AMP: Adaptive Masked Proxies for Few-Shot Segmentation

Mennatullah Siam
University of Alberta
mennatul@ualberta.ca

Boris N. Oreshkin
Element AI
boris@elementai.com

Martin Jagersand
University of Alberta
jag@cs.ualberta.ca

Abstract

Deep learning has thrived by training on large-scale datasets. However, in robotics applications sample efficiency is critical. We propose a novel adaptive masked proxies method that constructs the final segmentation layer weights from few labelled samples. It utilizes multi-resolution average pooling on base embeddings masked with the label to act as a positive proxy for the new class, while fusing it with the previously learned class signatures. Our method is evaluated on PASCAL-5^i dataset and outperforms the state-of-the-art in the few-shot semantic segmentation. Unlike previous methods, our approach does not require a second branch to estimate parameters or prototypes, which enables it to be used with 2-stream motion and appearance based segmentation networks. We further propose a novel setup for evaluating continual learning of object segmentation which we name incremental PASCAL (iPASCAL) where our method outperforms the baseline method. Our code is publicly available at `https://github.com/MSiam/AdaptiveMaskedProxies`.

1. Introduction

Children are able to adapt their knowledge and learn about their surrounding environment with limited samples [18]. One of the main bottlenecks in the current deep learning methods is their dependency on the large-scale training data. However, it is intractable to collect one large-scale dataset that contains all the required object classes for different environments. This motivated the emergence of few-shot learning methods [12, 38, 32, 26, 27]. These early works were primarily focused on solving few-shot image classification tasks, where a support set consists of a few images and their class labels. The earliest attempt to solve the few-shot segmentation task seems to be the approach proposed by Shaban et al. [28] that predicts the parameters of the final segmentation layer. This and other previous methods require the training of an additional branch to guide the backbone segmentation network. The additional network introduces extra computational burden. On top of that, existing approaches cannot be trivially extended to handle the continuous stream of data containing annotations for both novel and previously learned classes.

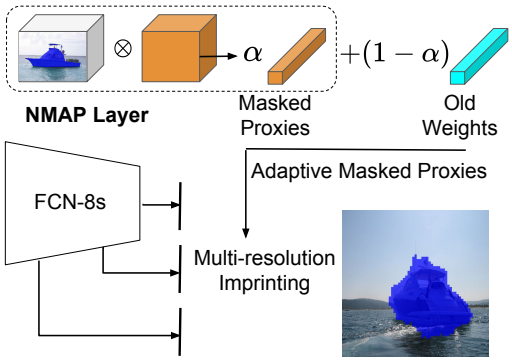

Figure 1: Multi-resolution adaptive imprinting in AMP.

To address these shortcomings, we propose a novel sample efficient adaptive masked proxies method, which we call AMP. It constructs weights of the final segmentation layer via multi-resolution imprinting. AMP does not rely on a second guidance branch, as shown in Figure 1. Following the terminology of [19], a proxy is a representative signature of a given class. In the few-shot segmentation setup, the support set contains pixel-wise class labels for each support image. Therefore, the response of the backbone fully convolutional network (FCN) to a set of images from a given class in the support set can be masked by segmentation labels and then average pooled to create a proxy for this class. This forms what we call a normalized masked average pooling layer (NMAP in Fig. 1). The computed proxies are used to set the 1x1 convolutional filters for the new classes, forming the process known as weight imprinting [23]. Multi-resolution weight imprinting is proposed to improve the segmentation accuracy of our method.

We further consider the continual learning setup in which a few-shot algorithm may be presented with a sequence of support sets (continuous semantic segmentation scenario). In connection with this scenario, we propose to adapt the previously learned class weights with the new proxies from

each incoming support set. Imprinting only the weights for the positive class, i.e. the newly added class, is insufficient as new samples will incorporate new information about other classes as well. For example, learning a new class for *boat* will also entail learning new information about the *background* class, which should include *sea*. To address this, a novel method for updating the weights of the previously learned classes without back-propagation is proposed. The adaptation part of our method is inspired by the classical approaches in learning adaptive correlation filters [1, 7]. Correlation filters date back to 1980s [8]. More recently, the fast object tracking method [1] relied on hand crafted features to form the correlation filters and adapted them using a running average. In our method the adaptation of the previously learned weights is based on a similar approach, yielding the ability to process the continuous stream of data containing novel and existing classes. This opens the door toward leveraging segmentation networks to continually learn semantic segmentation in a sample efficient manner.

