Integrative Few-Shot Learning for Classification and Segmentation

Dahyun Kang    Minsu Cho
Pohang University of Science and Technology (POSTECH), South Korea

http://cvlab.postech.ac.kr/research/iFSL

Abstract

We introduce the integrative task of few-shot classification and segmentation (FS-CS) that aims to both classify and segment target objects in a query image when the target classes are given with a few examples. This task combines two conventional few-shot learning problems, few-shot classification and segmentation. FS-CS generalizes them to more realistic episodes with arbitrary image pairs, where each target class may or may not be present in the query. To address the task, we propose the integrative few-shot learning (iFSL) framework for FS-CS, which trains a learner to construct class-wise foreground maps for multi-label classification and pixel-wise segmentation. We also develop an effective iFSL model, attentive squeeze network (ASNet), that leverages deep semantic correlation and global self-attention to produce reliable foreground maps. In experiments, the proposed method shows promising performance on the FS-CS task and also achieves the state of the art on standard few-shot segmentation benchmarks.

1. Introduction

Few-shot learning [15,16,30,82,84] is the learning problem where a learner experiences only a limited number of examples as supervision. In computer vision, it has been most actively studied for the tasks of image classification [22,29,66] and semantic segmentation [6,41,48,59] among many others [21,49,54,94,99]. Few-shot classification (FS-C) aims to classify a query image into target classes when a few support examples are given for each target class. Few-shot segmentation (FS-S) is to segment out the target class regions on the query image in a similar setup. While being closely related to each other [33,92,100], these two few-shot learning problems have so far been treated individually. Furthermore, the conventional setups for the few-shot problems, FS-C and FS-S, are limited and do not reflect realistic scenarios; FS-C [28,55,76] presumes that the query always contains one of the target classes in classification, while FS-S [25,52,63] allows the presence of multiple classes but does not handle the absence of the target classes in segmentation. These respective limitations prevent few-shot learning from generalizing to and evaluating on more realistic cases in the wild. For example, when a query image without any target class is given as in Fig. 1, FS-S learners typically segment out arbitrary salient objects in the query.

To address the aforementioned issues, we introduce the integrative task of few-shot classification and segmentation (FS-CS) that combines the two few-shot learning problems into a multi-label and background-aware prediction problem. Given a query image and a few-shot support set for target classes, FS-CS aims to identify the presence of each target class and predict its foreground mask from the query. Unlike FS-C and FS-S, it does not presume either the class exclusiveness in classification or the presence of all the tar-
As a learning framework for FS-CS, we propose **integrative few-shot learning** (iFSL) that learns to construct shared foreground maps for both classification and segmentation. It naturally combines multi-label classification and pixel-wise segmentation by sharing class-wise foreground maps and also allows to learn with class tags or segmentation annotations. For effective iFSL, we design the **attentive squeeze network** (ASNet) that computes semantic attentive squeeze network at-pixel-wise segmentation by sharing class-wise foreground map by strided self-attention. It generates reliable foreground maps for iFSL by leveraging multi-layer neural features [44, 45] and global self-attention [11, 75]. In experiments, we demonstrate the efficacy of the iFSL framework on FS-CS and compare ASNet with recent methods [44, 86–88]. Our method significantly improves over the other methods on FS-CS in terms of classification and segmentation accuracy and also outperforms the recent FS-S methods on the conventional FS-S. We also cross-validate the task transferability between the FS-C, FS-S, and FS-CS learners, and show the FS-CS learners effectively generalize when transferred to the FS-C and FS-S tasks.

Our contribution is summarized as follows:

- We introduce the task of **integrative few-shot classification and segmentation** (FS-CS), which combines few-shot classification and few-shot segmentation into an integrative task by addressing their limitations.

- We propose the **integrative few-shot learning framework** (iFSL), which learns to both classify and segment a query image using class-wise foreground maps.

- We design the **attentive squeeze network** (ASNet), which squeezes semantic correlations into a foreground map for iFSL via strided global self-attention.

- We show in extensive experiments that the framework, iFSL, and the architecture, ASNet, are both effective, achieving a significant gain on FS-S as well as FS-CS.

