# Supplementary Materials for Memory-augmented Variational Adaptation for Online Few-shot Segmentation

Jie Liu[1], Yingjun Du[1], Zehao Xiao[1], Cees G.M Snoek[1], Jan-Jakob Sonke[2], Efstratios Gavves[1]
[1]University of Amsterdam, Netherlands [2]The Netherlands Cancer Institute, Netherlands
[1]{j.liu5, y.du, z.xiao, cgmsnoek, egavves}@uva.nl [2]j.sonke@nki.nl

## 1. Derivations of ELBO

For the sample $x_t$ at time step $t$, we begin with maximizing log-likelihood of the conditional distribution $\log p(y_t|x_t, \mathcal{M}_t)$ to derive the ELBO. By applying Jensen's inequality, we have the following steps as

$$\begin{aligned}
&\log p(y_t|x_t, \mathcal{M}_t) \\
&= \log \int p(y_t|x_t, w_t) p(w_t|x_t, \mathcal{M}_t) dw_t \\
&= \log \int p(y_t|x_t, w_t) \frac{p(w_t|x_t, \mathcal{M}_t)}{q(w_t|x_t, y_t, \mathcal{M}_t)} q(w_t|x_t, y_t, \mathcal{M}_t) dw_t \\
&\geq \int \log[\frac{p(y_t|x_t, w_t) p(w_t|x_t, \mathcal{M}_t)}{q(w_t|x_t, y_t, \mathcal{M}_t)}] q(w_t|x_t, y_t, \mathcal{M}_t) dw_t \\
&= \mathbb{E}_{q(w_t)}[p(y_t|x_t, w_t)] - \mathbb{D}_{KL}[q(w_t|x_t, y_t, \mathcal{M}_t)||p(w_t|x_t, \mathcal{M}_t)],
\end{aligned} \tag{1}$$

which is consistent with Eq.4 as in the main paper.

## 2. Implementation

### 2.1. Datasets details

Pascal-$5^i$ and COCO-$20^i$ are two widely-used benchmarks in traditional few-shot segmentation (FSS). Cross validation is adopted in FSS to test model performance on different novel classes, we provide the class split of different folds in Table 1 and Table 2, respectively. In online few-shot segmentation (OFSS), we adopt two nature image dataest (PASCAL and COCO) and one medical image dataset ABD-MR-20 to verify the effectiveness of online few-shot segmentation models. For PASCAL and COCO, we implement most experiments on the fold-0 of Pascal-$5^i$ and COCO-$20^i$, i.e., classes in fold-0 serve as testing classes, while remaining classes are training classes. We also provide results on different folds in Table 6, and more detailed results can be found in Table. ABD-MRI-20 is a MRI dataset from ISBI 2019 Combined Healthy Abdominal Organ Segmentation Challenge [2]. We choose spleen as the testing class, Liver, left and right kidney as training classes. Furthermore, we adopt 5 scans with spleen for evaluation and remaining 15 scans with other organs for training.

| Fold | Testing (novel) classes |
|---|---|
| Fold-0 | Aeroplane, Bicycle, Bird, Boat, Bottle |
| Fold-1 | Bus, Car, Cat, Chair, Cow |
| Fold-2 | Diningtable, Dog, Horse, Motorbike, Person |
| Fold-3 | Potted plant, Sheep, Sofa, Train, Tvmonitor |

Table 1. Testing classes split for each fold in PASCAL-$5^i$ dataset.

| Fold | Testing (novel) classes |
|---|---|
| Fold-0 | Person, Airplane, Boat, Parking meter, Dog, Elephant, Backpack,Suitcase, Sports Ball, Skateboard, Wine glass, Spoon, Sandwich, Hot dog, Chair, Dining table, Mouse, Microwave, Scissorse |
| Fold-1 | Bicycle, Bus, Traffic light, Bench, Horse, Bear, Umbrella, Frisbee, Kite, Surfboard , Cup, Bowl, Orange, Pizza, Couch, Toilet, Remote, Oven, Book, Teddy bear |
| Fold-2 | Car, Train, Fire hydrant, Bird, Sheep, Zebra, Handbag, Skis, Baseball bat, Tennis racket, Fork, Banana, Broccoli, Donut, Potted plant, Tv, Keyboard, Sink, Toaster, Clock, Hair drier |
| Fold-3 | Motorcycle, Truck, Stop sign, Cat, Cow, Giraffe, Tie, Snowboard, Baseball glove, Bottle, Knife, Apple, Carrot, Cake, Bed, Laptop, Cell phone, Refrigerator, Vase, Toothbrush |

Table 2. Testing classes split for each fold in COCO-$20^i$ dataset.

