Hypercorrelation Squeeze for Few-Shot Segmentation
—Supplementary Material—

Juhong Min Dahyun Kang Minsu Cho
Pohang University of Science and Technology (POSTECH), South Korea
http://cvlab.postech.ac.kr/research/HSNet/

In this supplementary material, we provide additional implementation details, experimental results, analyses of the proposed method, and extensive qualitative results in many challenging cases.

1. Additional implementation details

For the backbone networks, we employ VGG [16] and ResNet [4] families pre-trained on ImageNet [2], *e.g.*, VGG16, ResNet50, and ResNet101. For the VGG16 backbone, we extract features after every conv layer in the last two building blocks: from `conv4_x` to `conv5_x`, and after the last maxpooling layer. For the ResNet backbones, we extract features at the end of each bottleneck before ReLU activation: from `conv3_x` to `conv5_x`. This feature extracting scheme results in 3 pyramidal layers ($P = 3$) for every backbone. We set spatial sizes of both support and query images to 400×400, *i.e.*, $H, W = 400$, thus having $H_1, W_1 = 50$, $H_2, W_2 = 25$, and $H_3, W_3 = 13$ for both ResNet50 and ResNet101 bakcbones and $H_1, W_1 = 50$, $H_2, W_2 = 25$, and $H_3, W_3 = 12$ for the VGG16 backbone. The network is implemented in PyTorch [11] and optimized using Adam [5] with learning rate of 1e-3. We train our model with batch size of 20, 40, and 20 for PASCAL-5^i, COCO-20^i, and FSS-1000 respectively. We freeze the pre-trained backbone networks to prevent them from learning class-specific representations of the training data. The intermediate tensor dimensions, the number of parameters of each layers and other additional details of the network are demonstrated in Tab. S5, S6, and S7 for respective backbones of VGG16, ResNet50, and ResNet101.

2. Additional results and analyses

Additional K-shot results. Following the work of [1, 17, 20], we conduct K-shot experiments with $K \in \{1, 5, 10\}$. Table S1 compares our results with the recent methods [1, 17, 20] on PASCAL-5^i and COCO-20^i. The significant performance improvements on both datasets clearly indicate the effectiveness of our approach. Achieving 2.5%p and 4.6%p mIoU improvements over the previous best method [1] on respective PASCAL-5^i and COCO-20^i, our model again sets a new state of the art in 10-shot setting as well, showing notable improvements with larger K.

Method	PASCAL-5^i			COCO-20^i		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
RPMM [20]	56.3	57.3	57.6	30.6	35.5	33.1
PFENet [17]	60.8	61.9	62.1	35.8	39.0	39.7
RePRI [1]	59.7	66.6	68.1	34.1	41.6	44.1
HSNet (ours)	**64.0**	**69.5**	**70.6**	**39.2**	**46.9**	**48.7**

Table S1: Results on K-shot with ResNet50 backbone network where $K \in \{1, 5, 10\}$. The results of other methods are from [1].

Methods	1-shot					5-shot				
	5^0	5^1	5^2	5^3	mean	5^0	5^1	5^2	5^3	mean
$\mathbf{C}_p^{\text{shallow}}$	57.1	64.7	57.5	57.0	59.1	63.7	69.8	66.0	63.4	65.7
$\mathbf{C}_p^{\text{deep}}$	60.6	68.6	58.2	59.2	61.7	66.7	72.0	65.9	65.4	67.5
$\mathbf{C}_p$ (ours)	**67.3**	**72.3**	**62.0**	**63.1**	**66.2**	**71.8**	**74.4**	**67.0**	**68.3**	**70.4**

Table S2: Numerical results of Figure 6 of our main paper. All experiments are performed with ResNet101 backbone [4].

Methods	1-shot					5-shot				
	5^0	5^1	5^2	5^3	mean	5^0	5^1	5^2	5^3	mean
$\mathcal{C}^{(3)}$	54.3	62.8	52.0	52.8	55.5	60.2	67.0	59.4	59.9	61.6
$\mathcal{C}^{(2:3)}$	64.3	70.3	60.5	60.4	63.9	69.7	73.2	65.2	64.9	68.2
$\mathcal{C}$ (ours)	**67.3**	**72.3**	**62.0**	**63.1**	**66.2**	**71.8**	**74.4**	**67.0**	**68.3**	**70.4**

Table S3: Numerical results of Figure 7 of our main paper. All experiments are performed with ResNet101 backbone [4].

Numerical comparisons of ablation study. We tabularize Figures 6 and 7 in our main paper, *e.g.*, ablation study on hypercorrelations and pyramidal layers, in Tables S2 and S3 respectively. Achieving 4.5%p mIoU improvements over $\mathbf{C}_p^{\text{deep}}$, our method clearly benefits from diverse feature correlations from multi-level CNN layers ($\mathbf{C}_p$) as seen in Tab. S2. A large performance gap between $\mathcal{C}^{(2:3)}$ and $\mathcal{C}^{(3)}$ in Tab. S3 (63.9 vs. 55.5) reveals that the intermediary second pyramidal layer ($p = 2$) is especially effective in robust mask prediction compared to the first pyramidal layer ($p = 1$).

Evaluation results without using `ignore_label` on PASCAL-5^i. The benchmarks of PASCAL-5^i [14], COCO-20^i [7], and FSS-1000 [6] consist of segmentation mask

Backbone network	Methods	1-shot 5^0	5^1	5^2	5^3	mIoU	FB-IoU	5-shot 5^0	5^1	5^2	5^3	mIoU	FB-IoU	# learnable params
VGG16 [16]	SG-One [24]	40.2	58.4	48.4	38.4	46.3	63.1	41.9	58.6	48.6	39.4	47.1	65.9	19.0M
	CRNet [8]	-	-	-	-	**55.2**	66.4	-	-	-	-	58.5	71.0	-
	HSNet (ours)	**53.6**	**61.7**	**55.0**	**50.3**	**55.2**	**70.5**	**58.3**	**64.7**	**58.9**	**54.6**	**59.1**	**73.5**	**2.6M**
ResNet50 [4]	CANet [23]	52.5	65.9	51.3	51.9	55.4	66.2	55.5	67.8	51.9	53.2	57.1	69.6	19.0M
	RPMM [20]	55.2	**66.9**	52.6	50.7	56.3	-	56.3	67.3	54.5	51.0	57.3	-	19.7M
	CRNet [8]	-	-	-	-	55.7	66.8	-	-	-	-	58.8	71.5	-
	HSNet (ours)	**57.4**	66.8	**55.8**	**56.5**	**59.1**	**73.8**	**62.6**	**69.2**	**62.5**	**62.4**	**64.2**	**77.4**	**2.6M**
ResNet101 [4]	HSNet (ours)	**60.1**	**67.8**	**57.3**	**59.0**	**61.1**	**74.4**	**63.9**	**69.9**	**62.0**	**63.6**	**64.8**	**77.1**	**2.6M**

Table S4: Evaluation results on PASCAL-5^i [14] benchmark in mIoU and FB-IoU evaluation metrics without the use of `ignore_label`. The results of other methods are from [8, 9, 17, 18, 20].

annotations in which each pixel is labeled with either background or one of the predefined object categories. As pixel-wise segmentation near object boundaries is ambiguous to perform even for human annotators, PASCAL-5^i uses a special kind of label called `ignore_label` which marks pixel regions ignored during training and evaluation to mitigate the ambiguity*.

Most recent few-shot segmentation work [1, 9, 10, 12, 14, 15, 17, 18, 19, 22] adopt this evaluation criteria but we found that some methods [8, 20, 23, 24] do not utilize `ignore_label` in their evaluations. Therefore, the methods are unfairly evaluated as fine-grained mask prediction near object boundaries is one of the most challenging part in segmentation problem. For fair comparisons, we intentionally exclude the methods of [8, 20, 23, 24] from Tab. 1 of our main paper and compare the results of our model evaluated without the use of `ignore_label` with those methods [8, 20, 23, 24]. The results are summarized in Tab. S4. Even without using `ignore_label`, the proposed method sets a new state of the art with ResNet50 backbone, outperforming the previous best methods of [20] and [8] by (1-shot) 2.8%p and (5-shot) 5.4%p respectively. With VGG16 backbone, our method performs comparably effective to the previous best method [8] while having the smallest learnable parameters.