### 2. Related work

**Few-shot classification (FS-C).** Recent FC-S methods typically learn neural networks that maximize positive class similarity and suppress the rest to predict the most probable class. Such a similarity function is obtained by a) meta-learning embedding functions [1, 23, 26, 28, 51, 67, 76, 93, 96], b) meta-learning to optimize classifier weights [17, 61, 69], or c) transfer learning [7, 9, 20, 37, 50, 58, 71, 81], all of which aim to generalize to unseen classes. This conventional formulation is applicable if a query image corresponds to no less or more than a single class among target classes. To generalize FC-C to classify images associated with either none or multiple classes, we employ the multi-label classification [4, 8, 12, 31, 42]. While the conventional FC-C methods make use of the class uniqueness property via using the categorical cross-entropy, we instead devise a learning framework that compares the binary relationship between the query and each support image individually and estimates a binary presence of the corresponding class.

**Few-shot semantic segmentation (FS-S).** A prevalent FS-S approach is learning to match a query feature map with a set of support feature embeddings that are obtained by collapsing spatial dimensions at the cost of spatial structures [10, 18, 38, 40, 47, 65, 78, 89, 90, 95, 98]. Recent methods [73, 86–88, 97] focus on learning structural details by leveraging dense feature correlation tensors between the query and each support. HSNet [44] learns to squeeze a dense feature correlation tensor and transform it to a segmentation mask via high-dimensional convolutions that analyze the local correlation patterns on the correlation pyramid. We inherit the idea of learning to squeeze correlations and improve it by analyzing the spatial context of the correlation with effective global self-attention [75]. Note that several methods [68, 77, 91] adopt non-local self-attention [80] of the query-key-value interaction for FS-S, but they are distinct from ours in the sense that they learn to transform image feature maps, whereas our method focuses on transforming dense correlation maps via self-attention.

FS-S has been predominantly investigated as an one-way segmentation task, i.e., foreground or background segmentation, since the task is defined so that every target (support) class object appears in query images, thus being not straightforward to extend to a multi-class problem in the wild. Consequently, most work on FS-S except for a few [10, 40, 70, 78] focuses on the one-way segmentation, where the work of [10, 70] among the few presents two-way segmentation results from person-and-object images only, e.g., images containing (person, dog) or (person, table).

**Comparison with other few-shot approaches.** Here we contrast FS-CS with other loosely-related work for generalized few-shot learning. Few-shot open-set classification [39] brings the idea of the open-set problem [14, 62] to few-shot classification by allowing a query to have no target classes. This formulation enables background-aware classification as in FS-CS, whereas multi-label classification is not considered. The work of [19, 72] generalizes few-shot segmentation to a multi-class task, but it is mainly studied under the umbrella of incremental learning [5, 43, 56]. The work of [64] investigates weakly-supervised few-shot segmentation using image-level vision and language supervision, while FS-CS uses visual supervision only. The aforementioned tasks generalize few-shot learning but differ from FS-CS in the sense that FS-CS integrates two related problems under more general and relaxed constraints.
3. Problem formulation

Given a query image and a few support images for target classes, we aim to identify the presence of each class and predict its foreground mask from the query (Fig. 1), which we call the integrative few-shot classification and segmentation (FS-CS). Specifically, let us assume a target (support) class set $C_s$ of $N$ classes and its support set $S = \{(x_i^{(i)}, y_i^{(i)}) | y_i^{(i)} \in C_s\}_{i=1}^K$, which contains $K$ labeled instances for each of the $N$ classes, i.e., N-way K-shot [55,76]. The label $y_i^{(i)}$ is either a class tag (weak label) or a segmentation annotation (strong label). For a given query image $x$, we aim to identify the multi-class occurrence $y_C$ and also predict the segmentation mask $Y_S$ corresponding to the classes. We assume the class set of the query $C$ is a subset of the target class set, i.e., $C \subseteq C_s$, thus it is also possible to obtain $y_C = \emptyset$ and $Y_S = \emptyset$. This naturally generalizes the existing few-shot classification [67,76] and few-shot segmentation [52,63].