### 2.2. Implementation details

**Task setup** Online few-shot segmentation takes sequential samples as input and outputs mask prediction for each sample in the sequence. All samples in a specific sequence contain the same class object. Denoting the length of the input sequence as $T$, we set $T = 6$ at both training and testing stage. For natural image datasets, we randomly sample thousands of sequences from training classes to train our model at the training stage. At the testing stage, we randomly sample 1000 sequences from novel classes to evaluate model performance. The input resolutions of the model is set as 473×473. For the medical dataset, we focus on the segmentation of 2D slices. At the training stage, we first select one 3D MRI scan, then randomly sample $T$ 2D slices that contain the target organ as one sequence. At the testing stage, we set the testing number of sequences as 100, and the input resolution is 200×200. **Training details** We train all baseline models and the proposed model with learning rate 0.0025 for 100 and 50 epochs on PASCAL and COCO, respectively. For experiments on ABD-MRI-20, we set the

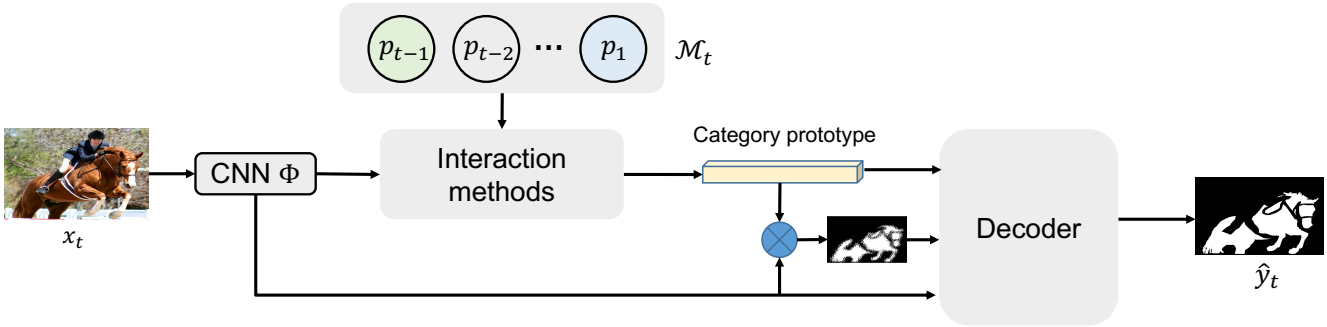


Figure 1. **Architecture of baseline models**. We compare our model with four baseline models, which adopt different interaction methods between the prototype memory $\mathcal{M}_t$ and CNN features of the current sample to generate the category prototype.

| Method | 0-shot | 1-shot | 2-shot | 3-shot | 4-shot | 5-shot | 6-shot | 7-shot | 8-shot | 9-shot | 10-shot | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FSS-1shot | 0 | 53.62 | 53.27 | 53.93 | 53.97 | 53.44 | 53.27 | 53.53 | 53.47 | 53.30 | 53.43 | 53.51 |
| FSS-5shot | 0 | 53.47 | 56.40 | 56.87 | 57.27 | 57.17 | 57.27 | 57.43 | 57.33 | 57.57 | 57.33 | 56.91 |
| OPN | 35.47 | 52.63 | 55.87 | 57.97 | 56.47 | 59.30 | 58.62 | 59.53 | 58.32 | 58.64 | 58.23 | 57.55 |
| LSTM | 39.70 | 55.40 | 57.37 | 58.97 | 57.37 | 59.63 | 59.20 | 60.03 | 58.37 | 58/50 | 58.03 | 58.29 |
| PIFS | 40.09 | 57.09 | 61.60 | 58.83 | 60.25 | 60.66 | 62.77 | 62.01 | 61.69 | 60.52 | 60.78 | 60.62 |
| Ours | 49.39 | 59.41 | 60.21 | 62.82 | 61.42 | 62.91 | 62.48 | 62.91 | 62.40 | 61.74 | 62.06 | 61.83 |

