Supplementary Material for Attribution-aware Weight Transfer: A Warm-Start Initialization for Class-Incremental Semantic Segmentation

Dipam Goswami†§ René Schuster‡ Joost van de Weijer‡ Didier Stricker‡

dipamgoswami01@gmail.com rene.schuster@dfki.de joost@cvc.uab.es didier.stricker@dfki.de

† DFKI - German Research Center for Artificial Intelligence, Kaiserslautern
§ Birla Institute of Technology and Science, Pilani ‡ Computer Vision Center, Barcelona

Introduction

In this supplementary material to our main paper Attribution-aware Weight Transfer: A Warm-Start Initialization for Class-Incremental Semantic Segmentation, we discuss the details of the gradient-based attribution method. Integrated Gradients [11] used in our Attribution-aware Weight Transfer (AWT) initialization. We further share more details of our implementation for better reproducibility, and perform additional ablative experiments to analyze the impact of the proposed warm-start initialization. Finally, we present the qualitative results of AWT with a given target class (background class in our method).

2. Reproducibility

Datasets: We evaluate our models on Pascal-VOC 2012 [8], ADE20K [13] and Cityscapes [5]. VOC contains 10,582 images for training and 1,449 images for testing. ADE20K contains 20,210 and 2,000 images for training and testing respectively. Cityscapes contains 2,975 training images and 500 testing images.


We re- implement SSUL by training for 60 epochs on ADE20K dataset. We follow the same training settings for SSUL as proposed in [3] for VOC and ADE20K. For Cityscapes, we trained SSUL with a learning rate of 0.01 and a batch size of 24. We train the other models of FT, PLOP, RCIL for Cityscapes with SGD and a learning rate of 2 × 10^{-2} for the first step only and 10^{-3} for the incremental steps.

Class order: For all the quantitative experiments, we order the classes by increasing class id, i.e. the default order of the respective datasets.

For the ablation experiment using random orders on VOC 15-1, we sampled the following 10 class sequences:

```plaintext
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
[12, 9, 20, 7, 15, 8, 14, 16, 5, 19, 4, 1, 13, 2, 11, 17, 3, 6, 18, 10]
[13, 19, 15, 17, 9, 8, 5, 20, 4, 3, 10, 11, 18, 16, 7, 12, 14, 6, 1, 2]
[15, 3, 2, 12, 14, 18, 20, 16, 11, 1, 19, 8, 10, 7, 17, 6, 5, 13, 9, 4]
[7, 13, 5, 11, 9, 2, 15, 12, 14, 3, 20, 1, 16, 4, 18, 8, 6, 10, 19, 17]
[7, 5, 9, 1, 15, 18, 14, 3, 20, 10, 4, 19, 11, 17, 16, 12, 8, 6, 2, 13]
[12, 9, 19, 6, 4, 10, 5, 18, 14, 15, 16, 3, 8, 7, 11, 13, 2, 20, 17, 1]
```

Layer Integrated Gradients: Layer Integrated Gradients [10] is designed for computing attributions corresponding to inputs or outputs of a specific layer of the network. For a given layer, the size of the attribution maps is the same as the layer’s input or output dimensions, based on whether we attribute to the inputs or outputs of that layer. In our method, we compute the attributions for the inputs to the final classifier layer. We obtain the attributions corresponding to a given target class (background class in our method).
Table 7: Ablation study for significance of weight transfer on Pascal-VOC 2012.

<table>
<thead>
<tr>
<th>New Classifier Init</th>
<th>Iterations</th>
<th>VOC (15-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0-15 16-20 all</td>
</tr>
<tr>
<td>Random</td>
<td>× 1</td>
<td>45.7 5.3 36.1</td>
</tr>
<tr>
<td>Random</td>
<td>× 2</td>
<td>39.7 6.6 31.8</td>
</tr>
<tr>
<td>Random</td>
<td>× 4</td>
<td>29.9 7.5 24.6</td>
</tr>
<tr>
<td>Weight transfer - MiB [2]</td>
<td>× 1</td>
<td>48.1 15.8 40.4</td>
</tr>
<tr>
<td>Weight transfer - AWT (Ours)</td>
<td>× 1</td>
<td><strong>59.1 17.2 49.1</strong></td>
</tr>
</tbody>
</table>

Table 8: Ablation study for selection of threshold using MiB+AWT on Pascal-VOC 2012.

<table>
<thead>
<tr>
<th>Threshold for channel selection</th>
<th>VOC (15-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0-15 16-20 all</td>
</tr>
<tr>
<td>Top 10%</td>
<td>51.0 11.0 41.5</td>
</tr>
<tr>
<td>Top 25%</td>
<td><strong>59.1 17.2 49.1</strong></td>
</tr>
<tr>
<td>Top 50%</td>
<td>58.3 17.6 48.6</td>
</tr>
<tr>
<td>Top 75%</td>
<td>56.8 14.9 46.8</td>
</tr>
</tbody>
</table>

[13, 10, 15, 8, 7, 19, 4, 3, 16, 12, 14, 11, 5, 20, 6, 2, 18, 9, 17, 1]
[1, 14, 9, 5, 2, 15, 8, 20, 6, 16, 18, 7, 11, 10, 19, 3, 4, 17, 12, 13]
[16, 13, 1, 12, 18, 6, 14, 3, 7, 9, 20, 19, 15, 4, 2, 10, 8, 17]

3. Additional Ablation Experiments

Additional experiments are performed to analyze the effect of the initialization and the number of training iterations per step. We show in Table 7 that training the model with random initialization for a higher number of iterations (×2, ×4) cannot reach the performance of AWT initialization or even the one proposed by [2]. Instead, training for more iterations causes higher forgetting of old classes.

Furthermore, we vary the threshold $k$ to select the most significant 10%, 25%, 50% and 75% of the channels for weight transfer. Based on the results of this experiment shown in Table 8, our final AWT uses a ratio of 25% for all our experiments in the main paper.

To discuss the role of AWT on reducing the effect of background shift, we analyze the performance of the newly added classes after every step of training for VOC 15-1 and ADE20K 100-10 settings in Figure 7. We observe that MiB+AWT learns the new set of classes which transitions from the previous background to current foreground. This indicates reduced effect of the background shift with AWT across multiple steps.

4. Additional Qualitative Evaluation

Figure 8 shows the comparison of predictions using MiB, MiB+AWT, SSUL, and SSUL+AWT on some test samples of Pascal-VOC 2012 using models trained in the 10-1 setting. Over both the methods, AWT improves the predictions for multiple classes like TV, car, aeroplane, bird, chair, table, horse, person, dog, and many more.

References


Figure 8: Visualization of predictions using MiB, MiB+AWT, SSUL and SSUL+AWT in 10-1 setting on test images of Pascal-VOC 2012.