Multi-label background-aware prediction. The conventional formulation of few-shot classification (FS-C) [17,67,76] assigns the query to one class among the target classes exclusively and ignores the possibility of the query belonging to none or multiple target classes. FS-CS tackle this limitation and generalizes FS-C to multi-label classification with a background class. A multi-label few-shot classification learner $f_C$ compares semantic similarities between the query and the support images and estimates class-wise occurrences: $\hat{y}_C = f_C(x,S;\theta)$ where $\hat{y}_C$ is an $N$-dimensional multi-hot vector each entry of which indicates the occurrence of the corresponding target class. Note that the query is classified into a background class if none of the target classes were detected. Thanks to the relaxed constraint on the query, i.e., the query not always belonging to exactly one class, FS-CS is more general than FS-C.

Integration of classification and segmentation. FS-CS integrates multi-label few-shot classification with semantic segmentation by adopting pixel-level spatial reasoning. While the conventional FS-S [47,52,63,65,78] assumes the query class set exactly matches the support class set, i.e., $C = C_s$, FS-CS relaxes the assumption such that the query class set can be a subset of the support class set, i.e., $C \subseteq C_s$. In this generalized segmentation setup along with classification, an integrative FS-CS learner $f$ estimates both class-wise occurrences and their semantic segmentation maps: $\{\hat{y}_C, Y_S\} = f(x,S;\theta)$. This combined and generalized formulation gives a high degree of freedom to both of the few-shot learning tasks, which has been missing in the literature; the integrative few-shot learner can predict multi-label background-aware class occurrences and segmentation maps simultaneously under a relaxed constraint on the few-shot episodes.

4. Integrative Few-Shot Learning (iFSL)

To solve the FS-CS problem, we propose an effective learning framework, integrative few-shot learning (iFSL). The iFSL framework is designed to jointly solve few-shot classification and few-shot segmentation using either a class tag or a segmentation supervision. The integrative few-shot learner $f$ takes as input the query image $x$ and the support set $S$ and then produces as output the class-wise foreground maps. The set of class-wise foreground maps $Y$ is comprised of $Y^{(n)} \in \mathbb{R}^{H \times W}$ for $N$ classes:

$$Y = f(x,S;\theta) = \{Y^{(n)}\}_{n=1}^N, \quad (1)$$

where $H \times W$ denotes the size of each map and $\theta$ is parameters to be meta-learned. The output at each position on the map represents the probability of the position being on a foreground region of the corresponding class.

Inference. iFSL infers both class-wise occurrences and segmentation masks on top of the set of foreground maps $Y$. For class-wise occurrences, a multi-hot vector $\hat{y}_C \in \mathbb{R}^N$ is predicted via max pooling followed by thresholding:

$$\hat{y}_C^{(n)} = \begin{cases} 1 & \text{if } \max_{p \in [H] \times [W]} Y^{(n)}(p) \geq \delta, \\ 0 & \text{otherwise}, \end{cases} \quad (2)$$

where $p$ denotes a 2D position, $\delta$ is a threshold, and $[k]$ denotes a set of integers from $1$ to $k$, i.e., $[k] = \{1,2,\cdots,k\}$. We find that inference with average pooling is prone to miss small objects in multi-label classification and thus choose to use max pooling. The detected class at any position on the spatial map signifies the presence of the class.

For segmentation, a segmentation probability tensor $Y_S \in \mathbb{R}^{H \times W \times (N+1)}$ is derived from the class-wise foreground maps. As the background class is not given as a separate support, we estimate the background map in the context of the given supports; we combine $N$ class-wise background maps into an episodic background map on the fly. Specifically, we compute the episodic background map $Y_{bg}$ by averaging the probability maps of not being foreground and then concatenate it with the class-wise foreground maps to obtain a segmentation probability tensor $Y_S$:

$$Y_{bg} = \frac{1}{N} \sum_{n=1}^N (1 - Y^{(n)}), \quad (3)$$

$$Y_S = [Y||Y_{bg}] \in \mathbb{R}^{H \times W \times (N+1)}, \quad (4)$$

The final segmentation mask $\hat{Y}_S \in \mathbb{R}^{H \times W}$ is obtained by computing the most probable class label for each position:

$$\hat{Y}_S = \arg\max_{n \in [N+1]} Y_S. \quad (5)$$

Learning objective. The iFSL framework allows a learner to be trained using a class tag or a segmentation annotation
using the classification loss or segmentation loss, respectively. The classification loss is formulated as the average binary cross-entropy between the spatially average-pooled class scores and its ground-truth segmentation annotation:

\[ \mathcal{L}_C = -\frac{1}{N} \sum_{n=1}^{N} y_{gt}^{(n)} \log \frac{1}{HW} \sum_{p \in [H][W]} Y^{(n)}(p), \]  

(6)

where \( y_{gt} \) denotes the multi-hot encoded ground-truth class.

