main paper we mentioned it can be verified by forwarding a pseudoepisode constructed from the support set. We provide Algorithm 1 for a detailed description of the process. For the Chest X-ray dataset for example, it is mostly not beneficial, such that the refinement is mostly not applied. This also reflects in Chest X-ray's slightly inverse relationship between performances *Ours(no-pp)* and *Ours* in Tab. 4 of the main paper.

Algorithm 1 Dynamic Refinement Decision.

Require: Query image I^q, Support set I^s, M^s ▷ Test Task
Require: Orig. Support Features $\hat{F}^s$
Require: Augm. Support Features $\hat{F}^{\tilde{s}_1}$ ▷ backprojected
Require: Prediction $\hat{q}_{fused}$ ▷ Result of main paper Eq. 7
  $Q \leftarrow \hat{F}^s$ ▷ pseudoquery
  $K \leftarrow \hat{F}^{\tilde{s}_1}$ ▷ pseudosupport
  $V \leftarrow M^s$
  $\hat{s}_{fused} \leftarrow forward(Q, K, V)$ ▷ main paper Eq. 6-7
  $\tau \leftarrow thresh(\hat{s}_{fused})$
  $\hat{M}^s \leftarrow \hat{s}_{fused} > \tau$
  $\hat{M}^{s,ref} \leftarrow crf(I^s, \hat{s}_{fused}, \tau)$
  if $iou(\hat{M}^{s,ref}, M^s) > iou(\hat{M}^s, M^s)$ **then**
    $\hat{M}^q \leftarrow crf(I^q, \hat{q}_{fused}, thresh(\hat{q}_{fused}))$ ▷ apply
  else
    $\hat{M}^q \leftarrow \hat{q}_{fused} > thresh(\hat{q}_{fused})$ ▷ not apply
  end if

Function $iou(\hat{M}, M)$ calculates Eq. 2 given prediction $\hat{M}$ and ground truth M. Function $crf(I, \hat{m}, \tau)$ calculates [4] with unaries from softmax generated as $sigmoid(T(\hat{m}-\tau))$, temperature T=1 for simplicity, input RGB image I, our soft prediction $\hat{m}$ and the calculated threshold τ.

5. Further architectural validation

We test our method under modified configurations. Tab. 2 reports the performance gap under these changes. To overcome the limitations of 1x1 convolutions that do not learn relations to the spatial neighborhood, an intuitive idea would be to use a kernel size larger than 1. However, this quadratically increases the number of learnable parameters from the sparse data and thus performs worse than the 1x1 convolutions. We also find that geometric transformation, in our case the random shearing, is more suitable for establishing the dense consistency than operations like color jitter or blur.

6. Qualitative Comparison

Fig. 6 provides a qualitative comparison of our results. The samples are the same as in Fig. 5, such that intermediate level and final results can be compared.

	Configuration Change	Metric	Deepglobe	ISIC	Chest-Xray	FSS-1000	Avg.
a)	kernelsize 1→3	mIoU	-8.27	-7.90	2.61	-1.60	-3.79
		FB-IoU	-5.57	-14.62	2.01	-2.25	-5.11
b)	out_channels 64→32	mIoU	-0.01	-5.28	-0.09	-1.60	-1.75
		FB-IoU	-0.20	-5.41	-0.20	-1.37	-1.80
c)	out_channels 64→128	mIoU	-0.19	-6.25	-0.69	0.16	-1.74
		FB-IoU	-0.11	-5.74	-0.39	0.26	-1.50
d)	n_epochs 25→10	mIoU	0.21	-6.91	-2.69	-0.48	-2.47
		FB-IoU	0.12	-6.36	-1.82	-0.10	-2.04
e)	Jitter 0→0.3 Shear 20→0	mIoU	-3.67	-3.06	0.05	-2.08	-2.19
		FB-IoU	-3.28	-2.76	0.08	-1.35	-1.83

Table 2. Performance differences under modified configurations of our attached layers. a) Replacing 1x1 convolutions through 3x3, b) Decreasing or c) increasing the number of target channels for task-adapted features, d) Fitting for less epochs, e) Replacing the augmentation method, color jitter instead of affine shearing.

