A. Model Specifications

We provide details of the model specifications shown in Fig. 7 (elaborated on the Fig. 3 in the paper) and Fig. 8 (elaborated on the Fig. 4 in the paper).

B. Implementation Details

B.1. Experimental Details of Image Classification

For image classification in the IN1K [12], all models in Sec. 4.1 are trained on the `training set` for fair comparisons with the top-1 accuracy (%) on the `validation set`. The training receipt is adopted from DeiT [43], which has been widely used in training ViT variants. Table 5 shows the exact configurations used in our experiments. **Data Augmentation in Training:** we apply random cropping, random horizontal flipping [38], label-smoothing regularization [39], mixup [61], and random erasing [63] as data augmentations. During training, we employ AdamW [34] with a momentum of 0.9, a mini-batch size of 128, and a weight decay of 0.05 to optimize models. The initial base learning rate is set to 5×10^{-4} and decreases following the cosine schedule [33]. The drop-path regularization is also used [23]. All of our PaCa ViT models are trained for 300 epochs from scratch on 10 A100 GPUs with a learning rate auto-scaling heuristic method applied (see Table 5). **Evaluation:** We apply a single center crop (224×224) on the validation set in evaluating the classification accuracy. We us the latest `timm` package [49].

Config.	Value
batch_size	128
train_interpolation	'bicubic'
epochs	300
opt	'adamw'
opt_eps	1e-8
opt_betas	(0.9, 0.999)
momentum	0.9
weight_decay	0.05
auto_scale_lr	true
lr	5e-4
min_lr	5e-6
sched	'cosine'
warmup_epochs	5
warmup_lr	5e-7
cooldown_epochs	0
amp	True
clip_grad	none (T, S) / 1.0 (B)
clip_mode	norm
drop_path_rate	0.1 (T, S) / 0.5 (B)
color_jitter	0.4
smoothing	0.1
reprob	0.25
remode	'pixel'
recount	1
aa	'rand-m9-mstd0.5-inc1'
mixup	0.8
cutmix	1.0
mixup_prob	1.0
mixup_switch_prob	0.5
mixup_mode	'batch'

Table 5. Training configurations used in training the proposed PaCa ViT models in IN1K following the `timm` package [49]. We train three model specifications: Tiny (T), Small (S) and Base (B). This training receipt is adapted from [43] and often applied and tuned for training with 8 GPUs. We use 10 GPUs to take the full advantage of the server we have and to speed up the experiments. Accordingly, we apply a heuristic "auto_scale_lr" setting which scales "lr", "min_lr" and "warmup_lr" in this table with the factor "batch_size $\times$ nb_gpus / 512" (i.e., 2.25 in our settings) to account for the increased number of total images per batch with 10 GPUs used. We note that scaling these learning rate related hyperparamters often has slightly negative effects on performance.

B.2. Experimental Details of Object Detection and Instance Segmentation

We use the proposed PaCa ViT models (Tiny, Small and Base) as the feature backbones in the Mask R-CNN [21] and test them on the MS-COCO [31] dataset. All models in Sec. 4.2 are trained on MS-COCO `train2017` (118k images) and evaluated on `val2017` (5k images). We use the MMDetection [5] package (version 2.25.2) in experiments. We apply the weights pre-trained on IN1K to initialize the backbone and Xavier [16] in initializing the remaining layers in the Mask R-CNN (the default in the MMDetection). We adopt the 1x schedule in training (i.e., 12 epochs used in training). In both training and evaluation, the shorter side of the input image is fixed to 800 pixels with the longer side retained not exceeding 1,333 pixels. We train Mask R-CNN with our PaCa ViT backbones using batch size 16 on 8 A100 GPUs (i.e., 2 images per GPU) [3], following the recipes in the MMDetection package which use the AdamW [34] optimizer with an initial learning rate of 1×10^{-4}, and a weight decay 0.05. The parameters of the normalization layers are excluded from the weight decay.

[3]We follow the provided recipes and do not apply the auto-scaling heuristic to take advantage of the 10 GPUs we have on the server (that is done for IN1K training, see Table 5), since we observe the auto-scaling heuristic has more significantly negative impacts on performance on the downstream tasks and the training on the downstream tasks consumes much less time than that in IN1K.