3. Full derivation of center-pivot 4D conv

In this section, we extend Sec. 4.4 of our main paper to provide a complete derivation of the center-pivot 4D convolution. Note that a typical 4D convolution parameterized by a kernel $k \in \mathbb{R}^{\hat{k}\times\hat{k}\times\hat{k}\times\hat{k}}$ on a correlation tensor $c \in \mathbb{R}^{H\times W\times H\times W}$ at position $(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^4$ is formulated as

$$(c * k)(\mathbf{x}, \mathbf{x}') = \sum_{(\mathbf{p},\mathbf{p}')\in\mathcal{P}(\mathbf{x},\mathbf{x}')} c(\mathbf{p}, \mathbf{p}')k(\mathbf{p} - \mathbf{x}, \mathbf{p}' - \mathbf{x}'), \tag{1}$$

where $\mathcal{P}(\mathbf{x}, \mathbf{x}')$ denotes a set of neighbourhood regions within the local 4D window centered on position $(\mathbf{x}, \mathbf{x}')$, *i.e.*, $\mathcal{P}(\mathbf{x}, \mathbf{x}') = \mathcal{P}(\mathbf{x}) \times \mathcal{P}(\mathbf{x}')$ as visualized in Fig. S1. Now we design a light-weight, efficient 4D convolution via a reasonable weight-sparsification; from a set of neighborhood positions within a local 4D window of interest, our kernel aims to disregard a large number of activations located at fairly insignificant positions in the 4D window, thereby focusing only on a small subset of relevant activations for capturing complex patterns in the correlation tensor. Specifically, we consider activations at positions that ***pivots*** either one of 2-dimensional ***centers***, *e.g.*, $\mathbf{x}$ or $\mathbf{x}'$, as the foremost influential ones. Given 4D position $(\mathbf{x}, \mathbf{x}')$, we collect its neighbors if and only if they are adjacent to either $\mathbf{x}$ or $\mathbf{x}'$ in its corresponding 2D subspace and define two respective sets as

$$\mathcal{P}_c(\mathbf{x}, \mathbf{x}') = \{(\mathbf{p}, \mathbf{p}') \in \mathcal{P}(\mathbf{x}, \mathbf{x}') : \mathbf{p} = \mathbf{x}\}, \tag{2}$$

and

$$\mathcal{P}_{c'}(\mathbf{x}, \mathbf{x}') = \{(\mathbf{p}, \mathbf{p}') \in \mathcal{P}(\mathbf{x}, \mathbf{x}') : \mathbf{p}' = \mathbf{x}'\}. \tag{3}$$

The set of center-pivot neighbours $\mathcal{P}_{\text{CP}}(\mathbf{x}, \mathbf{x}')$ is defined as a union of the two subsets:

$$\mathcal{P}_{\text{CP}}(\mathbf{x}, \mathbf{x}') = \mathcal{P}_c(\mathbf{x}, \mathbf{x}') \cup \mathcal{P}_{c'}(\mathbf{x}, \mathbf{x}'). \tag{4}$$

Based on this small subset of neighbors, center-pivot 4D (CP 4D) convolution can be formulated as a union of two separate 4D convolutions:

$$(c * k_{\text{CP}})(\mathbf{x}, \mathbf{x}') = (c * k_c)(\mathbf{x}, \mathbf{x}') + (c * k_{c'})(\mathbf{x}, \mathbf{x}'), \tag{5}$$

*The use of `ignore_label` was originally adopted in PASCAL VOC dataset [3]. The same evaluation criteria is naturally transferred to PASCAL-5^i [14] as it is created from PASCAL VOC.

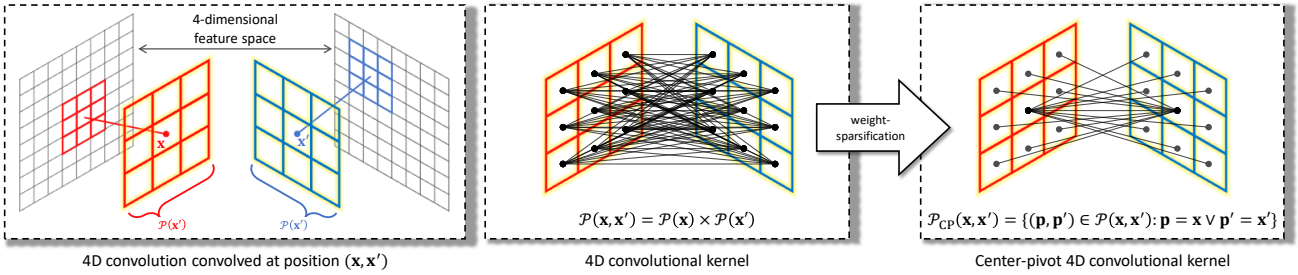

Figure S1: 4D convolution (left) and weights of 4D kernel [13, 21] (middle) and center-pivot 4D kernel (right). Each black wire that connects two different pixel locations represent a single weight of the 4D kernel. The kernel size in this example is $(3, 3, 3, 3)$, *i.e.*, $\hat{k} = 3$.

where k_c and $k_{c'}$ are 4D kernels with their respective neighbours $\mathcal{P}_c(\mathbf{x}, \mathbf{x}')$ and $\mathcal{P}_{c'}(\mathbf{x}, \mathbf{x}')$. Now consider below

$$\begin{aligned}(c * k_c)(\mathbf{x}, \mathbf{x}') &= \sum_{(\mathbf{p}, \mathbf{p}') \in \mathcal{P}_c(\mathbf{x}, \mathbf{x}')} c(\mathbf{p}, \mathbf{p}') k(\mathbf{p} - \mathbf{x}, \mathbf{p}' - \mathbf{x}') \\ &= \sum_{\mathbf{p}' \in \mathcal{P}(\mathbf{x}')} c(\mathbf{x}, \mathbf{p}') k(\mathbf{x} - \mathbf{x}, \mathbf{p}' - \mathbf{x}') \\ &= \sum_{\mathbf{p}' \in \mathcal{P}(\mathbf{x}')} c(\mathbf{x}, \mathbf{p}') k(\mathbf{0}, \mathbf{p}' - \mathbf{x}') \\ &= \sum_{\mathbf{p}' \in \mathcal{P}(\mathbf{x}')} c(\mathbf{x}, \mathbf{p}') k_c^{\text{2D}}(\mathbf{p}' - \mathbf{x}'), \end{aligned} \tag{6}$$

which is equivalent to a convolution with a 2D kernel $k_c^{\text{2D}} = k(\mathbf{0}, :) \in \mathbb{R}^{\hat{k} \times \hat{k}}$ performed on 2D slice of the 4D tensor $c(\mathbf{x}, :)$. Similarly,

$$\begin{aligned}(c * k_{c'})(\mathbf{x}, \mathbf{x}') &= \sum_{(\mathbf{p}, \mathbf{p}') \in \mathcal{P}_{c'}(\mathbf{x}, \mathbf{x}')} c(\mathbf{p}, \mathbf{p}') k(\mathbf{p} - \mathbf{x}, \mathbf{p}' - \mathbf{x}') \\ &= \sum_{\mathbf{p} \in \mathcal{P}(\mathbf{x})} c(\mathbf{p}, \mathbf{x}') k(\mathbf{p} - \mathbf{x}, \mathbf{x}' - \mathbf{x}') \\ &= \sum_{\mathbf{p} \in \mathcal{P}(\mathbf{x})} c(\mathbf{p}, \mathbf{x}') k(\mathbf{p} - \mathbf{x}, \mathbf{0}) \\ &= \sum_{\mathbf{p} \in \mathcal{P}(\mathbf{x})} c(\mathbf{p}, \mathbf{x}') k_{c'}^{\text{2D}}(\mathbf{p} - \mathbf{x}), \end{aligned} \tag{7}$$

where $k_{c'}^{\text{2D}} = k(:, \mathbf{0}) \in \mathbb{R}^{\hat{k} \times \hat{k}}$. Based on above derivations, we rewrite Eqn. 5 as follows

$$\begin{aligned}(c * k_{\text{CP}})(\mathbf{x}, \mathbf{x}') = &\sum_{\mathbf{p}' \in \mathcal{P}(\mathbf{x}')} c(\mathbf{x}, \mathbf{p}') k_c^{\text{2D}}(\mathbf{p}' - \mathbf{x}') \\ &+ \sum_{\mathbf{p} \in \mathcal{P}(\mathbf{x})} c(\mathbf{p}, \mathbf{x}') k_{c'}^{\text{2D}}(\mathbf{p} - \mathbf{x}), \end{aligned} \tag{8}$$

which performs two different convolutions on separate 2D subspaces, having a linear complexity.