The segmentation loss is formulated as the average cross-entropy between the class distribution at each individual position and its ground-truth segmentation annotation:

\[ \mathcal{L}_S = -\frac{1}{(N+1)} \frac{1}{HW} \sum_{n=1}^{N+1} \sum_{p \in [H][W]} Y_{gt}^{(n)}(p) \log Y_{s}^{(n)}(p), \]  

(7)

where \( Y_{gt} \) denotes the ground-truth segmentation mask.

These two losses share a similar goal of classification but differ in whether to classify each image or each pixel. Either of them is thus chosen according to the given level of supervision for training.

5. Model architecture

In this section, we present Attentive Squeeze Network (ASNet) of an effective iFSL model. The main building block of ASNet is the attentive squeeze layer (AS layer), which is a high-order self-attention layer that takes a correlation tensor and returns another level of correlational representation. ASNet takes as input the pyramidal cross-correlation tensors between a query and a support image feature pyramids, i.e., a hypercorrelation [44]. The pyramidal correlations are fed to pyramidal AS layers that gradually squeeze the spatial dimensions of the support image, and the pyramidal outputs are merged to a final foreground map in a bottom-up pathway [34,35,44]. Figure 2 illustrates the overall process of ASNet. The \( N \)-way output maps are computed in parallel and collected to prepare the class-wise foreground maps in Eq. (1) for iFSL.

5.1. Attentive Squeeze Network (ASNet)

Hypercorrelation construction. Our method first constructs \( NK \) hypercorrelations [44] between a query and each \( NK \) support image and then learns to generate a foreground segmentation mask w.r.t. each support input. To prepare the input hypercorrelations, an episode, i.e., a query and a support set, is enumerated into a paired list of the query, a support image, and a support label: \( \{(x, (x_{q(i)}, y_{s(i)}))\}_{i=1}^{NK} \). The input image is fed to stacked convolutional layers in a CNN and its mid- to high-level output feature maps are collected to build a feature pyramid \( \{F^{(l)}\}_{l=1}^{L} \), where \( l \) denotes the index of a unit layer, e.g., Bottleneck layer in ResNet50 [22]. We then compute cosine similarity between each pair of feature maps from the pair of query and support feature pyramids to obtain 4D correlation tensors of size \( H_q^{(l)} \times W_q^{(l)} \times H_s^{(l)} \times W_s^{(l)} \), which is followed by ReLU [46]:

\[ C^{(l)}(p_q, p_s) = \text{ReLU} \left( \frac{F^{(l)}(p_q) \cdot F^{(l)}(p_s)}{||F^{(l)}(p_q)|| \cdot ||F^{(l)}(p_s)||} \right). \]  

(8)

These \( L \) correlation tensors are grouped by \( P \) groups of the identical spatial sizes, and then the tensors in
each group are concatenated along a new channel dimension to build a hypercorrelation pyramid: \( \mathbf{C}(p) \mid \mathbf{C}(p) \in \mathbb{R}^{H_i(p) \times W_i(p) \times H_i(p) \times W_i(p) \times c_i(p)} \mid P \) such that the channel size \( C_i(p) \) corresponds to the number of concatenated tensors in the \( p \)-th group. We denote the first two spatial dimensions of the correlation tensor, i.e., \( \mathbb{R}^{H_s \times W_s} \), as query dimensions, and the last two spatial dimensions, i.e., \( \mathbb{R}^{H_s \times W_s} \), as support dimensions hereafter.

