1. Implementation Details

1.1. Architecture

The architecture of GroupViT is based on ViT-S [7, 14] with 12 Transformer layers. Each layer consists of a multi-head self-attention block and an MLP block. The input to each block is normalized by layer normalization [1]. We connect the group tokens in the different grouping stages via MLP-Mixer layers [13]. Our text-encoder consists of 12 Transformer layers, each with a hidden dimension of 256. Following [10], the Transformer operates on a lower-cased byte pair encoding (BPE) representation of the text with a vocabulary of 49,152 words.

1.2. Fully-Supervised Transfer to Semantic Segmentation

To implement the baselines for fully-supervised transfer to semantic segmentation, we fine-tune the pre-trained ViT model jointly with a 1×1 convolutional layer appended to it for pixel-wise classification. We scale each input image by a randomly selected factor in the range of [0.5, 2] and then crop random 224×224 patches from each image during training. We use the Adam [8] optimizer with a weight decay of 0.05 and a learning rate 0.001. We train all models for 4k iterations with a batch size of 16. During inference, we resize each input image to have a shorter side of size 448 pixels. We open-source our code at https://github.com/NVlabs/GroupViT.

2. Qualitative Results

PASCAL VOC 2012 We show additional qualitative results of GroupViT on the PASCAL VOC 2012 dataset in Fig. 2. We select some group tokens and highlight their attended regions across images from the same category in Fig. 3; and multiple objects from different categories in Fig. 4. Observe that GroupViT successfully groups and correctly classifies the objects in these various challenging scenarios.

*Jiarui Xu was an intern at NVIDIA during the project.

Supplementary to
GroupViT: Semantic Segmentation Emerges from Text Supervision

Jiarui Xu1*  Shalini De Mello2  Sifei Liu2  Wonmin Byeon2
Thomas Breuel2  Jan Kautz2  Xiaolong Wang1
1UC San Diego  2NVIDIA
Figure 2. **Qualitative Results of GroupViT on PASCAL VOC 2012.** The results in columns labeled “Stage 1/2” show grouping results prior to assigning labels, where the regions belonging to the same group are indicated by the same color. All these examples contain a single object from a category.
Figure 3. **Qualitative Results of GroupViT on PASCAL VOC 2012.** The results in columns labeled “Stage 1/2” show grouping results prior to assigning labels. The regions belonging to the same group are indicated by the same color. These examples contain multiple objects from the same category.
Figure 4. Qualitative Results of GroupViT on PASCAL VOC 2012. The results in columns labeled “Stage 1/2” show grouping results prior to assigning labels, where the regions belong to the same group are in the same color. These examples contain multiple objects from multiple different categories.

3.1. Image Classification

We compare the performances of the GroupViT and ViT architectures for the task of object classification on ImageNet. Following CLIP [10], here we train both architectures using supervision from text only via an image-text contrastive loss. In Table 1, we report both the zero-shot and the linear probing accuracy on the ImageNet [5] validation split. The zero-shot and linear probing evaluation follow the same setting as CLIP [10]. GroupViT’s ImageNet classification performance is comparable to (if not better than) that of ViT, thus demonstrating that our proposed grouping mechanism enhances the baseline ViT architecture with the capability to perform semantic pixel grouping and zero-shot transfer to semantic segmentation, without affecting its object classification performance.

<table>
<thead>
<tr>
<th>model</th>
<th>zero-shot Acc@1</th>
<th>linear Acc@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViT</td>
<td>42.4</td>
<td>69.2</td>
</tr>
<tr>
<td>GroupViT</td>
<td>42.9</td>
<td>69.8</td>
</tr>
</tbody>
</table>

Table 1. ImageNet Accuracy.

3.2. Mask Probing

We follow the procedure outlined in DINO [2] to evaluate the quality of the masks generated by GroupViT and by the baseline ViT model pre-trained using prior methods in a fully supervised [14], self-supervised [2, 4] or text-supervised [10] manner. For the ViT models, similar to DINO [2] for each final attention head, we compute its similarity to the [CLS] token and derive an attention mask for the pixels with the highest attention values. We then compute the Jaccard similarity of each head’s attention mask to the ground truth mask and retain the attention mask with the highest similarity. As for GroupViT, it does not have a multi-head design in the Grouping Block. Thus, we directly select the group most similar to the ground truth, as measured by the Jaccard index for each image. As Table 2 shows, the mask probing result of GroupViT is significantly better than that of all variants of the baseline ViT architecture. Hence, compared to ViT, our GroupViT more effec-
Figure 5. Qualitative Results of GroupViT on PASCAL Context. Columns labeled “Stage 1/2” show grouping results prior to assigning labels, where the regions belonging to the same group are indicated by the same color. GroupViT can successfully segment object and stuff classes, e.g. cat and window in row 2, dog and water in row 6.
Table 2. Comparison of mask probing performance. GroupViT outperforms all other variants of the baseline ViT architecture at effectively grouping image regions on semantic groups.

3.3. Limitations

We find that the mIoU of GroupViT on PASCAL Context is significantly lower than that on PASCAL VOC 2012. This could be attributed to the presence of background classes in PASCAL Context, e.g., ground, road and wall that result in low IoU (~1.5) on zero-shot transferring GroupViT to semantic segmentation on PASCAL Context. Through visual inspection, we find that while the pixels belonging to these background classes are typically correctly grouped into a single group by GroupViT, the group as a whole may be miss-classified into the wrong class on being compared to the text embedding of the various class labels. We hypothesize that this, in turn, happens due to the low probability of the background classes being described in textual sentences used during training. We show examples of such failure case in Fig. 6. We further conduct an oracle experiment to verify this finding. In the oracle experiment, for each output group from GroupViT, we compute its IoU with all ground truth masks and assign to each group the class label that results in the maximum IoU. This represents the upper bound of GroupViT’s performance since here we leverage ground truth masks and assign to each group the class label that results in the maximum IoU. This represents the upper bound of GroupViT’s performance since here we leverage ground truth masks to predict each group’s class label. We use our 2-stage GroupViT trained on the CC and YFCC datasets for this oracle experiment, which is the same model labeled “Ours” in Table 5 of the main paper. We report the oracle experiment’s results on PASCAL Context in Table 3. The large gap between the performance of the original and oracle mIoU values on the PASCAL Context dataset, shows that while GroupViT’s grouping results are reasonably good, there is room to further improve the groups’ classification to segmentation class labels via image-text embedding similarity computation.

Table 3. Original versus oracle results on PASCAL Context.

Table 4. Results on COCO Dataset.

3.4. COCO Dataset

We evaluate the performance of GroupViT on the COCO dataset [9], which contains 80 object classes. We combine the instance masks of the same category to get the semantic segmentation masks for each image. We report semantic segmentation mIoU on COCO in Table 4. It demonstrates that GroupViT is able to transfer to complex datasets with various number of classes.

Table 5. Results trained with CC+RedCaps.

References

Figure 6. **Failure cases on PASCAL Context.** “Oracle” shows the results of assigning groups to segmentation classes based on their IoU with the ground truth masks. Although GroupViT successfully groups *stuff* classes, e.g. ground, road and wall, it is not able to classify them correctly using the similarity between the visual and text embedding.