To sum up, AMP is shown to provide sample efficiency in three scenarios: (1) few-shot semantic segmentation, (2) video object segmentation and (3) continuous semantic segmentation. Unlike previous methods, AMP can easily operate with any pre-trained network without the need to train a second branch, which entails fewer parameters. In the video object segmentation scenario we show that our method can be used with a 2-stream motion and appearance network without any additional guidance branch. AMP is flexible and still allows coupling with back-propagation using the support image-label pair. The proxy weight imprinting steps can be interleaved with the back-propagation steps to boost the adaptation process. AMP is evaluated on PASCAL-5^i [28], DAVIS benchmark [22], FBMS [20] and our proposed iPASCAL setup. The novel contributions of this paper can be summarized as follows.

- **Normalized masked average pooling layer** that efficiently computes a class signature from the backbone FCN response without relying on an additional branch.
- **Multi-resolution imprinting scheme** that imprints the proxies from several resolutions of the backbone FCN to increase accuracy.
- **Novel adaptation mechanism** that updates the weights of known classes based on the new proxies.
- **Empirical results** that demonstrate that our method is state-of-the-art on PASCAL-5^i, and on DAVIS'16.
- **iPASCAL, a new version of PASCAL-VOC** to evaluate the continuous semantic segmentation.

2. Related Work

2.1. Few-shot Classification

In few-shot classification, the model is provided with a support set and a query image. The support set contains a few labelled samples that can be used to train the model, while the query image is used to test the final model. The setup is formulated as k-shot n-way, where k denotes the number of samples per class, while n denotes the number of classes in the support set. An early approach to solve the few-shot learning problem relied on Bayesian methodology [6]. More recently, Vinyals et al. proposed matching networks approach that learns an end-to-end differentiable nearest neighbour [38]. Following that, Snell et al. proposed prototypical networks based on the assumption that there exists an embedding space in which points belonging to one class cluster around their corresponding centroid [32]. Qiao et al. proposed a parameter predictor method [24]. Finally, a method for computing imprinted weights was proposed by Qi et al. [23].

2.2. Few-shot Semantic Segmentation

Unlike the classification scenario that assumes the availability of image level class labels, the few-shot segmentation relies on pixel-wise class labels for support images. A popular dataset used to evaluate few-shot segmentation is PASCAL-5^i [28]. The dataset is sub-divided into 4 folds each containing 5 classes. A fold contains labelled samples from 5 classes that are used for evaluating the few-shot learning method. The rest 15 classes are used for training. Shaban et al. proposed a 2-branch method [28], where the second branch predicts the parameters for the final segmentation layer. The baselines proposed by Shaban et al. [28] included nearest neighbour, siamese network, and naive fine-tuning. Rakelly et al. proposed a 2-branch method where the second branch acts as a conditioning branch instead [25]. Finally, Dong et al. inspired from prototypical networks, designed another 2-branch method to learn prototypes for the few-shot segmentation problem [4]. Clearly, most of the previously proposed methods require an extra branch trained in a simulated few-shot setting. They cannot be trivially extended to continue adaptation whilst processing a continuous stream of data with multiple classes.

In a concurrent work, Zhang et al. [41] proposed a single branch network deriving guidance features from masked average pooling layer. This is similar to our NMAP layer. Zhang et al. [41] use the output of their pooling layer to compute a guidance to the base network. AMP uses NMAP output to imprint the 1x1 convolutional layer weights. AMP has the following advantages: (i) it allows the adaptation of imprinted weights in continuous data stream, (ii) it can be seamlessly coupled with any pre-trained networks, including 2-stream networks for video object segmentation.

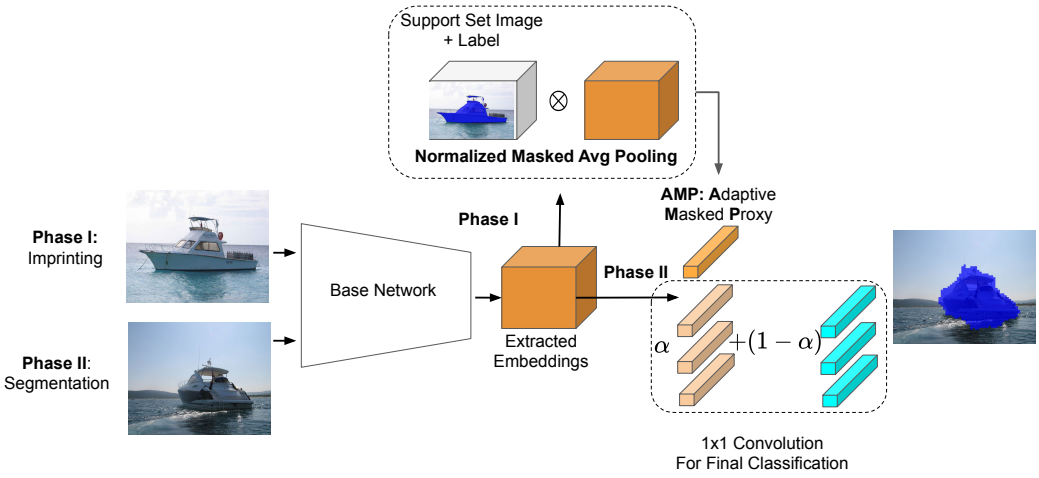

Figure 2: AMP using the NMAP Layer. For simplicity it shows the imprinting on the final layer solely. Nonetheless, our scheme is applied on multiple resolution levels.