**Attentive squeeze layer (AS layer).** The AS layer transforms a correlation tensor to another with a smaller support dimension via strided self-attention. The tensor is recast as a block matrix of size \( H_s \times W_s \) in a hypercorrelation pyramid, we start by reshaping the correlation tensor as a block matrix of size \( H_s \times W_s \) with each element corresponding to a correlation tensor of \( \mathbf{C}(x_q) \in \mathbb{R}^{H_s \times W_s \times C_{in}} \) on the query position \( x_q \) such that

\[
\mathbf{C}^{\text{block}} = \begin{bmatrix}
\mathbf{C}((1, 1)) & \cdots & \mathbf{C}((1, W_q)) \\
\vdots & \ddots & \vdots \\
\mathbf{C}((H_q, 1)) & \cdots & \mathbf{C}((H_q, W_q))
\end{bmatrix}. \tag{9}
\]

We call each element a support correlation tensor. The goal of an AS layer is to analyze the global context of each support correlation tensor and extract a correlational representation with a reduced support dimension while the query dimension is preserved: \( \mathbb{R}^{H_s \times W_s \times H_s \times W_s \times C_{in}} \rightarrow \mathbb{R}^{H_s' \times W_s' \times H_s' \times W_s' \times C_{out}} \), where \( H_s' \leq H_s \) and \( W_s' \leq W_s \).

To learn a holistic pattern of each support correlation, we adopt the global self-attention mechanism \([75]\) for correlational feature transform. The self-attention weights are shared across all query positions and processed in parallel.

Let us denote a support correlation tensor on any query position \( x_q \) by \( \mathbf{C}^o = \mathbf{C}^{\text{block}}(x_q) \) for notational brevity as all positions share the following computation. The self-attention computation starts by embedding a support correlation tensor \( \mathbf{C}^o \) to a target \(^1\) key, value triplet: \( \mathbf{T}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{H_i' \times W_i' \times C_{out}} \), using three convolutions of which strides greater than or equal to one to govern the output size. The resultant target and key correlational representations, \( \mathbf{T} \) and \( \mathbf{K} \), are then used to compute an attention context. The attention context is computed as following matrix multiplication:

\[
\mathbf{A} = \mathbf{TK}^T \in \mathbb{R}^{H_i' \times W_i' \times H_i' \times W_i'}. \tag{10}
\]

Next, the attention context is normalized by softmax such that the votes on key foreground positions sum to one with masking attention by the support mask annotation \( \mathbf{Y}_s \) if available to attend more on the foreground region:

\[
\hat{\mathbf{A}}(p_i, p_k) = \frac{\exp(\mathbf{A}(p_i, p_k)Y_s(p_k))}{\sum_{p_k} \exp(\mathbf{A}(p_i, p_k)Y_s(p_k))},
\]

where \( Y_s(p_k) = \begin{cases} 1 & \text{if } p_k \in [H_s'] \times [W_s'] \text{ is foreground,} \\ -\infty & \text{otherwise}. \end{cases} \tag{11} \]

The masked attention context \( \hat{\mathbf{A}} \) is then used to aggregate the value embedding \( \mathbf{V} \):

\[
\mathbf{C}^a = \hat{\mathbf{A}} \mathbf{V} \in \mathbb{R}^{H_i' \times W_i' \times C_{out}}. \tag{12}
\]

The attended representation is fed to an MLP layer, \( \mathbf{W}_o \), and added to the input. In case the input and output dimensions mismatch, the input is optionally fed to a convolutional layer, \( \mathbf{W}_l \). The addition is followed by an activation layer \( \varphi(\cdot) \) consisting of a group normalization \([85]\) and a ReLU activation \([46]\):

\[
\mathbf{C}_o^a = \varphi(\mathbf{W}_o\mathbf{C}_A^a + \mathbf{W}_l(\mathbf{C}^a)) \in \mathbb{R}^{H_i' \times W_i' \times C_{out}}. \tag{13}
\]

The output is then fed to another MLP that concludes a unit operation of an AS layer:

\[
\mathbf{C}^{a'} = \varphi(\mathbf{W}_E(\mathbf{C}_o^a) + \mathbf{C}_o^a) \in \mathbb{R}^{H_i' \times W_i' \times C_{out}}. \tag{14}
\]

which is embedded to the corresponding query position in the block matrix of Eq. (9). Note that the AS layer can be stacked to progressively reduce the size of support correlation tensor, \( H_s' \times W_s' \), to a smaller size. The overall pipeline of AS layer is illustrated in the supplementary material.