4. Qualitative results

We present additional qualitative results on PASCAL-5^i [14], COCO-20^i [10], and FSS-1000 [6] benchmark datasets. All the qualitative results are best viewed in electronic forms. Example results in presence of large scale-differences, truncations, and occlusions are shown in Fig. S2, S3, and S4. Figure S5 visualizes model predictions under large illumination-changes in support and query images. Figure S6 visualizes some sample predictions given exceptionally small objects in either support or query images. As seen in Fig. S7, we found that our model sometimes predicts more reliable segmentation masks than ground-truth ones. Some qualitative results in presence of large intra-class variations and noisy clutters in background are shown in Fig. S8 and S9. Given only a single support image-annotation pair, our model effectively segments multiple instances in a query image as visualized in Fig. S10. Figure S11 shows representative failure cases; our model fails to localize target objects in presence of severe occlusions, intra-class variances and extremely tiny support (or query) objects. As seen in Fig. S12, the model predictions become much reliable given multiple support image-mask pairs, *i.e.*, $K > 1$. Figure S13 shows some example results on FSS-1000 dataset.

Results without support feature masking. As demonstrated in Sec. 5.1 of our main paper, we conduct experiments without support feature masking (Eqn. 1 of our main paper), similarly to co-segmentation problem with stronger demands for generalizibility. Figure S14 visualizes some example results on PASCAL-5^i dataset. Even without the use of support masks (in both training and testing), our model effectively segments target instances in query images. The results indicate that learning patterns of feature correlations from multiple visual aspects is effective in fine-grained segmentation as well as identifying 'common' instances in the support and query images.

Layer	Input		Output		Operation	# params.
VGG16 Backbone	I^{q}	(3, 400, 400)	$\{\mathbf{F}_l^{\mathrm{q}}\}_{l=1}^{7}$	(512, 12, 12) × 1 (512, 25, 25) × 3 (512, 50, 50) × 3	SERIES OF 2D CONVS	14.7M (frozen)
	I^{s}	(3, 400, 400)	$\{\mathbf{F}_l^{\mathrm{s}}\}_{l=1}^{7}$	(512, 12, 12) × 1 (512, 25, 25) × 3 (512, 50, 50) × 3		
Masking Layer	$\{\mathbf{F}_l^{\mathrm{s}}\}_{l=1}^{7}$	(512, 12, 12) × 1 (512, 25, 25) × 3 (512, 50, 50) × 3	$\{\hat{\mathbf{F}}_l^{\mathrm{s}}\}_{l=1}^{7}$	(512, 12, 12) × 1 (512, 25, 25) × 3 (512, 50, 50) × 3	BILINEAR INTERPOLATION HADAMARD PRODUCT	-
	$\mathbf{M}^{\mathrm{s}}$	(1, 400, 400)				
Correlation Layer	$\{\mathbf{F}_l^{\mathrm{q}}\}_{l=1}^{7}$	(512, 12, 12) × 1 (512, 25, 25) × 3 (512, 50, 50) × 3	$\{\mathbf{C}_p\}_{p=1}^{3}$	(1, 12, 12, 12, 12) (3, 25, 25, 25, 25) (3, 50, 50, 50, 50)	COSINE SIMILARITY	-
	$\{\hat{\mathbf{F}}_l^{\mathrm{s}}\}_{l=1}^{7}$	(512, 12, 12) × 1 (512, 25, 25) × 3 (512, 50, 50) × 3				
Squeezing Block f_3^{sqz}	$\mathbf{C}_3$	(1, 12, 12, 12, 12)	$\mathbf{C}_3^{\mathrm{sqz}}$	(128, 12, 12, 2, 2)	(CP 4D CONV GROUP NORM RELU) × 3	167K
Squeezing Block f_2^{sqz}	$\mathbf{C}_2$	(3, 25, 25, 25, 25)	$\mathbf{C}_2^{\mathrm{sqz}}$	(128, 25, 25, 2, 2)	(CP 4D CONV GROUP NORM RELU) × 3	169K
Squeezing Block f_1^{sqz}	$\mathbf{C}_1$	(3, 50, 50, 50, 50)	$\mathbf{C}_1^{\mathrm{sqz}}$	(128, 50, 50, 2, 2)	(CP 4D CONV GROUP NORM RELU) × 3	202K
Mixing Block f_2^{mix}	$\mathbf{C}_3^{\mathrm{sqz}}$	(128, 12, 12, 2, 2)	$\mathbf{C}_2^{\mathrm{mix}}$	(128, 25, 25, 2, 2)	BILINEAR INTERPOLATION ELEMENT-WISE ADDITION (CP 4D CONV GROUP NORM RELU) × 3	886K
	$\mathbf{C}_2^{\mathrm{sqz}}$	(128, 25, 25, 2, 2)				
Mixing Block f_1^{mix}	$\mathbf{C}_2^{\mathrm{mix}}$	(128, 25, 25, 2, 2)	$\mathbf{C}_1^{\mathrm{mix}}$	(128, 50, 50, 2, 2)	BILINEAR INTERPOLATION ELEMENT-WISE ADDITION (CP 4D CONV GROUP NORM RELU) × 3	886K
	$\mathbf{C}_1^{\mathrm{sqz}}$	(128, 50, 50, 2, 2)				
Pooling Layer	$\mathbf{C}_1^{\mathrm{mix}}$	(128, 50, 50, 2, 2)	$\mathbf{Z}$	(128, 50, 50)	AVERAGE-POOLING	-
Decoder Layer	$\mathbf{Z}$	(128, 50, 50)	$\hat{\mathbf{M}}^{\mathrm{q}}$	(2, 400, 400)	SERIES OF 2D CONVS WITH BILINEAR INTERPOLATION	259K

Table S5: Hypercorrelation Squeeze Networks (HSNet) with VGG16 [16] backbone network. The reported number of parameters in VGG16 backbone network (14.7M) excludes those in fully-connected layers (unused in our model). The total number of 'learnable' parameters amounts to 2.6M. The number of intermediate features extracted from backbone network amounts to 7, *i.e.*, $L = 7$. The CP 4D CONV refers to the proposed center-pivot 4D convolution.