3. AMP: Adaptive Masked Proxies

Our approach, which we call AMP, is rooted deeply in the concept of weight imprinting [23]. The imprinting process was initially proposed in the context of classification [23]. The method used the normalized responses of the base feature extractor as weights of the final fully connected layer. In this context, the normalized response of the feature extractor for a given class is called a proxy. The justification behind such learning scheme is based on the relation between metric learning, proxy-NCA loss and softmax cross-entropy loss [19]. 1x1 convolutional layers are equivalent to fully connected layers. Hence we propose to utilize base segmentation network activations as proxies to imprint the 1x1 convolutional filters of the final segmentation layer. When convolved with the query image, the imprinted proxy activates pixels maximally similar to its class signature.

However, it is not trivial to perform weight imprinting in semantic segmentation, unlike in classification. First, in the classification setup the output embedding vector corresponds to a single class and hence can be used directly for imprinting. By contrast, a segmentation network outputs 3D embeddings, which incorporate features for a multitude of different classes, both novel and previously learned. Second, unlike classification, multi-resolution support is essential in segmentation.

We propose the following novel architectural components to address the challenges outlined above. First, in Section 3.1 and in Section 3.2 we propose the proxy masking and adaptation methods to handle multi-class segmentation. Second, in Section 3.3 we propose a multi-resolution weight imprinting scheme to maintain the segmentation accuracy during imprinting. The contribution of each method to the overall accuracy is further motivated experimentally in Section 4.2.

3.1. Normalized Masked Average Pooling

We propose to address the problem of imprinting the 3D segmentation base network embeddings that contain responses from multiple classes in a single image by masking the embeddings prior to averaging and normalization. We encapsulate this function in a NMAP layer (refer to Figures 1 and 2). To construct a proxy for one target class, the NMAP layer bilinearly upsamples segmentation base network outputs and masks them via the pixel-wise labels for the target class available in the support set. This is followed by average pooling and normalization as follows:

$$P_l^r = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{N}\sum_{x\in X} F^{ri}(x)Y_l^i(x), \tag{1a}$$

$$\hat{P}_l^r = \frac{P_l^r}{\|P_l^r\|_2}. \tag{1b}$$

Here Y_l^i is a binary mask for i^{th} image with the novel class l, F^{ri} is the corresponding output feature maps for i^{th} image and rth resolution. X is the set of all possible spatial locations and N is the number of pixels that are labelled as foreground for class l. The normalized output from the masked average pooling layer $\hat{P}_l^r$ can be further used as proxies representing class l and resolution r. In the case of a novel class the proxy can be utilized directly as filter weights. In the case of few-shot learning, the average of all the NMAP processed features for the samples provided in the support set for a given class is used as its proxy.

3.2. Adaptive Proxies

The NMAP layer solves the problem of processing a single support set. However, in practice many of the applications require the ability to process a continuous stream of support sets. This is the case in continuous semantic segmentation and video object segmentation scenarios. In this context the learning algorithm is presented with a sequence of support sets. Each incoming support set may provide information on both the new class and the previously learned classes. It is valuable to utilize both instead of solely imprinting the new class weights. At the same time, in the case of the previously learned classes, e.g. background, it is not wise to simply override what the network learned from the large-scale training either. A good example illustrating the need for updating the negative classes is the addition of class *boat*. It is obvious that the *background* class needs to be updated to match the *sea background*, especially if the images with sea background are not part of the large scale training dataset.

To take advantage of the information available in the continuous stream of data, we propose to adapt class proxies with the information obtained from each new support set. We propose the following exponentially smoothed adaptive scheme with update rate α:

$$\hat{W}_l^r = \alpha \hat{P}_l^r + (1-\alpha) W_l^r. \tag{2}$$

Here $\hat{P}_l^r$ is the normalized masked proxy for class l, W_l^r is the previously learned 1x1 convolutional filter at resolution r, $\hat{W}_l^r$ is the updated W_l^r. The update rate can be either treated as as a hyper-parameter or learned.

The adaptation mechanism is applied differently in the few-shot setup and in the continual learning setup. In the few-shot setup, the support set contains segmentation masks for each new class foreground and background. The adaptation process is performed on the background class weights from the large scale training. The proxies for the novel classes are derived directly from the NMAP layer via imprinting with no adaptation. In the continual learning setup, the proxies for all the classes learned up to the current task are available when a new support set is processed. Thus, we adapt all the proxies learned in all the previous tasks for which samples are available in the support set of the current task.