B.3. Experimental Details of Image Semantic Segmentation

We use the proposed PaCa ViT models (Tiny, Small and Base) as the feature backbones and two different segmentation head sub-networks, the UpperNet [53] and our proposed PaCa segmentation head (Sec. 3.4). We test them on the MIT-ADE20k [64] dataset. In training, we randomly resize and crop images to the resolution of 512×512. In evaluation, images are resized to have a shorter side of 512

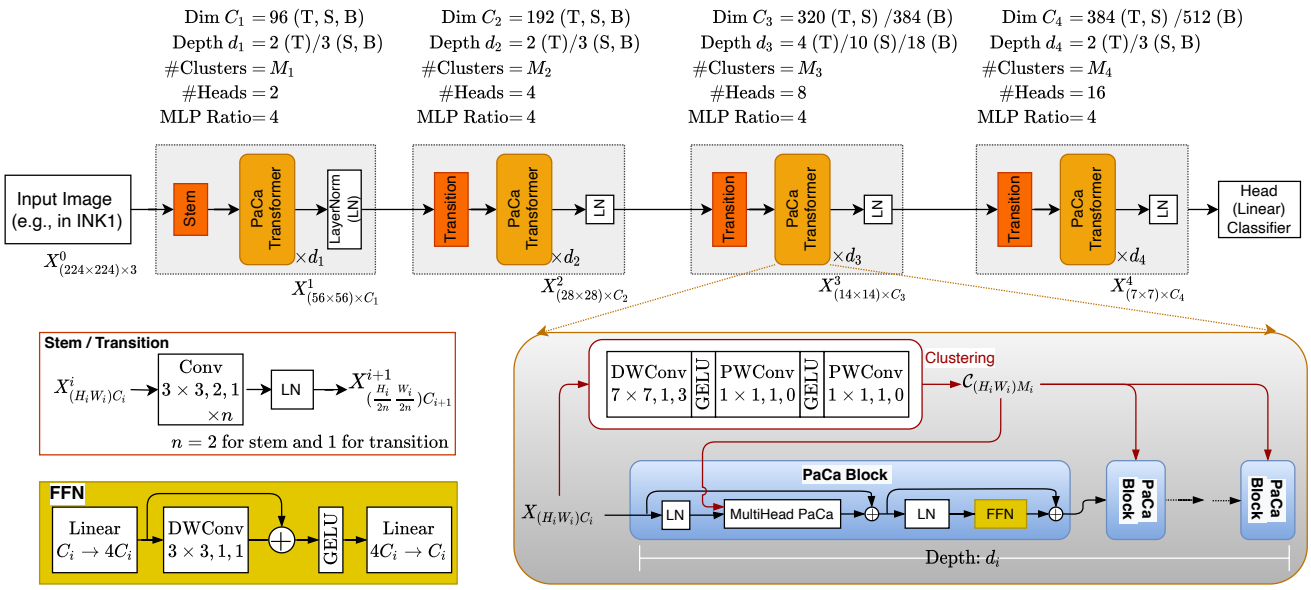

Figure 7. Mode specifications in the main experiments (elaborated on the Fig. 3 in the paper). We test three configurations: Tiny (T), Small (S) and Base (B). For the main results (see Tables 1, 2 and 3 in the paper), the number of clusters are $M_1 = M_2 = M_3 = 100$ and $M_4 = 0$ (i.e., degenerated back to the vanilla Transformer as done in the PVTv2 [47]), and the cluster assignment $\mathcal{C}_{(H_iW_i)M_i}$ is *shared* between all blocks in a stage as shown in the right-bottom. In the ablation studies, different configurations of the number of clusters at different stages are tested. A different clustering module based on a plain MLP is also tested (see Eqn. 6 in the paper). The FFN implementation is adapted from the Inverted Residual Block proposed in the MobileNetv2 [37], which is also used in PVTv2 [47]. We add the shortcut connection over the depth-wise convolution to induce it to play the role of positional encoding more faithfully as proposed in [10].