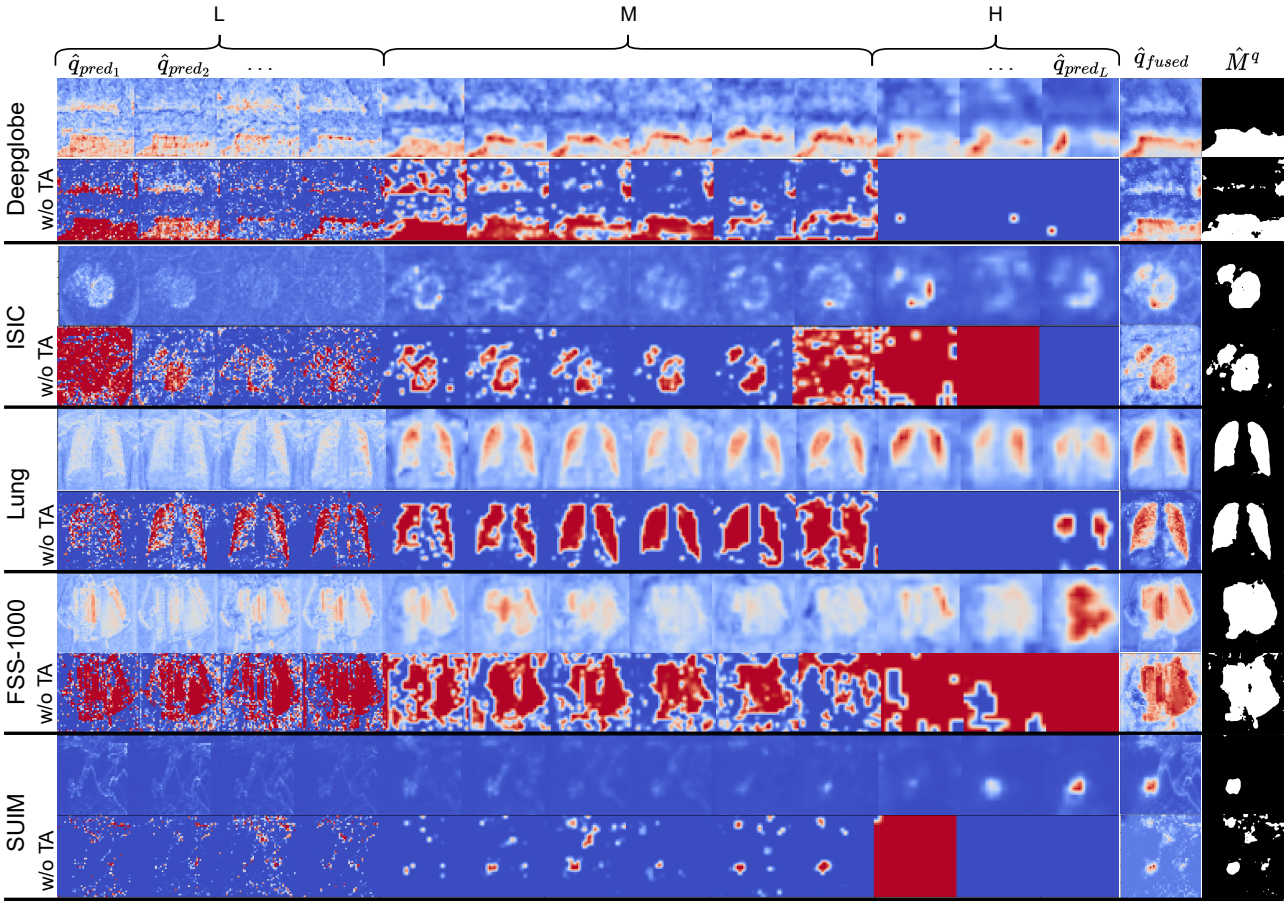

Figure 5. Layer-wise correlation maps $\hat{q}_{pred_l}$, their aggregation $\hat{q}_{fused}$ and binarization $\hat{M}^q$. For each dataset, the upper row represents our maps, whereas the lower row represents the results one would obtain for ResNet features, i.e. when calculating dense comparison on the feature pyramid *before* our attached *T*ask *A*daption layers. Besides the improvement introduced through *TA*, we can observe how considering all levels is important for CD-FSS where the target domain is unknown: *L*ow-level features are meaningful for Deepglobe and ISIC datasets, whereas *H*igh level features are more suitable for FSS and SUIM. Consistent with prior findings in FSS, mid-level layers demonstrate their utility across various datasets. Compare Fig. 6 for the sampled input images.