3.3. Multi-resolution Imprinting Scheme

In the classification scenario, in which imprinting was originally proposed, the resolution aspect is not naturally prominent. In contrast, in the segmentation scenario, resolution is naturally important to obtain very accurate segmentation mask predictions. On top of that, we argue that imprinting the outputs of several resolution levels and fusing the probability maps from those in the final probability map

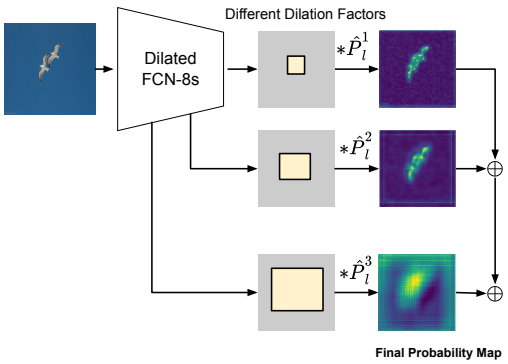

Figure 3: Multi-resolution imprinting using proxies from different resolution levels.

can be used to improve overall segmentation accuracy. This is illustrated in Fig. 3, showing the output heatmaps from 1x1 convolution using our proposed proxies as imprinted weights at three different resolutions, $\hat{P}_l^1$, $\hat{P}_l^2$, $\hat{P}_l^3$. Clearly, the coarse resolution captures blobs necessary for global alignment, while the fine resolution provides the granular details required for an accurate segmentation.

This idea is further supported by the T-SNE [17] plot of the proxies learned in the proposed NMAP layer at different resolutions depicted in Fig. 4. It shows the 5 classes belonging to fold 0 in PASCAL-5^i at 3 resolutions imprinted by our AMP model. A few things catch attention in Fig. 4. First, clustering is different at different resolutions. Fusing probability maps at different resolutions may therefore be advantageous from statistical standpoint, as slight segmentation errors at different resolutions may cancel each other. Second, the class-level clustering is not necessarily tightest at the highest resolution level: mid-resolution layer L2 seems to provide the tightest clustering. This may seem counter-intuitive. Yet, this is perfectly in line with the latest empirical results in weakly-supervised learning (see [2] and related work). For example, [2] clearly demonstrates that convolutional networks store most of the class level information in the middle layers, and mid-resolution features result in the best transfer learning classification results.

3.4. Base Network Architectures

The backbone architecture used in our segmentation network is a VGG-16 [31] that is pre-trained on ImageNet [3]. Similar to the FCN8s architecture [16] skip connections are used to benefit from higher resolution feature maps, and a 1x1 convolution layers are used to map from the feature space to the label space. Unlike FCN8s we utilize bilinear interpolation layers with fixed weights for upsampling. This is to simplify the imprinting of weights based on the support set (transposed convolutions are hard to imprint). We also rely on an extension to the above base network us-

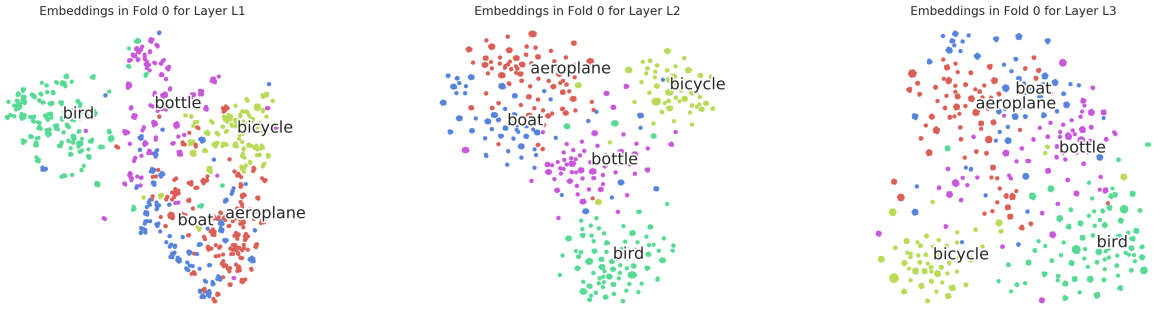

Figure 4: Visualization for the T-SNE [17] embeddings for the generated masked proxies. Layers L1, L2, L3 denote the smaller to higher resolution feature maps.

ing dilated convolution [40], which we call DFCN8s. The last two pooling layers are replaced by dilated convolution with dilation factors 2 and 4 respectively. This increases the receptive field without affecting the resolution. Finally, a more compact version of the network with two final convolutional layers removed is denoted as Reduced-DFCN8s. The final classification layer, and the two 1x1 convolutional layers following dilated convolutions in the case of DFCN8s and the Reduced-DFCN8s are the ones imprinted.