**Multi-layer fusion.** The pyramid correlational representations are merged from the coarsest to the finest level by cascading a pair-wise operation of the following three steps: upsampling, addition, and non-linear transform. We first bi-linearly upsample the bottommost correlational representation to the query spatial dimension of its adjacent earlier one and then add the two representations to obtain a mixed one \( \mathbf{C}^{\text{mix}} \). The mixed representation is fed to two sequential AS layers until it becomes a point feature of size \( H_s' = W_s' = 1 \), which is fed to the subsequent pyramidal fusion. The output from the earliest fusion layer is fed to a convolutional decoder, which consists of interleaved 2D convolution and bi-linear upsampling that map the \( C \)-dimensional channel to \( 2 \) (foreground and background) and the output spatial size to the input query image size. See Fig. 2 for the overall process of multi-layer fusion.

**Class-wise foreground map computation.** The \( K \)-shot output foreground activation maps are averaged to produce a mask prediction for each class. The averaged output map is normalized by softmax over the two channels of the binary segmentation map to obtain a foreground probability prediction \( \mathbf{Y}(n) \in \mathbb{R}^{H \times W} \).
Table 1. Performance comparison of ASNet and others on FS-CS and Pascal-5\(^i\) [63]. All methods are trained and evaluated under the iFSL framework given strong labels, i.e., class segmentation masks, except for ASNet\(_w\) that is trained only with weak labels, i.e., class tags.

<table>
<thead>
<tr>
<th>method</th>
<th>1-way 1-shot</th>
<th>2-way 1-shot</th>
<th>classification 0/1 exact ratio (%)</th>
<th>segmentation mIoU (%)</th>
<th>classification 0/1 exact ratio (%)</th>
<th>segmentation mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0(^{th})</td>
<td>1(^{st})</td>
<td>2(^{nd})</td>
<td>avg.</td>
<td>0(^{th})</td>
<td>1(^{st})</td>
</tr>
<tr>
<td>PANet [78]</td>
<td>69.9</td>
<td>67.7</td>
<td>68.8</td>
<td>69.4</td>
<td>69.0</td>
<td>32.8</td>
</tr>
<tr>
<td>PFENet [73]</td>
<td>69.8</td>
<td>82.4</td>
<td>68.1</td>
<td>77.9</td>
<td>74.6</td>
<td>38.3</td>
</tr>
<tr>
<td>HSNet [44]</td>
<td>86.6</td>
<td>84.8</td>
<td>76.9</td>
<td>86.3</td>
<td>83.7</td>
<td>49.1</td>
</tr>
<tr>
<td>ASNet(_w)</td>
<td>86.4</td>
<td>86.3</td>
<td>79.9</td>
<td>84.5</td>
<td>82.0</td>
<td>10.8</td>
</tr>
<tr>
<td>ASNet</td>
<td>84.9</td>
<td>89.6</td>
<td>79.0</td>
<td>86.2</td>
<td>84.9</td>
<td>51.7</td>
</tr>
</tbody>
</table>

Table 2. Performance comparison of ASNet and others on FS-CS and COCO-20\(^i\) [47].

<table>
<thead>
<tr>
<th>method</th>
<th>1-way 1-shot</th>
<th>2-way 1-shot</th>
<th>classification 0/1 exact ratio (%)</th>
<th>segmentation mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ER</td>
<td>mIoU</td>
<td>ER</td>
<td>mIoU</td>
</tr>
<tr>
<td>PANet [78]</td>
<td>66.7</td>
<td>25.2</td>
<td>48.5</td>
<td>23.6</td>
</tr>
<tr>
<td>PFENet [73]</td>
<td>71.4</td>
<td>31.9</td>
<td>36.5</td>
<td>22.6</td>
</tr>
<tr>
<td>HSNet [44]</td>
<td>77.0</td>
<td>34.3</td>
<td>62.5</td>
<td>29.5</td>
</tr>
<tr>
<td>ASNet</td>
<td>78.6</td>
<td>35.8</td>
<td>63.1</td>
<td>31.6</td>
</tr>
</tbody>
</table>

Figure 3. 2-way 1-shot segmentation results of ASNet on FS-CS. The examples cover all three cases of \(C = \emptyset, C \subset C_i,\) and \(C = C_i.\) The images are resized to square shape for visualization.