Table 3. **Per step results on PASCAL**. We report the results from 0-shot to 10-shot and the mean of 1-shot to 10-shot. Our method achieves consistent best performance. mIoU is adopted as metric.

| Method | 0-shot | 1-shot | 2-shot | 3-shot | 4-shot | 5-shot | 6-shot | 7-shot | 8-shot | 9-shot | 10-shot | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FSS-1shot | 0 | 33.38 | 37.64 | 38.42 | 37.72 | 38.06 | 38.86 | 39.78 | 39.14 | 38.63 | 39.64 | 38.13 |
| FSS-5shot | 0 | 38.76 | 41.98 | 43.09 | 44.08 | 43.90 | 44.67 | 45.26 | 45.21 | 44.08 | 45.34 | 43.64 |
| OPN | 11.02 | 39.59 | 44.37 | 42.60 | 42.53 | 45.22 | 44.56 | 45.11 | 46.01 | 47.06 | 45.13 | 44.22 |
| LSTM | 0.09 | 35.52 | 41.20 | 41.45 | 44.10 | 44.64 | 45.00 | 45.67 | 47.14 | 47.71 | 46.04 | 43.84 |
| PIFS | 22.86 | 40.15 | 45.83 | 42.45 | 45.12 | 46.73 | 45.62 | 45.27 | 46.92 | 46.74 | 46.13 | 45.09 |
| Ours | 25.17 | 43.08 | 47.57 | 45.96 | 46.71 | 49.17 | 48.46 | 48.30 | 49.90 | 49.93 | 48.82 | 47.79 |

Table 4. **Per step results on COCO**.We report the results from 0-shot to 10-shot and the mean of 1-shot to 10-shot. Our method achieves consistent best performance. mIoU is adopted as metric.

learning rate and training epochs as 0.0025 and 100, respectively. We adopt ResNet50[1] pretrained on ImageNet [3] as backbone network to extract features. The backbone is frozen for experiments on PASCAL and COCO to avoid the model outfitting to training classes. For experiments on ABD-MRI-20, we fine-tune the backbone network to learn robust feature representation for medical segmentation. For all three datasets, we set the Monte Carlo sampling number $L = 100$.

### 2.3. Baseline models

In our experiments, we compare the proposed MaVAN with one classical few-shot segmentation baseline (trained under 1-shot and 5-shot setting) and three online few-shot segmentation baseline models, i.e., Online Prototyical Network (OPN), LSTM, and PIFS. Both baseline models and the proposed MaVAN share similar architecture as shown in Figure 1 with different modification. For classical few-shot segmentation model baseline, we remove the prototype memory $\mathcal{M}_t$ interaction methods in Figure 1. the category prototype is directly generated from masked support images with mask average pooling. For the online few-shot segmentation baseline, OPN obtains the category prototype by averaging sample prototypes in the prototype memory, while LSTM adopt a single-layer LSTM to interact with the prototype memory to update the category prototype. For PIFS, we introduce the prototype-based distillation loss on both old and new sample prototypes. In practice, the old sample prototype is obtained by averaging the prototype memory, while the new sample prototype is obtained by applying global average pooling over sample feature. The decoder network is composed of three consecutive convolutions layers followed by a ASPP and a 1x1 convolution layers which is used for mask prediction.