In the video object segmentation scenario we use a 2-stream wide-resnet [39] architecture. Each stream has 11 residual blocks followed by multiplying the output activation from both motion and appearance. The motion is presented to the model as optical flow based on Liu et al. [15] and converted to RGB using a color wheel. The flexibility of our method enables it to work with different architectures without the overhead of designing another branch to provide guidance, predicted parameters or prototypes.

3.5. Training and Evaluation Methodology

Few-shot segmentation. We use the same setup as Shaban et al. [28]. The initial training phase relies on a large scale dataset D_{train} including semantic label maps for classes in L_{train}. During the test phase, a support set and a query image are sampled from D_{test} containing novel classes with labels in L_{test}, where $L_{train} \cap L_{test} = \emptyset$. The support set contains pairs $S = (I_i, Y_i(l))_{i=1}^{k}$, where I_i is the i^{th} image in the set and $Y_i(l)$ is the corresponding binary mask. The binary mask $Y_i(l)$ is constructed with novel class l labelled as foreground while the rest of the pixels are considered background. As before, k denotes the number of images provided in the support set. It is worth noting that during training only images that include at least one pixel belonging to L_{train} are included in D_{train} for large-scale training. If some images have pixels labelled as classes belonging to L_{test} they are ignored and not used in the back-propagation. Our model does not need to be trained in the few-shot regime by sampling a support set and a query image. It is trained in a normal fashion with image-label pairs.

Continuous Semantic Segmentation. In continuous semantic segmentation scenario, we propose the setup based on PASCAL VOC [5], following the class incremental learning scenario described in [37]. We call the proposed setup incremental PASCAL (iPASCAL). It is designed to assess sample efficiency of a method in the continual learning setting. The classes in the dataset are split into L_{train} and $L_{incremental}$ with 10 classes each, where $L_{train} \cap L_{incremental} = \emptyset$. The classes belonging to the L_{train} are used to construct the training dataset D_{train} and pre-train the segmentation network. Unlike the static setting in the few-shot case, the continuous segmentation mode provides the image-label pairs incrementally with different encountered tasks. The tasks are in the form of triplets $(t_i, (X_i, Y_i))$, where (X_i, Y_i) represent the overall batch of images and labels from task t_i. Each task t_i introduces two novel classes to learn in its batch. That batch contains samples with at least one pixel belonging to these two novel classes. The labels per task t_i include the two novel classes belonging to that task, and the previously learned classes in the encountered tasks $t_0, ..., t_{i-1}$.

4. Experimental Results

We evaluate the sample efficiency of the proposed AMP method in three different scenarios: (1) few-shot segmentation, (2) video object segmentation, and (3) continuous semantic segmentation. In the few-shot segmentation scenario we evaluate on pascal-5^i [28] (see Section 4.1). An ablation study is performed to demonstrate the improvement resulting from multi-resolution imprinting and proxy adaptation in Section 4.2. The study also compares weight imprinting coupled with back-propagation against back-propagation on randomly generated weights. Section 4.4 demonstrates the benefit of AMP in the context of con-

Table 1: mIoU for 1-way 1-shot segmentation on PASCAL-5^i. FT: Fine-tuning. AMP-1 and AMP-2: our method using DFCN8s and Reduced-DFCN8s, respectively. Red, Blue: best and second best methods. co-FCN evaluation is from [41].

	1-NN [28]	Siamese [28]	FT [28]	OSLSM [28]	co-FCN [25]	AMP-1 (ours)	AMP-2 (ours)
Fold 0	25.3	28.1	24.9	33.6	36.7	37.4	**41.9**
Fold 1	44.9	39.9	38.8	**55.3**	50.6	50.9	50.2
Fold 2	41.7	31.8	36.5	40.9	44.9	46.5	**46.7**
Fold 3	18.4	25.8	30.1	33.5	32.4	**34.8**	34.7
Mean	32.6	31.4	32.6	40.8	41.1	**42.4**	**43.4**

Table 2: mIoU for 1-way 5-shot segmentation on PASCAL-5^i. FT: Fine-tuning. AMP-2 + FT(2): our method with 2 fine-tuning iterations, respectively. Red, Blue: best and second best methods. co-FCN evaluation is from [41].