### 6. Experiments

In this section we report our experimental results regarding the FS-CS task, the iFSL framework, as well as the ASNet after briefly describing implementation details and evaluation benchmarks. See the supplementary material for additional results, analyses, and experimental details.

### 6.1. Experimental setups

**Experimental settings.** We select ResNet50 and ResNet-101 [22] pretrained on ImageNet [60] as our backbone networks for a fair comparison with other methods and freeze the backbone during training as similarly as the previous work [44, 73]. We train models using Adam [27] optimizer with learning rate of \(10^{-4}\) and \(10^{-5}\) for the classification loss and the segmentation loss, respectively. We train all models with 1-way 1-shot training episodes and evaluate the models on arbitrary \(N\)-way \(K\)-shot episodes. For inferring class occurrences, we use a threshold \(\delta = 0.5.\) All the AS layers are implemented as multi-head attention with 8 heads. The number of correlation pyramid is set to \(P = 3.\)

**Dataset.** For the new task of FS-CS, we construct a benchmark adopting the images and splits from the two widely-used FS-S datasets, Pascal-5\(^i\) [13, 63] and COCO-20\(^i\) [36, 47], which are also suitable for multi-label classification [83]. Within each fold, we construct an episode by randomly sampling a query and an \(N\)-way \(K\)-shot support set that annotates the query with \(N\)-way class labels and an \((N+1)\)-way segmentation mask in the context of the support set. For the FS-S task, we also use Pascal-5\(^i\) and COCO-20\(^i\) following the same data splits as [63] and [47], respectively.

**Evaluation.** Each dataset is split into four mutually disjoint class sets and cross-validated. For multi-label classification evaluation metrics, we use the 0/1 exact ratio \(ER = \frac{1}{N} \sum_{n} I[y_{gt}^{(n)} = y_{C}^{(n)}].\) In the supplementary material, we also report the results in accuracy \(acc = \frac{1}{N} \sum_{n} I[y_{gt}^{(n)} = y_{C}^{(n)}].\) For segmentation, we use mean IoU mIoU \(\frac{1}{N} \sum_{c} IoU_{c} \in [63, 78],\) where \(IoU_{c}\) denotes an IoU value of \(c\)th class.

### 6.2. Experimental evaluation of iFSL on FS-CS

In this subsection, we investigate the iFSL learning framework on the FS-CS task. All ablation studies are conducted using ResNet50 on Pascal-5\(^i\) and evaluated in 1-way 1-shot setup unless specified otherwise. Note that it is difficult to present a fair and direct comparison between the conventional FS-C and our few-shot classification task since FS-C is always evaluated on single-label classification benchmarks [2, 32, 57, 74, 76], whereas our task is evaluated on multi-label benchmarks [13, 36], which are irreducible to a single-label one in general.
Effectiveness of iFSL on FS-CS. We validate the iFSL framework on FS-CS and also compare the performance of ASNet with those of three recent state-of-the-art methods, PANet [78], PFENet [73], and HSNet [44], which are originally proposed for the conventional FS-S task; all the models are trained by iFSL for a fair comparison. Note that we exclude the background merging step (Eqs. 3 and 4) for PANet as its own pipeline produces a multi-class output including background. Tables 1 and 2 validate the iFSL framework on the FS-CS task quantitatively, where our ASNet surpasses other methods on both 1-way and 2-way setups in terms of few-shot classification as well as the segmentation performance. The 2-way segmentation results are also qualitatively demonstrated in Fig. 3 visualizing exhaustive inclusion relations between a query class set $\mathcal{C}$ and a target (support) class set $\mathcal{C}_s$ in a 2-way setup.