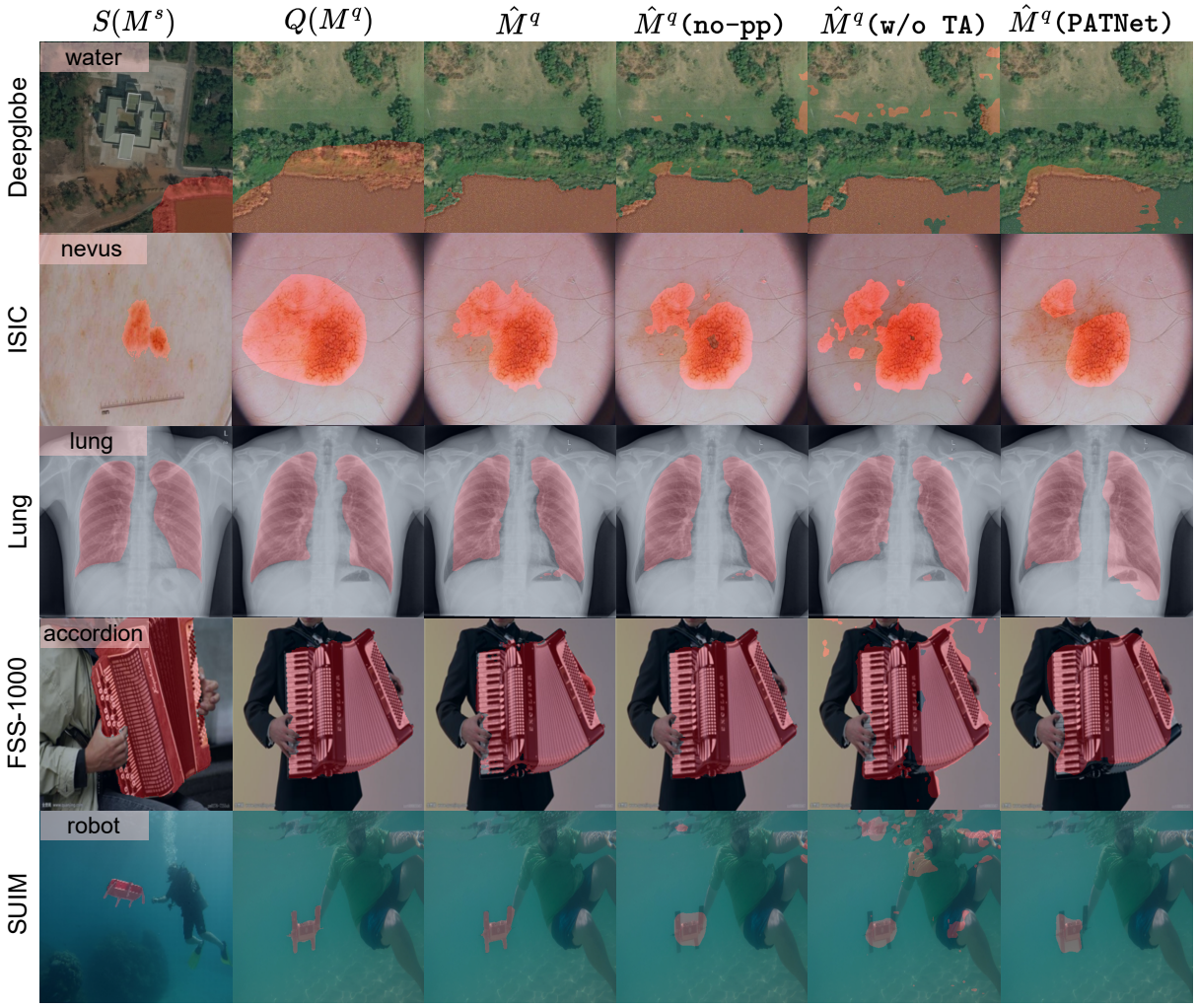

Figure 6. Qualitative comparison of results from proposed method ($\hat{M}^q$), its unrefined variant (*no-pp*), ResNet feature comparison without adaption *(w/o TA)* and previous CD-FSS benchmark SOTA[7] (*PATNet*). We show a 1-shot episode with one *S*upport image for each dataset.