Layer	Input		Output		Operation	# params.
ResNet50 Backbone	I^{q}	(3, 400, 400)	$\{\mathbf{F}_l^{\mathrm{q}}\}_{l=1}^{13}$	(2048, 13, 13) × 3 (1024, 25, 25) × 6 (512, 50, 50) × 4	SERIES OF 2D CONVS	23.6M (frozen)
	I^{s}	(3, 400, 400)	$\{\mathbf{F}_l^{\mathrm{s}}\}_{l=1}^{13}$	(2048, 13, 13) × 3 (1024, 25, 25) × 6 (512, 50, 50) × 4		
Masking Layer	$\{\mathbf{F}_l^{\mathrm{s}}\}_{l=1}^{13}$	(2048, 13, 13) × 3 (1024, 25, 25) × 6 (512, 50, 50) × 4	$\{\hat{\mathbf{F}}_l^{\mathrm{s}}\}_{l=1}^{13}$	(2048, 13, 13) × 3 (1024, 25, 25) × 6 (512, 50, 50) × 4	BILINEAR INTERPOLATION HADAMARD PRODUCT	-
	$\mathbf{M}^{\mathrm{s}}$	(1, 400, 400)				
Correlation Layer	$\{\mathbf{F}_l^{\mathrm{q}}\}_{l=1}^{13}$	(2048, 13, 13) × 3 (1024, 25, 25) × 6 (512, 50, 50) × 4	$\{\mathbf{C}_p\}_{p=1}^{3}$	(3, 13, 13, 13, 13) (6, 25, 25, 25, 25) (4, 50, 50, 50, 50)	COSINE SIMILARITY	-
	$\{\hat{\mathbf{F}}_l^{\mathrm{s}}\}_{l=1}^{13}$	(2048, 13, 13) × 3 (1024, 25, 25) × 6 (512, 50, 50) × 4				
Squeezing Block f_3^{sqz}	$\mathbf{C}_3$	(3, 13, 13, 13, 13)	$\mathbf{C}_3^{\mathrm{sqz}}$	(128, 13, 13, 2, 2)	$\begin{pmatrix}\text{CP 4D CONV}\\ \text{GROUP NORM}\\ \text{RELU}\end{pmatrix} \times 3$	168K
Squeezing Block f_2^{sqz}	$\mathbf{C}_2$	(6, 25, 25, 25, 25)	$\mathbf{C}_2^{\mathrm{sqz}}$	(128, 25, 25, 2, 2)	$\begin{pmatrix}\text{CP 4D CONV}\\ \text{GROUP NORM}\\ \text{RELU}\end{pmatrix} \times 3$	172K
Squeezing Block f_1^{sqz}	$\mathbf{C}_1$	(4, 50, 50, 50, 50)	$\mathbf{C}_1^{\mathrm{sqz}}$	(128, 50, 50, 2, 2)	$\begin{pmatrix}\text{CP 4D CONV}\\ \text{GROUP NORM}\\ \text{RELU}\end{pmatrix} \times 3$	203K
Mixing Block f_2^{mix}	$\mathbf{C}_3^{\mathrm{sqz}}$	(128, 13, 13, 2, 2)	$\mathbf{C}_2^{\mathrm{mix}}$	(128, 25, 25, 2, 2)	BILINEAR INTERPOLATION ELEMENT-WISE ADDITION $\begin{pmatrix}\text{CP 4D CONV}\\ \text{GROUP NORM}\\ \text{RELU}\end{pmatrix} \times 3$	886K
	$\mathbf{C}_2^{\mathrm{sqz}}$	(128, 25, 25, 2, 2)				
Mixing Block f_1^{mix}	$\mathbf{C}_2^{\mathrm{mix}}$	(128, 25, 25, 2, 2)	$\mathbf{C}_1^{\mathrm{mix}}$	(128, 50, 50, 2, 2)	BILINEAR INTERPOLATION ELEMENT-WISE ADDITION $\begin{pmatrix}\text{CP 4D CONV}\\ \text{GROUP NORM}\\ \text{RELU}\end{pmatrix} \times 3$	886K
	$\mathbf{C}_1^{\mathrm{sqz}}$	(128, 50, 50, 2, 2)				
Pooling Layer	$\mathbf{C}_1^{\mathrm{mix}}$	(128, 50, 50, 2, 2)	$\mathbf{Z}$	(128, 50, 50)	AVERAGE-POOLING	-
Decoder Layer	$\mathbf{Z}$	(128, 50, 50)	$\hat{\mathbf{M}}^{\mathrm{q}}$	(2, 400, 400)	SERIES OF 2D CONVS WITH BILINEAR INTERPOLATION	259K

Table S6: Hypercorrelation Squeeze Networks (HSNet) with ResNet50 [4] backbone network. The reported number of parameters in ResNet50 backbone network (23.6M) excludes those in fully-connected layers (unused in our model). The total number of 'learnable' parameters amounts to 2.6M. The number of intermediate features extracted from backbone network amounts to 13, *i.e.*, $L = 13$.

Layer	Input		Output		Operation	# params.
ResNet101 Backbone	I^{q}	(3, 400, 400)	$\{\mathbf{F}_l^{\text{q}}\}_{l=1}^{30}$	(2048, 13, 13) × 3 (1024, 25, 25) × 23 (512, 50, 50) × 4	SERIES OF 2D CONVS	42.6M (frozen)
	I^{s}	(3, 400, 400)	$\{\mathbf{F}_l^{\text{s}}\}_{l=1}^{30}$	(2048, 13, 13) × 3 (1024, 25, 25) × 23 (512, 50, 50) × 4		
Masking Layer	$\{\mathbf{F}_l^{\text{s}}\}_{l=1}^{30}$	(2048, 13, 13) × 3 (1024, 25, 25) × 23 (512, 50, 50) × 4	$\{\hat{\mathbf{F}}_l^{\text{s}}\}_{l=1}^{30}$	(2048, 13, 13) × 3 (1024, 25, 25) × 23 (512, 50, 50) × 4	BILINEAR INTERPOLATION HADAMARD PRODUCT	-
	$\mathbf{M}^{\text{s}}$	(1, 400, 400)				
Correlation Layer	$\{\mathbf{F}_l^{\text{q}}\}_{l=1}^{30}$	(2048, 13, 13) × 3 (1024, 25, 25) × 23 (512, 50, 50) × 4	$\{\mathbf{C}_p\}_{p=1}^{3}$	(3, 13, 13, 13, 13) (23, 25, 25, 25, 25) (4, 50, 50, 50, 50)	COSINE SIMILARITY	-
	$\{\hat{\mathbf{F}}_l^{\text{s}}\}_{l=1}^{30}$	(2048, 13, 13) × 3 (1024, 25, 25) × 23 (512, 50, 50) × 4				
Squeezing Block f_3^{sqz}	$\mathbf{C}_3$	(3, 13, 13, 13, 13)	$\mathbf{C}_3^{\text{sqz}}$	(128, 13, 13, 2, 2)	(CP 4D CONV GROUP NORM RELU) × 3	168K
Squeezing Block f_2^{sqz}	$\mathbf{C}_2$	(23, 25, 25, 25, 25)	$\mathbf{C}_2^{\text{sqz}}$	(128, 25, 25, 2, 2)	(CP 4D CONV GROUP NORM RELU) × 3	185K
Squeezing Block f_1^{sqz}	$\mathbf{C}_1$	(4, 50, 50, 50, 50)	$\mathbf{C}_1^{\text{sqz}}$	(128, 50, 50, 2, 2)	(CP 4D CONV GROUP NORM RELU) × 3	203K
Mixing Block f_2^{mix}	$\mathbf{C}_3^{\text{sqz}}$	(128, 13, 13, 2, 2)	$\mathbf{C}_2^{\text{mix}}$	(128, 25, 25, 2, 2)	BILINEAR INTERPOLATION ELEMENT-WISE ADDITION (CP 4D CONV GROUP NORM RELU) × 3	886K
	$\mathbf{C}_2^{\text{sqz}}$	(128, 25, 25, 2, 2)				
Mixing Block f_1^{mix}	$\mathbf{C}_2^{\text{mix}}$	(128, 25, 25, 2, 2)	$\mathbf{C}_1^{\text{mix}}$	(128, 50, 50, 2, 2)	BILINEAR INTERPOLATION ELEMENT-WISE ADDITION (CP 4D CONV GROUP NORM RELU) × 3	886K
	$\mathbf{C}_1^{\text{sqz}}$	(128, 50, 50, 2, 2)				
Pooling Layer	$\mathbf{C}_1^{\text{mix}}$	(128, 50, 50, 2, 2)	$\mathbf{Z}$	(128, 50, 50)	AVERAGE-POOLING	-
Decoder Layer	$\mathbf{Z}$	(128, 50, 50)	$\hat{\mathbf{M}}^{\text{q}}$	(2, 400, 400)	SERIES OF 2D CONVS WITH BILINEAR INTERPOLATION	259K

Table S7: Hypercorrelation Squeeze Networks (HSNet) with ResNet101 [4] backbone network. The reported number of parameters in ResNet101 backbone network (42.6M) excludes those in fully-connected layers (unused in our model). The total number of 'learnable' parameters amounts to 2.6M. The number of intermediate features extracted from backbone network amounts to 30, *i.e.*, $L = 30$.