Weakly-supervised iFSL. The iFSL framework is versatile across the level of supervision: weak labels (class tags) or strong labels (segmentation masks). Assuming weak labels are available but strong labels are not, ASNet is trainable with the classification learning objective of iFSL (Eq. 6) and its results are presented as ASNet$_w$ in Table 1. ASNet$_w$ performs on par with ASNet in terms of classification ER (82.0% vs. 84.9% on 1-way 1-shot), but performs ineffectively on the segmentation task (15.0% vs. 52.3% on 1-way 1-shot). The result implies that the class tag labels are sufficient for a model to recognize the class occurrences, but are weak to endorse model’s precise spatial recognition ability.

Multi-class scalability of FS-CS. In addition, FS-CS is extensible to a multi-class problem with arbitrary numbers of classes, while FS-S is not as flexible as FS-CS in the wild. Figure 4 compares the FS-CS performances of four methods by varying the $N$-way classes from one to five, where the other experimental setup follows the same one as in Table 1. Our ASNet shows consistently better performances than other methods on FS-CS in varying number of classes.

Robustness of FS-CS against task transfer. We evaluate the transferability between FS-CS, FS-C, and FS-S by training a model on one task and evaluating it on the other task. The results are compared in Fig. 5 in which ‘FS-S $\rightarrow$ FS-CS’ represents the result where the model trained on the FS-S task (with the guarantee of support class presence) is evaluated on the FS-CS setup. To construct training and validation splits for FS-C or FS-S, we sample episodes that satisfy the constraint of support class occurrences. For training FS-C models, we use the class tag supervision only. All the other settings are fixed the same, e.g., we use ASNet with ResNet50 and Pascal-$^5$.

The results show that FS-CS learners, i.e., models trained on FS-CS, are transferable to the two conventional few-shot learning tasks and yet overcome their shortcomings. The transferability between few-shot classification tasks, i.e., FS-C and FS-CS$_w$, is presented in Fig. 5 (a). On this setup, the FS-CS$_w$ learner is evaluated by predicting a higher class response between the two classes, although it is trained using the multi-label classification objective. The FS-CS learner closely competes with the FS-C learner on FS-C in terms of classification accuracy. In contrast, the task transfer between segmentation tasks, FS-S and FS-CS, results in asymmetric outcomes as shown in Fig. 5 (b) and (c). The FS-CS learner shows relatively small performance drop on FS-S, however, the FS-S learner suffers a severe performance drop on FS-CS. Qualitative examples in Fig. 1 demonstrate that the FS-S learner predicts a vast number of false-positive pixels and results in poor performances. In contrast, the FS-CS learner successfully distinguishes the region of interest by analyzing the semantic relevance of the query objects between the support set.

6.3. Comparison with recent FS-S methods on FS-S

Tables 3 and 4 compare the results of the recent few-shot semantic segmentation methods and ASNet on the conven-
6.4. Analyses on the model architecture

We perform ablation studies on the model architecture to reveal the benefit of each component. We replace the global self-attention in the ASNet layer with the local self-attention [53] to see the effect of the global self-attention (Table 5a). The local self-attention variant is compatible with the global ASNet in terms of the classification exact ratio but degrades the segmentation mIoU significantly, signifying the importance of the learning the global context of feature correlations. Next, we ablate the attention masking in Eq. (11), which verifies that the attention masking prior is effective (Table 5b). Lastly, we replace the multi-layer fusion path with spatial average pooling over the support dimensions followed by element-wise addition (Table 5c), and the result indicates that it is crucial to fuse outputs from the multi-layer correlations to precisely estimate class occurrence and segmentation masks.

7. Discussion

We have introduced the integrative task of few-shot classification and segmentation (FS-CS) that generalizes two existing few-shot learning problems. Our proposed integrative few-shot learning (iFSL) framework is shown to be effective on FS-CS, in addition, our proposed attentive squeeze network (ASNet) outperforms recent state-of-the-art methods on both FS-CS and FS-S. The iFSL design allows a model to learn either with weak or strong labels, that being said, learning our method with weak labels achieves low segmentation performances. This result opens a future direction of effectively boosting the segmentation performance leveraging weak labels in the absence of strong labels for FS-CS.

Acknowledgements. This work was supported by Samsung Advanced Institute of Technology (SAIT) and also by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by DAPA and ADD (UD190031RD).
References


[15] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2006. 1


[21] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In Proc. IEEE International Conference on Computer Vision (ICCV), 2021. 1