	1-NN [28]	LogReg [28]	OSLSM [28]	co-FCN [25]	AMP-2 (ours)	AMP-2 + FT(2) (ours)
Fold 0	34.5	35.9	35.9	37.5	40.3	**41.8**
Fold 1	53.0	51.6	**58.1**	50.0	55.3	55.5
Fold 2	46.9	44.5	42.7	44.1	49.9	**50.3**
Fold 3	25.6	25.6	39.1	33.9	**40.1**	39.9
Mean	40.0	39.3	43.9	41.4	**46.4**	**46.9**

tinuous semantic segmentation on the proposed incremental PASCAL VOC evaluation framework, iPASCAL. We further evaluate AMP in the online adaptation scenario on DAVIS [22] and FBMS [20] benchmarks for video object segmentation (see Section 4.3). We use mean intersection over union (mIoU) [28] as evaluation metric unless explicitly stated otherwise. mIoU denotes the average of the per-class IoUs per fold. Our training and evaluation code is based on the semantic segmentation work [29] and is made publicly available [1].

4.1. Few-Shot Semantic Segmentation

The setup for training and evaluation on PASCAL-5^i is as follows. The base network is trained using RMSProp [9] with learning rate 10^{-6} and L2 regularization weight $5\text{x}10^{-4}$. For each fold, models are pretrained on 15 train classes and evaluated on remaining 5 classes, unseen during pretraining. The few-shot evaluation is performed on 1000 randomly sampled tasks, each including a support and a query set, similar to OSLSM setup [28]. A hyper-parameter random search is conducted over the α parameter, the number of iterations, and the learning rate. The search is conducted by training on 10 classes from the training set and evaluating on the other 5 classes of the training set. Thus ensuring all the classes used are outside the fold used in the evaluation phase. The α parameter selected is 0.26. In the case of performing fine-tuning, the selected learning rate is $7.6\text{x}10^{-5}$ with 2 iterations for the 5-shot case.

Tables 1 and 2 show the mIoU for the 1-shot and 5-shot segmentation, respectively, on PASCAL-5^i (mIoU is computed on the foreground class as in [28]). Our method is compared to OSLSM [28] as well as other baseline methods for few-shot segmentation. AMP outperforms the baseline fine-tuning [28] method by 10.8% in terms of mIoU, without the need for extra back-propagation iterations by directly using the adaptive masked proxies. AMP outperforms OSLSM [28] in both the 1-shot and the 5-shot cases. Unlike OSLSM, our method does not need to train an extra guidance branch. This advantage provides the means to use AMP with a 2-stream motion and appearance based network as shown in Section 4.3. On top of that, AMP outperforms co-FCN method [25].

Table 3 reports our results in comparison to the state-of-the-art using the evaluation framework of [25] and [4]. In this framework the mIoU is computed as the mean of the foreground and background IoU averaged over folds. AMP outperforms the baseline FG-BG [4] in the 1-shot and 5-shot cases. When our method is coupled with two iterations of back-propagation through the last layers solely it outperforms co-FCN [25] in the 5-shot case by 3%.

Qualitative results on PASCAL-5^i are demonstrated in Figure 5 that shows both the support set image-label pair, and segmentation for the query image predicted by AMP. Importantly, segmentation produced by AMP does not seem to depend on the saliency of objects. In some of the query images, multiple potential objects can be categorized as salient, but AMP learns to segment what best matches the target class.

4.2. Ablation Study

We perform an ablation study to demonstrate the effectiveness of different components in AMP. Results are reported in Table 4. For our final method, it corresponds to the evaluation provided in Tables 1 and 2 on fold 0,

[1] https://github.com/MSiam/AdaptiveMaskedProxies

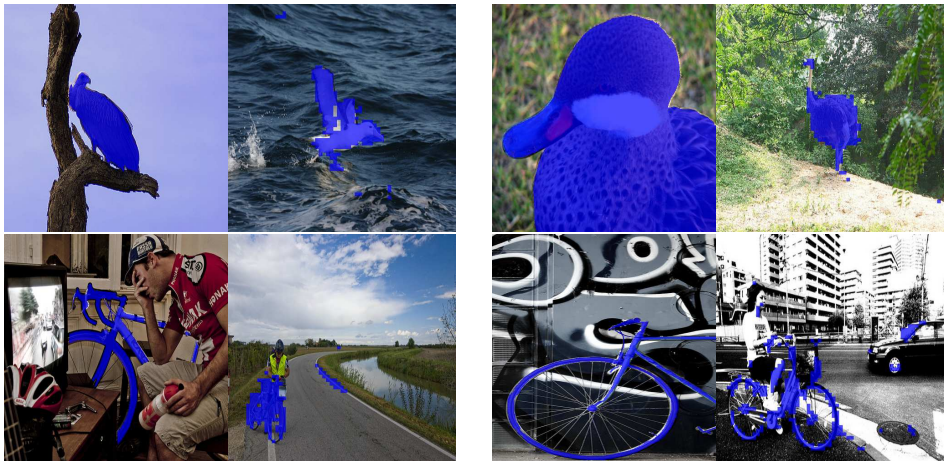

Figure 5: Qualitative evaluation on PASCAL-5^i 1-way 1-shot. The support set and prediction on the query image are shown.