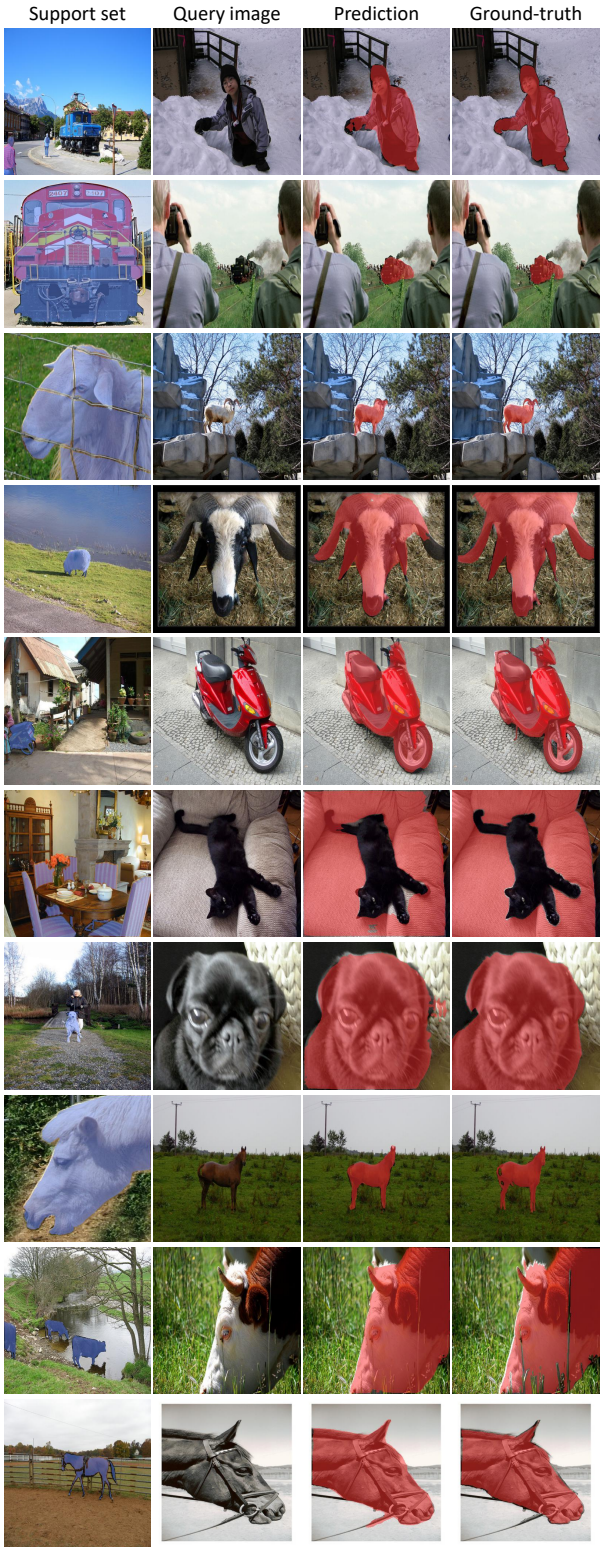

Figure S2: Qualitative (1-shot) results on PASCAL-5^i [14] dataset under large differences in object scales.

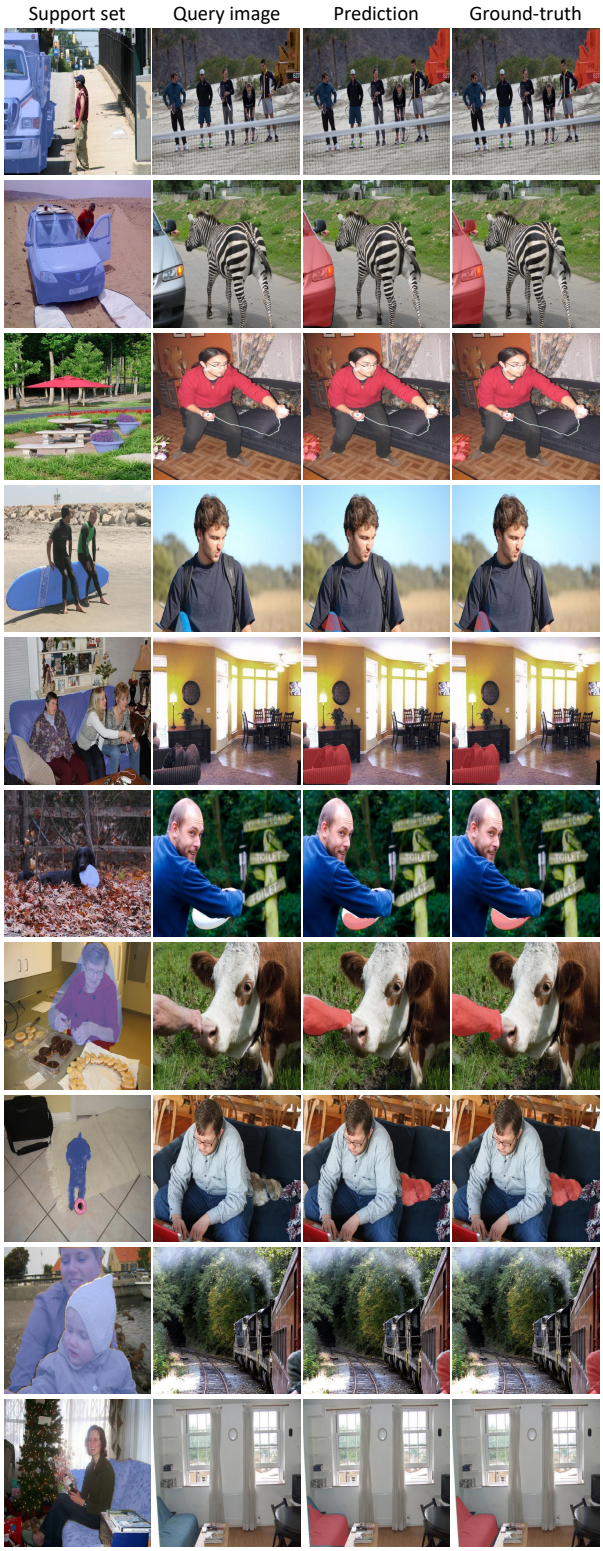

Figure S3: Qualitative (1-shot) results on PASCAL-5^i [14] and COCO-20^i [7] datasets under large truncations and occlusions.

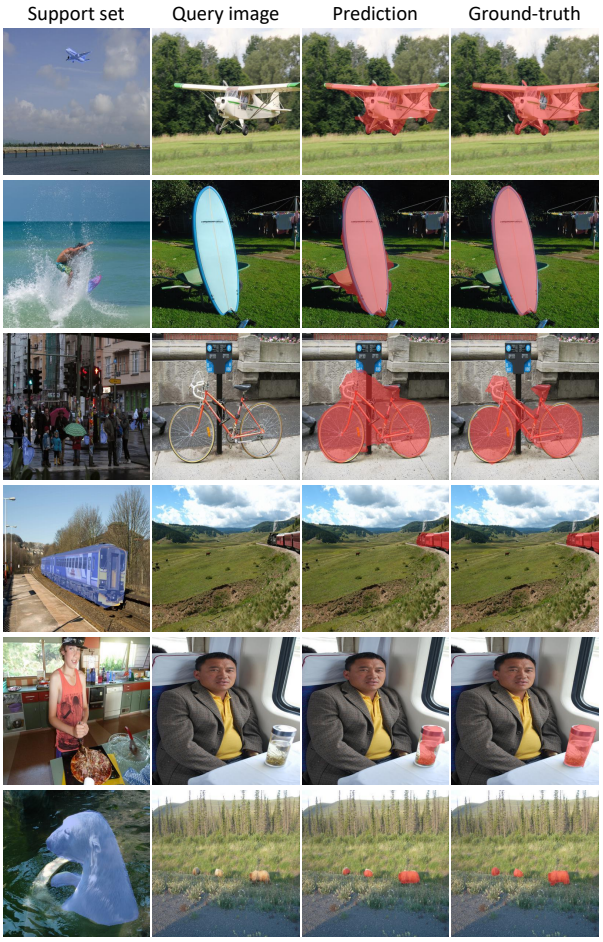

Figure S4: Qualitative (1-shot) results on COCO-20^i [7] dataset under large differences in object scales.