7. Detailed Adaption and Inference Procedure

In Sec. 4.3 of the main paper two operational modes were discussed. On the one hand, the standard evaluation in FSS, where each test task is processed independently, without any knowledge of a previous task. Consequently, parameters θ of attached layers g are estimated from scratch for every task, corresponding to Algorithm 2. On the other hand, in the *quick-infer* mode, parameters θ are kept constant for a task featuring a previously processed semantic class, corresponding to Algorithm 3. The latter mode is useful for computational efficiency in most real-word applications where the same category should be segmented in multiple images. For example, in the Chest-Xray dataset there is only one class, then one can run Algorithm 2 once and predict all further images only with Algorithm 3. If there are tasks with different classes, then for each class parameters θ are fitted once and stored for further episodes of the same class. Table 3 documents runtimes per operation. In the quick-infer mode, the remaining load is primarily merely the backbone, the prediction can run at 27|18 fps for (1|5)-shot as against 15|3 fps of baseline[5], shown in Table 4. [5] is similar in architecture to PATNet[7] and HSNet[8], but requires no test-time fine-tuning(TFI[7]) stage. Tesla P100 was chosen for convenience, though alternative hardware may better suit local target applications. Discrepancy between x50 factor reported in Sec. 4.3 is due to dataloader and metrics overhead. Runtimes should be taken as initial reference only as implementations are not optimized; exemplary, improvements could be achieved by transferring thresholding to GPU, replacing the here not considered CPU-based refinement and, for the adaption, by optimizing the loss calculation.

Algorithm 2 Adapt and infer

Require: ImageNet pre-trained ResNet params Φ, frozen
Require: Kaiming uniform initialized params $\{\theta_l\}_{l=1}^{L}$ of g

Require: $Task = \{I^q, \{I_i^s, M_i^s\}_{i=1}^k\}$ ▷ k-shot input task

// Apply augmentation
$I^{q,aug} \leftarrow augment(I^q)$
$\{I_i^{s,aug}\}_{i=1}^k \leftarrow \{augment(I_i^s)\}_{i=1}^k$

// Forward pass through the backbone
$F^q \leftarrow ResNet(I^q; \Phi))$
$\{F_i^s\}_{i=1}^k \leftarrow \{ResNet(I_i^s; \Phi)\}_{i=1}^k$
$F^{q,aug} \leftarrow ResNet(I^{q,aug}; \Phi)$
$\{F_i^{s,aug}\}_{i=1}^k \leftarrow \{ResNet(I_i^{s,aug}; \Phi)\}_{i=1}^k$

// Layer-wise adaption
for layer $l \leftarrow 1$ **to** L **do**
  for epoch $e \leftarrow 1$ **to** n_{epochs} **do**
    // Forward pass through attached layers
    $\hat{F}^q \leftarrow g_l(F^q; \theta_l)$
    $\{\hat{F}_i^s\}_{i=1}^k \leftarrow \{g_l(F_i^s; \theta_l)\}_{i=1}^k$
    $\hat{F}^{q,aug} \leftarrow g_l(F^{q,aug}; \theta_l)$
    $\{\hat{F}_i^{s,aug}\}_{i=1}^k \leftarrow \{g_l(F_i^{s,aug}; \theta_l)\}_{i=1}^k$