Table 3: Quantitative results for 1-way, 1-shot and 5-shot segmentation on PASCAL-5^i dataset, following evaluation in [4]. FT: Fine-tuning for 2 iterations in 1-shot and 5-shot setting. Red, Blue: best and second best methods.

Method	1-Shot	5-Shot
FG-BG [4]	55.1	55.6
OSLSM [28]	55.2	-
co-FCN [25]	60.1	60.8
PL+SEG [4]	61.2	**62.3**
AMP-2 (ours)	**61.9**	62.1
AMP-2 + FT (ours)	**62.2**	**63.8**

Table 4: Ablation study of the different design choices for the imprinting scheme. Adaptation: α parameter is non-zero. Multi-res: performing multi-resolution imprinting. Imp: imprinting weights using our proxies. FT: fine-tuning.

Method	Adaptation	Multi-res.	N-Shot	mIoU
FT only	✗	✓	5	28.7
Imp.	✓	✓	5	40.3
Imp. + FT	✓	✓	5	**41.8**
Imp.	✗	✓	1	13.6
Imp.	✓	✗	1	34.8
Imp.	✓	✓	1	**41.9**

following Shaban et al. [28]. First, AMP clearly outperforms naïve fine-tuning using randomly generated weights by 11.6%. Second, AMP can be effectively combined with the fine-tuning of imprinted weights to further improve performance. This is ideal for a continuous data stream processing. Third, AMP's proxy adaptation component is effective: no adaptation with α set to 0, degrades accuracy by 28.3% in the 1-shot scenario. Finally, multi-resolution imprinting is effective: not performing multi-resolution imprinting degrades mIoU in the 1-shot scenario. We conclude that simply imprinting the weights only for the new class is not optimal. Imprinting has to be coupled with the proposed adaptation and multi-resolution schemes to be effective in the segmentation scenario.

4.3. Video Object Segmentation

To assess AMP in the video object segmentation scenario, we use it to adapt 2-stream segmentation networks based on pseudo-labels and evaluate on the DAVIS-2016 benchmark [22]. Here our base network is a 2-stream Wide ResNet model similar to [30]. We make the model adapt to the appearance changes that the object undergoes in the video sequence using the proposed proxy adaptation scheme with α parameter set to 0.001. The adaptation mechanism operates on top of the masked proxies derived from the segmentation probability maps output from the model itself, since the model has learned background-foreground segmentation already. Therefore, we call this "self adaptation" as it is unsupervised video object segmentation. Since we do not employ manual segmentation masks, we compare our results against the state-of-the-art unsupervised methods that utilize motion and appearance based models. Table 5 shows the mIoU over the validation set for AMP and the baselines. Our method when followed with fully connected conditional random fields [14] post processing outperforms the state of the art (the CRF post-processing is commonly applied by most methods evaluated on DAVIS'16).

Table 6 shows our self adaptation results on FBMS dataset where it outperforms all methods except for MotAdapt [30], which it is on-par with. These results uncover one of the weaknesses of our method: it is unable to operate with high dilation rates since it relies on masked proxies. High dilation rates can lead to interference between back-

Table 5: Quantitative comparison between unsupervised methods and the adaptive masked imprinting scheme on DAVIS'16.

Measure		FSeg [10]	LVO [36]	MOTAdapt [30]	ARP [13]	PDB [33]	AMP + CRF (Ours)
$\mathcal{J}$	Mean	70.7	75.9	77.2	76.2	77.2	**78.9**
	Recall	83.5	89.1	87.8	91.1	90.1	**91.6**
	Decay	1.5	7.0	5.0	7.0	**0.9**	4.7
$\mathcal{F}$	Mean	65.3	72.1	77.4	70.6	74.5	**78.4**
	Recall	73.8	83.4	84.4	83.5	84.4	**87.3**
	Decay	1.8	1.3	3.3	7.9	**0.2**	2.7

Table 6: Quantitative results on FBMS dataset (test set).

Measure	FST [21]	CVOS [34]	CUT [11]	MPNet-V[35]	LVO[36]	MotAdapt [30]	AMP (ours)
$\mathcal{P}$	76.3	83.4	83.1	81.4	**92.1**	80.7	82.7
$\mathcal{R}$	63.3	67.9	71.5	73.9	67.4	**77.4**	75.7
$\mathcal{F}$	69.2	74.9	76.8	77.5	77.8	**79.0**	**79.0**

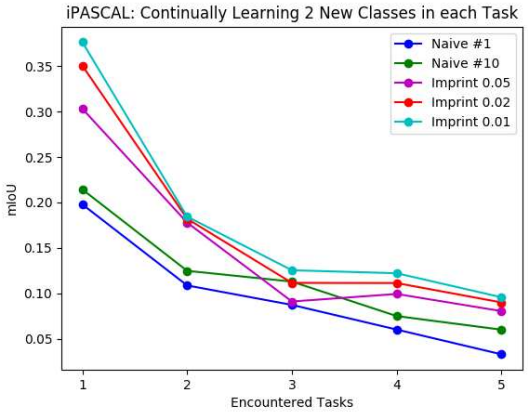

Figure 6: N-way evaluation on iPASCAL. Naive #M: fine-tuning with M iteration per sample. Imprint A: our method is used with α for the task classes set to A.

ground and foreground features in AMP. Another AMP's weakness is that it may face difficulties segmenting a specific instance, since it uses a proxy per class that aims to generalize across different instances.