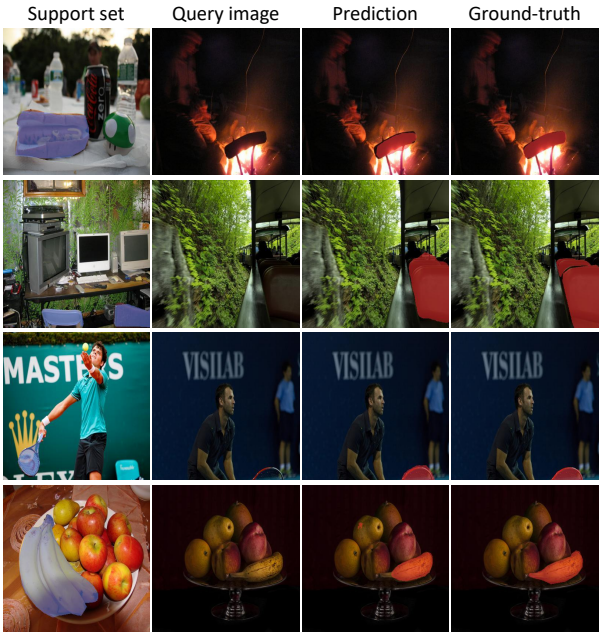

Figure S5: Qualitative (1-shot) results on COCO-20^i [7] dataset under large illumination-changes in support and query images.

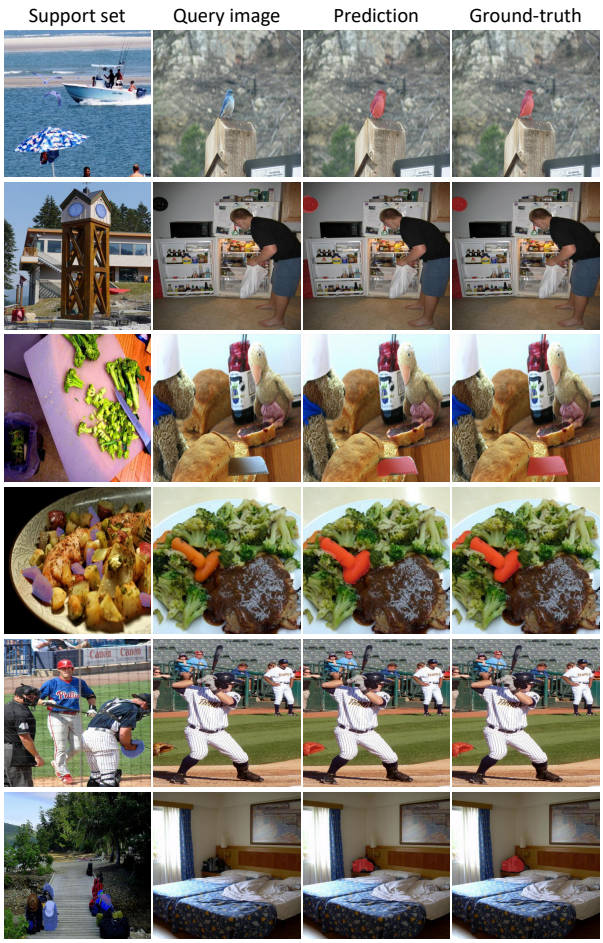

Figure S6: Qualitative (1-shot) results on COCO-20^i [7] dataset with exceptionally small objects.

Figure S7: Our network occasionally predicts more accurate segmentation masks than human-annotated ground-truths.

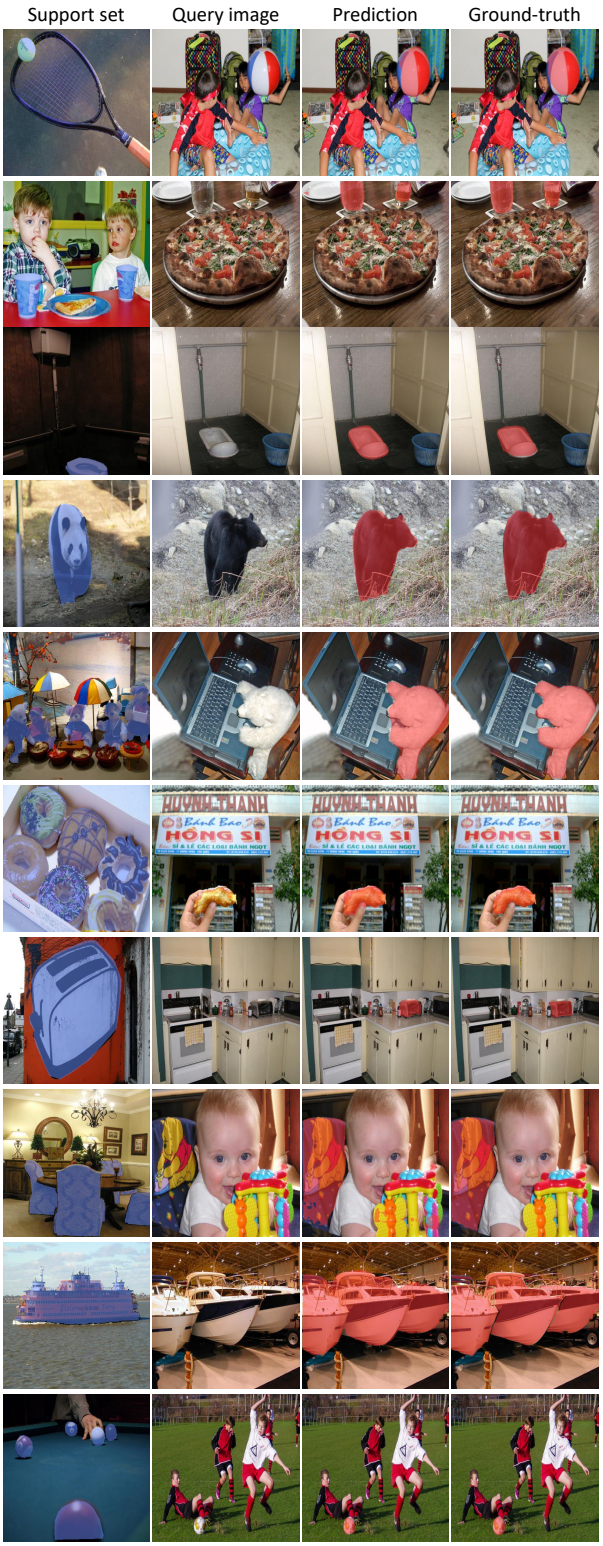

Figure S8: Qualitative (1-shot) results on PASCAL-5^i [14] and COCO-20^i [7] datasets under large intra-class variations.