    Evaluate $\mathcal{L}(g_l)$ with Eq. 5 from main paper
    Update θ_l with SGD: $\theta_l \leftarrow \theta_l - \alpha \bigtriangledown_{\theta_l} \mathcal{L}(g_l)$
  end for
  $\hat{q}_{pred_l} \leftarrow attention(\hat{F}^q, \hat{F}^s, M^s)$ ▷ compare, Eq.6
end for
$\hat{q}_{fused} \leftarrow \frac{1}{L} \sum_l upsample(\hat{q}_{pred_l})$ ▷ fuse, Eq. 7
$\hat{M}^q \leftarrow thresh(\hat{q}_{fused})$ ▷ binary pred., Eq. 8

Algorithm 3 Infere only (quick-infer)

Require: ImageNet pre-trained ResNet params Φ, frozen
Require: $\{\theta_l\}_{l=1}^{L}$ fitted by Algorithm 2, now frozen

Require: $Task = \{I^q, \{I_i^s, M_i^s\}_{i=1}^k\}$ ▷ k-shot input task

// Forward pass through the backbone
$F^q \leftarrow ResNet(I^q; \Phi))$
$\{F_i^s\}_{i=1}^k \leftarrow \{ResNet(I_i^s; \Phi)\}_{i=1}^k$

// Layer-wise adaption
for layer $l \leftarrow 1$ **to** L **do**

  // Forward pass through attached layers
  $\hat{F}^q \leftarrow g_l(F^q; \theta_l)$
  $\{\hat{F}_i^s\}_{i=1}^k \leftarrow \{g_l(F_i^s; \theta_l)\}_{i=1}^k$

  $\hat{q}_{pred_l} \leftarrow attention(\hat{F}^q, \hat{F}^s, M^s)$ ▷ compare, Eq.6
end for
$\hat{q}_{fused} \leftarrow \frac{1}{L} \sum_l upsample(\hat{q}_{pred_l})$ ▷ fuse, Eq. 7
$\hat{M}^q \leftarrow thresh(\hat{q}_{fused})$ ▷ binary pred., Eq. 8

		Lines							
shot	Alg.	1+	3+	5+	9+	11+	13+	16	18+
1	2	4	20	21	1	1	11	1	4
	3	-	20	-	1	-	-	1	4
5	2	10	35	63	1	1	16	1	4
	3	-	35	-	1	-	-	1	4

Table 3. Runtime per line execution in pseudocode, milliseconds, + indicates inclusion of the following line. Note that lines within the loop are executed more than once.

shot	baseline[5]	Alg. 2	Alg. 3
1	64	3700	36
5	320	5300	54

Table 4. Runtime for predicting one task in milliseconds.

In the pseudocode in Algorithms 2 and 3, operations printed in green are equally performed for both procedures, red lines are specific to the fitting process, equation numbers refer to the main paper, α is the learning rate, L is the number of layers, n_{epochs} is the number of iterations, all specified in the implementation details in Sec 4.1.

References

[1] Malik Boudiaf, Hoel Kervadec, Imtiaz Masud Ziko, Pablo Piantanida, Ismail Ben Ayed, and José Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13974–13983, 2021. 4

[2] Hao Chen, Yonghan Dong, Zheming Lu, Yunlong Yu, and Jungong Han. Pixel matching network for cross-domain few-shot segmentation. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 978–987, 2024. 3

[3] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse

the earth through satellite images. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 172–17209, 2018. 2

[4] https://github.com/lucasb-eyer/pydensecrf. 5

[5] Xinyang Huang, Chuanglu Zhu, and Wenkai Chen. Restnet: Boosting cross-domain few-shot segmentation with residual transformation network. In *British Machine Vision Conference*, 2023. 7, 8

[6] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011. 5

[7] Shuo Lei, Xuchao Zhang, Jianfeng He, Fanglan Chen, Bowen Du, and Chang-Tien Lu. Cross-domain few-shot semantic segmentation. In *European Conference on Computer Vision*, 2022. 2, 3, 7

[8] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmenation. *IEEE/CVF International Conference on Computer Vision*, pages 6921–6932, 2021. 2, 3, 4, 7

[9] Nobuyuki Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66, 1979. 4

[10] Xinyu Shi, Dong Wei, Yu Zhang, Donghuan Lu, Munan Ning, Jiashun Chen, Kai Ma, and Yefeng Zheng. Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In *European Conference on Computer Vision*, pages 151–168, 2022. 4