4.4. Continuous Semantic Segmentation

To demonstrate the benefit of AMP in the continuous semantic segmentation scenario, we conducted experiments on iPASCAL. iPASCAL provides triplets for the task, the corresponding images and semantic labels. For each task, semantic labels include labels of new classes encountered in the current task as well as the labels of classes encountered in the previous tasks (please see Section 3.5 for more details on the setup definition). Figure 6 compares naïve fine-tuning from random weights against AMP without any fine-tuning, in terms of mIoU (average over 5 runs). Multiple runs are evaluated with different seeds that control random assignment of unseen classes in new tasks. The mIoU is reported per task on all the classes learned up to the current task. Fine-tuning was conducted using RMSProp with the best learning rate from the 1-shot setup $9.06\text{x}10^{-5}$. Fine-tuning is applied to the last layers responsible for pixel-wise classification, while the feature extraction weights are kept fixed. We are focusing on improving sample efficiency by imprinting the weights of the final layer, therefore we perform the fine-tuning on the final weights only. Figure 6 demonstrates that in the continual learning scenario, weight imprinting via AMP is more effective than fine-tuning, which suffers from over-fitting that is very hard to overcome.

It is worth noting that the current evaluation setting is a n-way where n increases with 2 additional classes with each encountered task resulting in 10-way evaluation in the last task. This explains the difference between the mIoU in Table 1 and Figure 6, which we attribute to the fact that n-way classification is more challenging than 1-way.

5. Conclusion

In this paper we proposed a sample efficient method to segment unseen classes via multi-resolution imprinting of adaptive masked proxies (AMP). AMP constructs the final segmentation layer weights from few labelled support set samples by imprinting the masked multi-resolution response of the base feature extractor and by fusing it with the previously learned class signatures. AMP is empirically validated to be superior in the few-shot segmentation on PASCAL-5^i with 5.5% in 5-shot case. It is also validated on video object segmentation on DAVIS16 as well as on the proposed iPASCAL.

References

[1] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE, 2010.

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

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

[4] Nanqing Dong and Eric P. Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, page 4, 2018.

[5] 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*, 111(1):98–136, 2015.

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[7] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[8] Charles F. Hester and David Casasent. Multivariant technique for multiclass pattern recognition. *Applied Optics*, 19(11):1758–1761, 1980.

[9] Geoff Hinton. Neural Networks for Machine Learning, Lecture Notes: overview of mini-batch gradient descent. URL: `https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

[10] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *arXiv preprint arXiv:1701.05384*, 2017.

[11] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3271–3279, 2015.

[12] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[13] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction.

[14] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[15] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[17] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[18] Ellen M. Markman. *Categorization and naming in children: Problems of induction*. MIT Press, 1989.

[19] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017.

[20] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2014.

[21] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

[22] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[23] Hang Qi, Matthew Brown, and David G Lowe. Learning with imprinted weights. *arXiv preprint arXiv:1712.07136*, 2017.

[24] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations. *arXiv preprint arXiv:1706.03466*, 2, 2017.

[25] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. 2018.

[26] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[27] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016.

[28] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.

[29] Meet P. Shah. Semantic segmentation architectures implemented in PyTorch. *https://github.com/meetshah1995/pytorch-semseg*, 2017.

[30] Mennatullah Siam, Chen Jiang, Steven Lu, Laura Petrich, Mahmoud Gamal, Mohamed Elhoseiny, and Martin Jagersand. Video segmentation using teacher-student adaptation in a human robot interaction (hri) setting. *arXiv preprint arXiv:1810.07733*, 2018.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[33] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 715–731, 2018.

[34] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4268–4276, 2015.

[35] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. *arXiv preprint arXiv:1612.07217*, 2016.

[36] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. *arXiv preprint arXiv:1704.05737*, 2017.

[37] Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning. *CoRR*, abs/1904.07734, 2019.

[38] Oriol Vinyals, Charles Blundell, Tim Lillicrap, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[39] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[40] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[41] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas Huang. SG-One: Similarity guidance network for one-shot semantic segmentation. *arXiv preprint arXiv:1810.09091*, 2018.