In Figure 4 (a) and (b) of the main paper, we compare our model with classical few-shot segmentation models trained under 1-shot and 5-shot settings. When the number of samples increases over time, we directly average support fore-

| Method | 0-shot | 1-shot | 2-shot | 3-shot | 4-shot | 5-shot | 6-shot | 7-shot | 8-shot | 9-shot | 10-shot | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PFENet | 0 | 40.73 | 42.36 | 32.89 | 39.17 | 43.63 | 38.15 | 40.83 | 36.21 | 38.86 | 39.09 | 46.29 |
| OPN | 13.65 | 35.40 | 39.72 | 30.95 | 34.73 | 36.86 | 32.88 | 35.89 | 30.78 | 33.15 | 32.57 | 34.29 |
| LSTM | 12.29 | 34.66 | 37.80 | 29.08 | 32.23 | 35.82 | 30.92 | 33.97 | 30.11 | 31.72 | 31.07 | 32.74 |
| PIFS | 15.44 | 38.19 | 42.32 | 31.78 | 36.49 | 38.07 | 34.15 | 37.29 | 31.92 | 34.31 | 33.62 | 35.81 |
| Ours | 18.42 | 39.57 | 44.94 | 34.48 | 38.90 | 41.26 | 36.53 | 39.72 | 34.87 | 36.53 | 36.38 | 38.32 |

Table 5. **Per step results on ABD-MRI-20**.We report the results from 0-shot to 10-shot and the mean of 1-shot to 10-shot. Our method achieves consistent best performance. Dice score is adopted as metric.

| Settings | PASCAL | | | | COCO | | | |
|---|---|---|---|---|---|---|---|---|
| | Fold-0 | Fold-1 | Fold-2 | Fold-3 | Fold-0 | Fold-1 | Fold-2 | Fold-3 |
| Naive classifier | 57.82 ±0.04 | 66.15 ±0.26 | 52.35 ±0.12 | 49.66 ±0.26 | 41.74 ±0.36 | 37.23 ±0.24 | 16.43 ±0.45 | 25.20 ±0.37 |
| **Variational Test-time adaptation** | **61.83** ±0.10 | **68.87** ±0.31 | **53.17** ±0.05 | **51.46** ±0.32 | **47.79** ±0.29 | **41.14** ±0.38 | **18.63** ±0.11 | **27.60** ±0.42 |

Table 6. **Cross validation on different unseen classes. For each fold, testing samples come from different unseen classes.** Our method consistently outperforms baseline method on different folds of PASCAL and COCO datasets.

ground prototypes to get the category prototype. For instance, at time step $t = 5$, we first obtain foreground prototypes of previous four samples, then we average four prototypes to get the category prototype, which is finally used to preform the segmentation of the fifth sample.

# 3. More results

## 3.1. Per step results

We report per step results of our model and baseline models in Table 3, Table 4, and Table 5 for PASCAL, COCO, and ABD-MRI-20, respectively. At the same time, we set the total time steps as 11 for evaluation to test the model performance over long sequence. As shown in above Tables, our models achieves considerably better performance than baseline models in all three dataset. Our model achieves substantial performance improvement with time step increases, even though experiences some fluctuation. This attributes to the capacity of our model in generating sample-specific weights for each sample in the sequence. Interestingly, our model also learns to distinguish salient objects from complex backgrounds. For zero-shot segmentation, in which online few-shot segmentation models give random guess on the first image of a specific novel class, our model also achieves best performance.

## 3.2. Cross validation on different unseen classes

To investigate the effectiveness of our model on different novel classes, we implement cross validation on unseen classes and report results in Table 6. We compare our method with naive classifier implemented with a $1\times1$ convolutional layer, i.e., test-time adaptation vs. naive classifier. As shown in Table 6, our method achieves the best performance across different folds on both PASCAL and COCO datasets. We can conclude that our model shows superior performance for online few-shot segmentation and is robust to different novel classes.

## 3.3. Visualization

We provide more visualization of the segmentation process of our model in dealing with a sequence of samples. Examples are shown in Figure 2 and Figure 3, respectively. We can see from the visualization that our model can effectively tacking the sample diversity problem with providing sample-specific weights for each sample. With time step increases, our model makes more and more accurate predictions on coming samples.