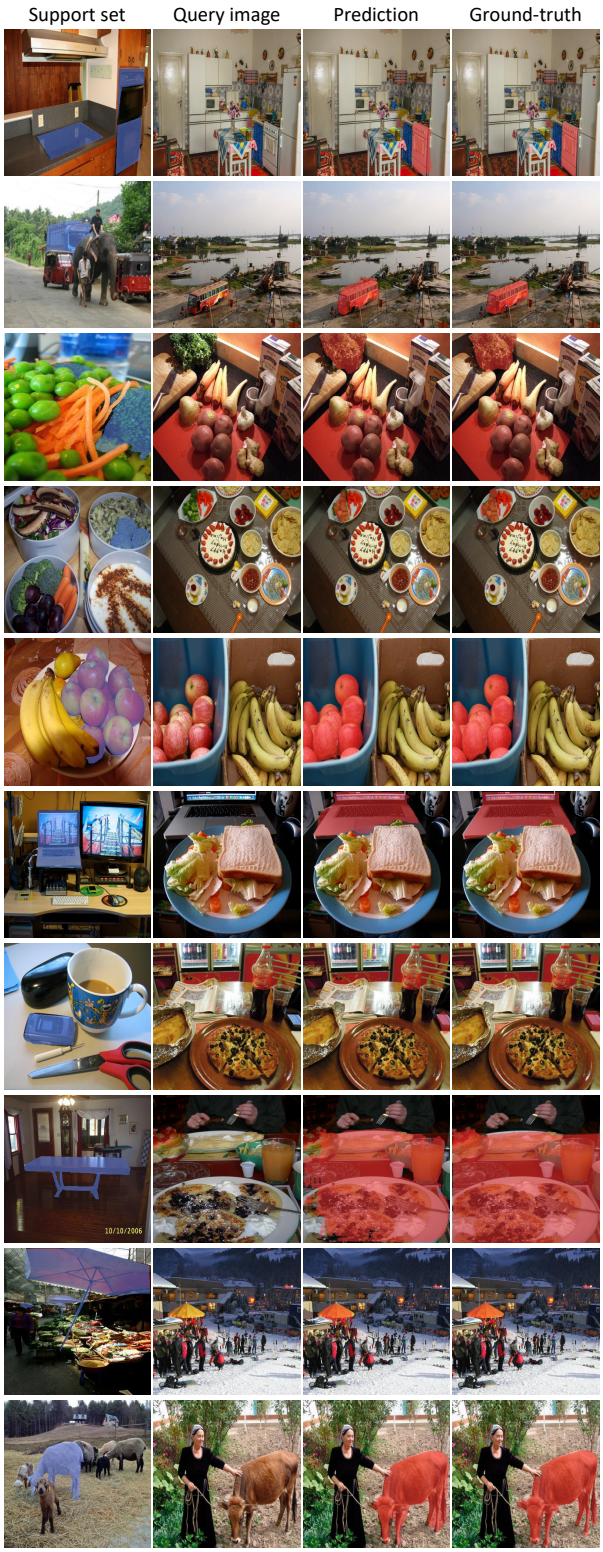

Figure S9: Qualitative (1-shot) results on PASCAL-5^i [14] and COCO-20^i [7] datasets in presence of noisy background clutters.

Figure S10: One-to-many and many-to-many (1-shot) results on PASCAL-5^i [14], COCO-20^i [7], and FSS-1000 [6] datasets.

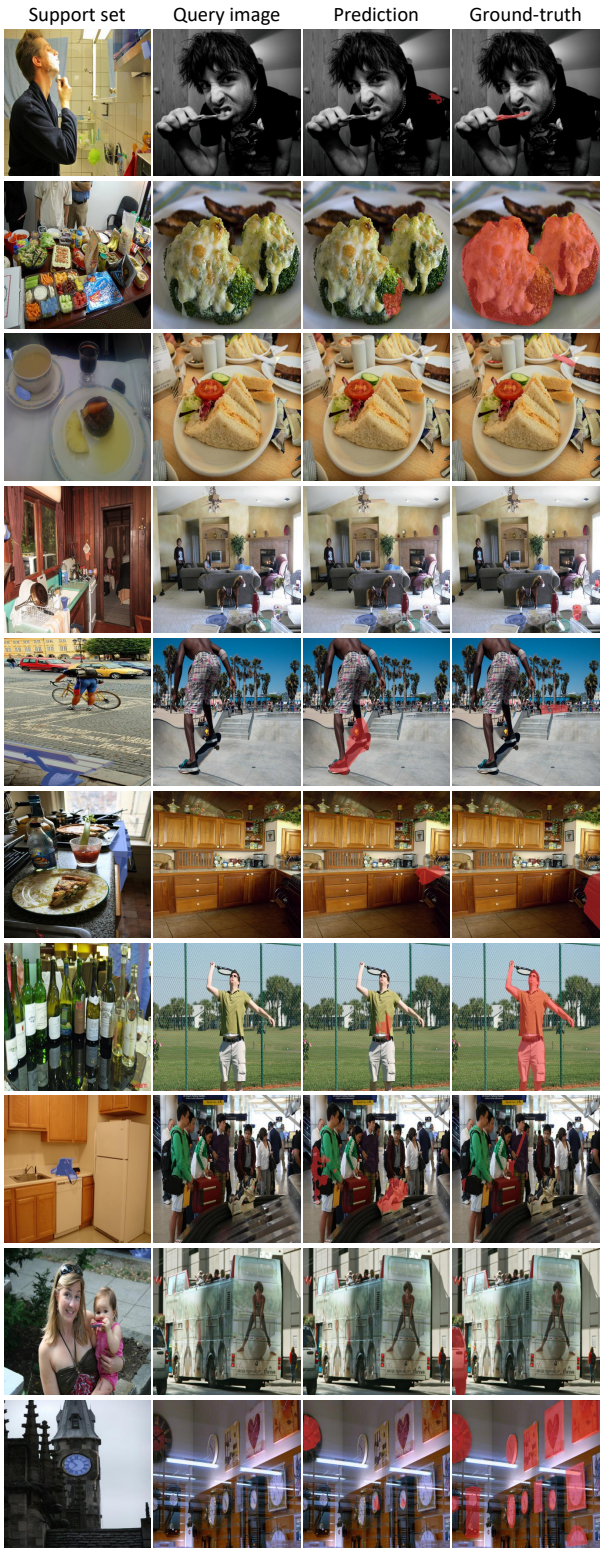

Figure S11: Representative failure cases in presence of severe occlusions, intra-class variances and extremely tiny objects.

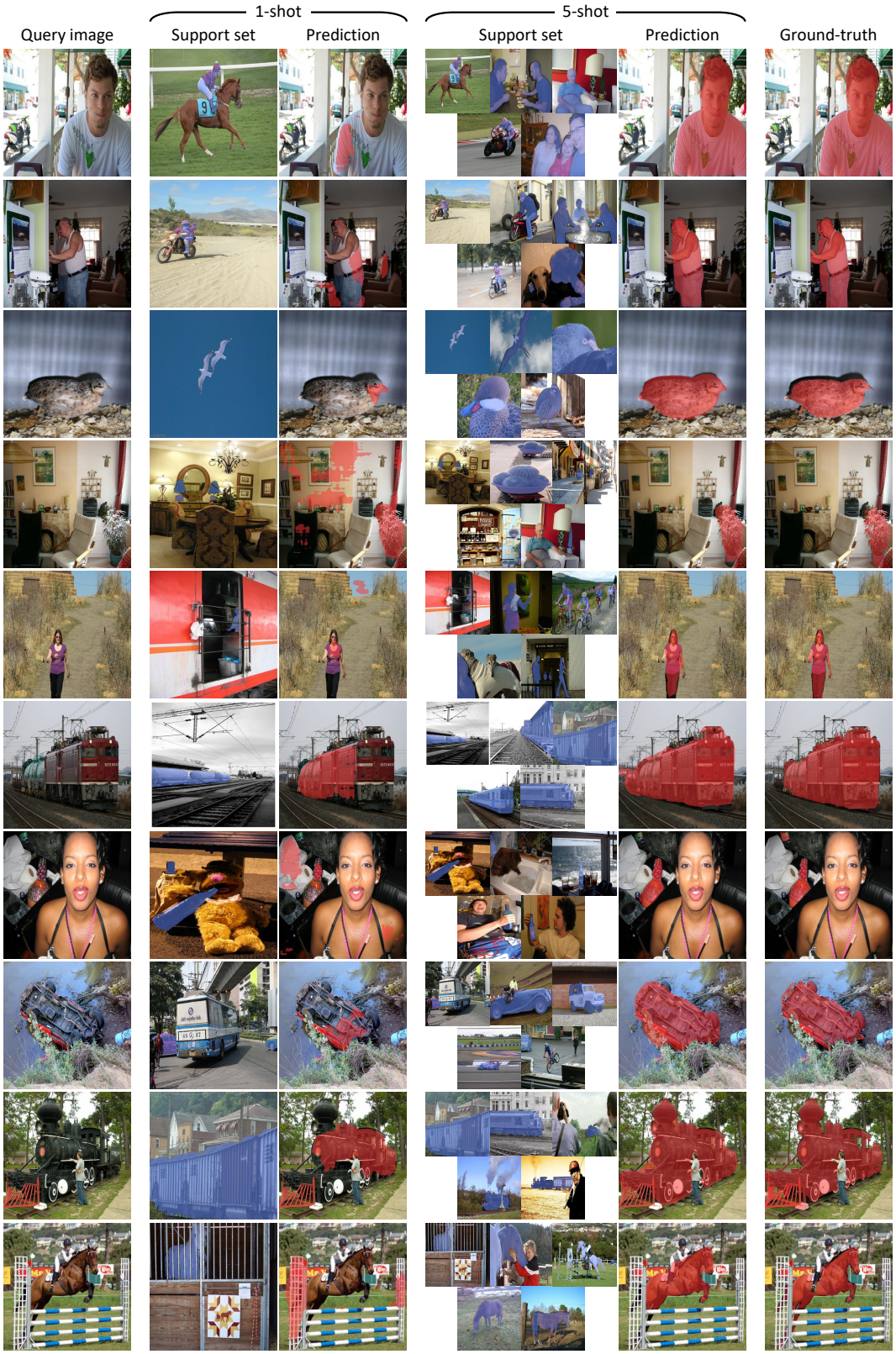

Figure S12: Comparison between 1-shot and 5-shot results on PASCAL-5^i dataset [14]. Multiple support images and mask annotations clearly help our model generate accurate mask predictions on query images in many challenging cases.