# References


[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[2] A Emre Kavur, N Sinem Gezer, Mustafa Barış, Sinem Aslan, Pierre-Henri Conze, Vladimir Groza, Duc Duy Pham, Soumick Chatterjee, Philipp Ernst, Savaş Özkan, et al. Chaos challenge-combined (ct-mr) healthy abdominal organ segmentation. *Medical Image Analysis*, 69:101950, 2021. 1

[3] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2

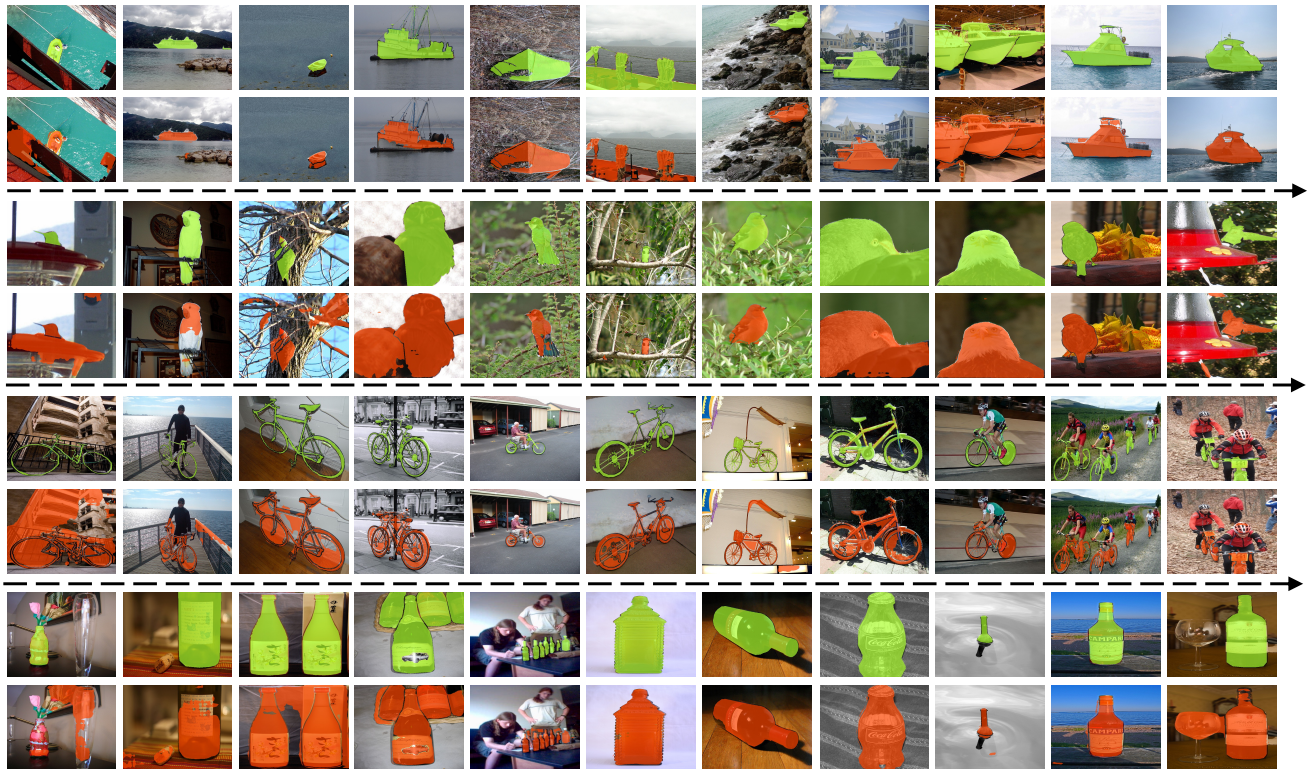

Figure 2. **Visualization of online few-shot segmentation performance on PASCAL.** Ground-truths are masked in green and predictions are masked in red. The length of sequence is set as $T = 11$, and 0-shot to 10-shot results are reported. The input sequence exhibits large sample diversity, our model shows superior capacity in tacking this problem.

Figure 3. **Visualization of online few-shot segmentation performance on COCO.** Ground-truths are masked in green and predictions are masked in red. The length of sequence is set as $T = 11$, and 0-shot to 10-shot results are reported. The input sequence exhibits large sample diversity, our model shows superior capacity in tacking this problem.