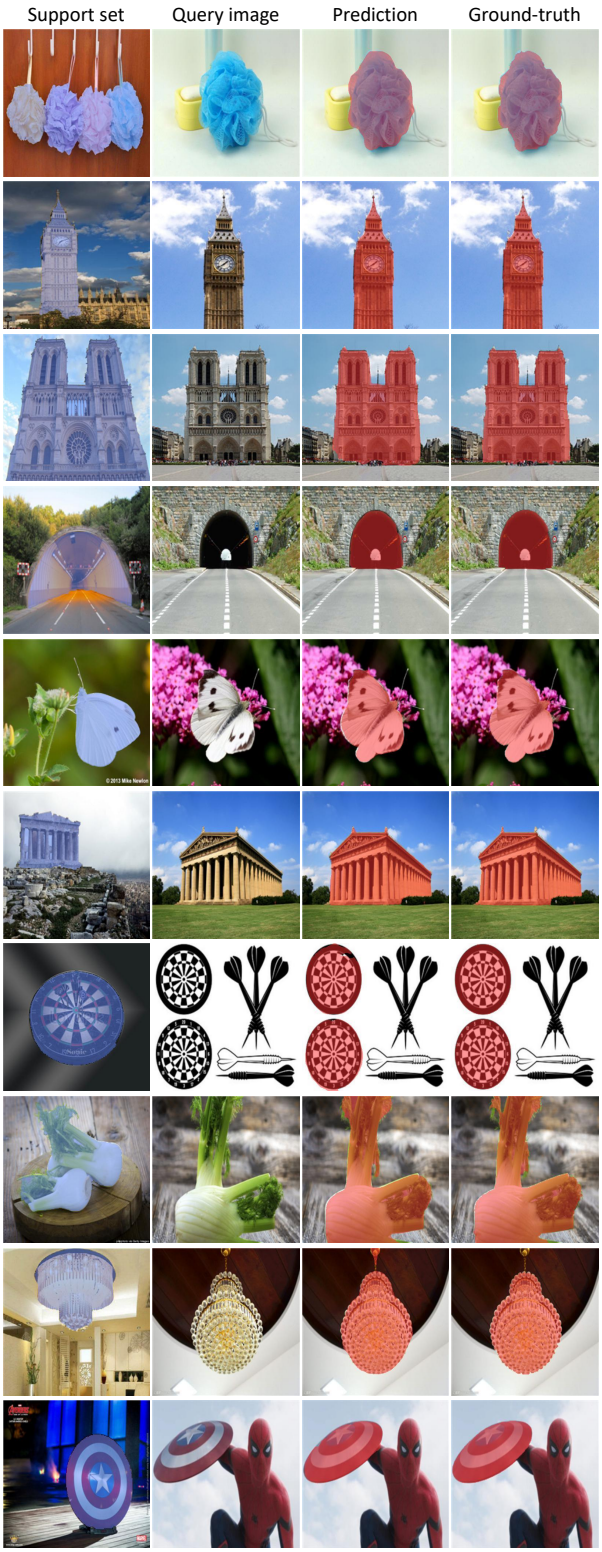

Figure S13: Example (1-shot) results on FSS-1000 [6] dataset consisting of diverse artificial/manmade and natural/organic objects.

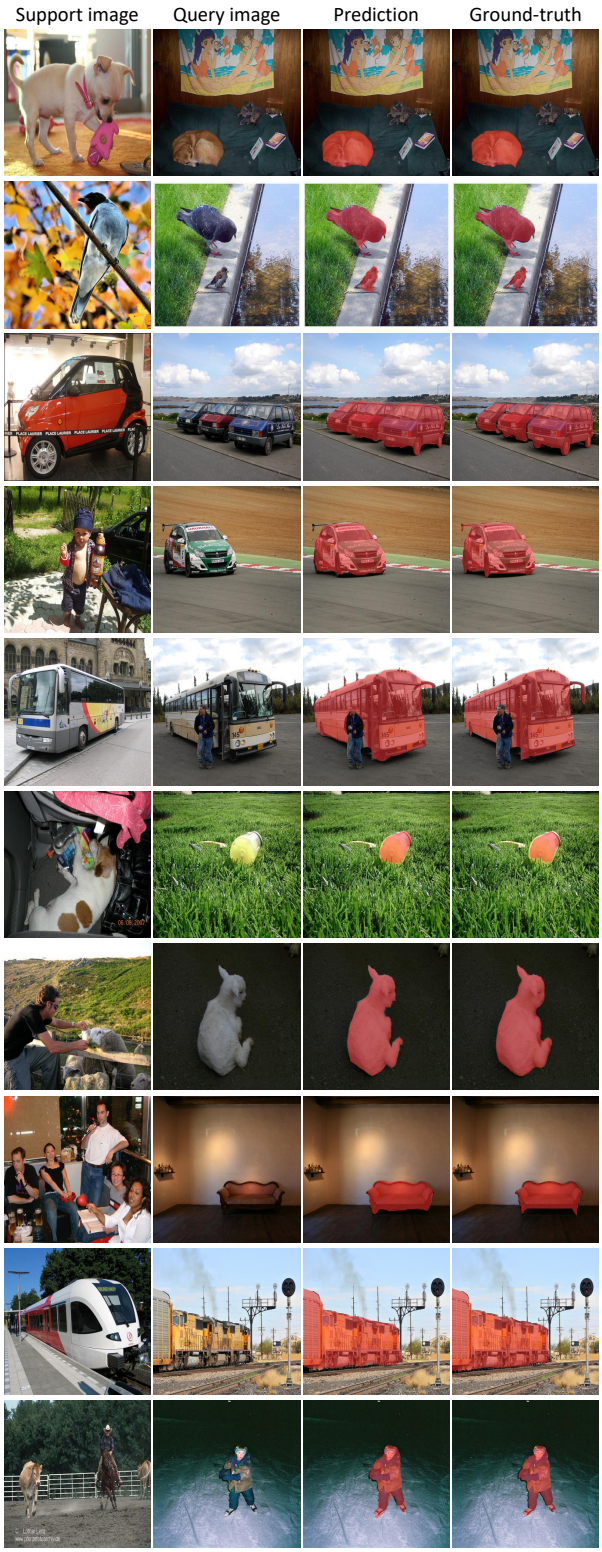

Figure S14: Example results without support feature masking (Eqn. 1 of our main paper) on PASCAL-5^i dataset.

References

[1] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 1

[3] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, Jan 2015. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 2, 5, 6

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[6] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3, 10, 12

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *arXiv preprint arXiv:1504.00325*, 2015. 1, 7, 8, 9, 10

[8] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[9] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[10] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 3

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2019. 1

[12] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. In *International Conference on Learning Representations Workshops (ICLRW)*, 2018. 2

[13] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[14] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *Proc. British Machine Vision Conference (BMVC)*, 2017. 1, 2, 3, 7, 9, 10, 11

[15] Mennatullah Siam, Boris N. Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 2, 4

[17] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1, 2

[18] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[19] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[20] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Ye Qixiang. Prototype mixture models for few-shot semantic segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[21] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

[22] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[23] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[24] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 2020. 2