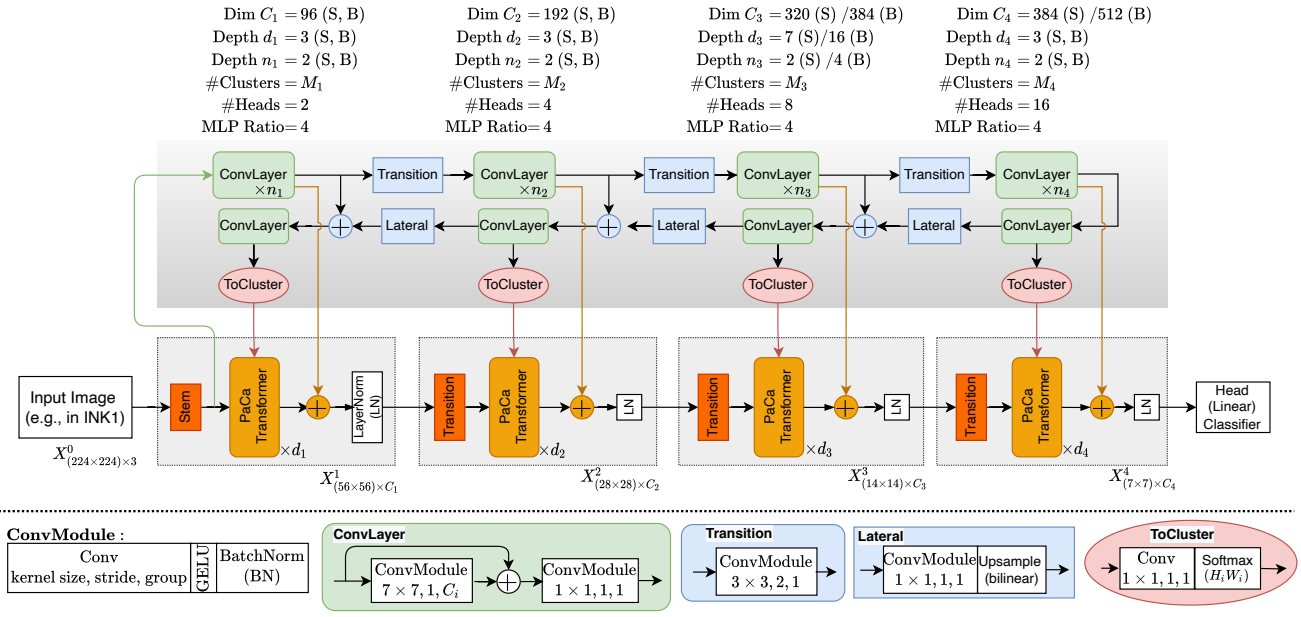

Figure 8. Mode specifications with external clustering teacher networks (elaborated on the Fig. 4 in the paper). We test two configurations: Small (S) and Base (B). The ViT branch has the same specifications as shown in Fig. 7. In the experiments, the number of clusters are $M_1 = M_2 = M_3 = M_4 = 100$, and the cluster assignment $\mathcal{C}_{(H_iW_i)M_i}$ from the teacher network is *shared* between all blocks in a stage. The "ConvLayer" module is adapted from the building block used in the ConvMixer [44].

pixels. The longer side is fixed not to exceed $2,048$ pixels. We use the MMSegmentation [11] package (version 0.29.0) in experiments. We apply the weights pre-trained on IN1K to initialize the backbone and Xavier [16] in initializ-

ing the head sub-network (the default in the MMSegmentation). We train our PaCa models with 160k iterations using batch size 16 on 8 A100 GPUs (i.e., 2 images per GPU). We adopt the default recipes provided in the MMSegmentation package, using the AdamW [34] optimizer with an initial learning rate of 6×10^{-5} for the backbone, and 6×10^{-4} for the head sub-network, and a weight decay 0.01. The parameters of the normalization layers are excluded from the weight decay. As mentioned in Sec. 4.3, we increase the number of clusters used in the backbone from 100 to 200 to account for the increased number of ground-truth classes in the MIT-ADE20k (150 classes). Due to this change, we set the initial learning rate to 6×10^{-4}, the same as the head sub-network, for the clustering layer (Eqn. 5).

C. Examples of Learned Clusters

We show all the clusters elaborating Fig. 6 in the paper in Figures 9 and 10.

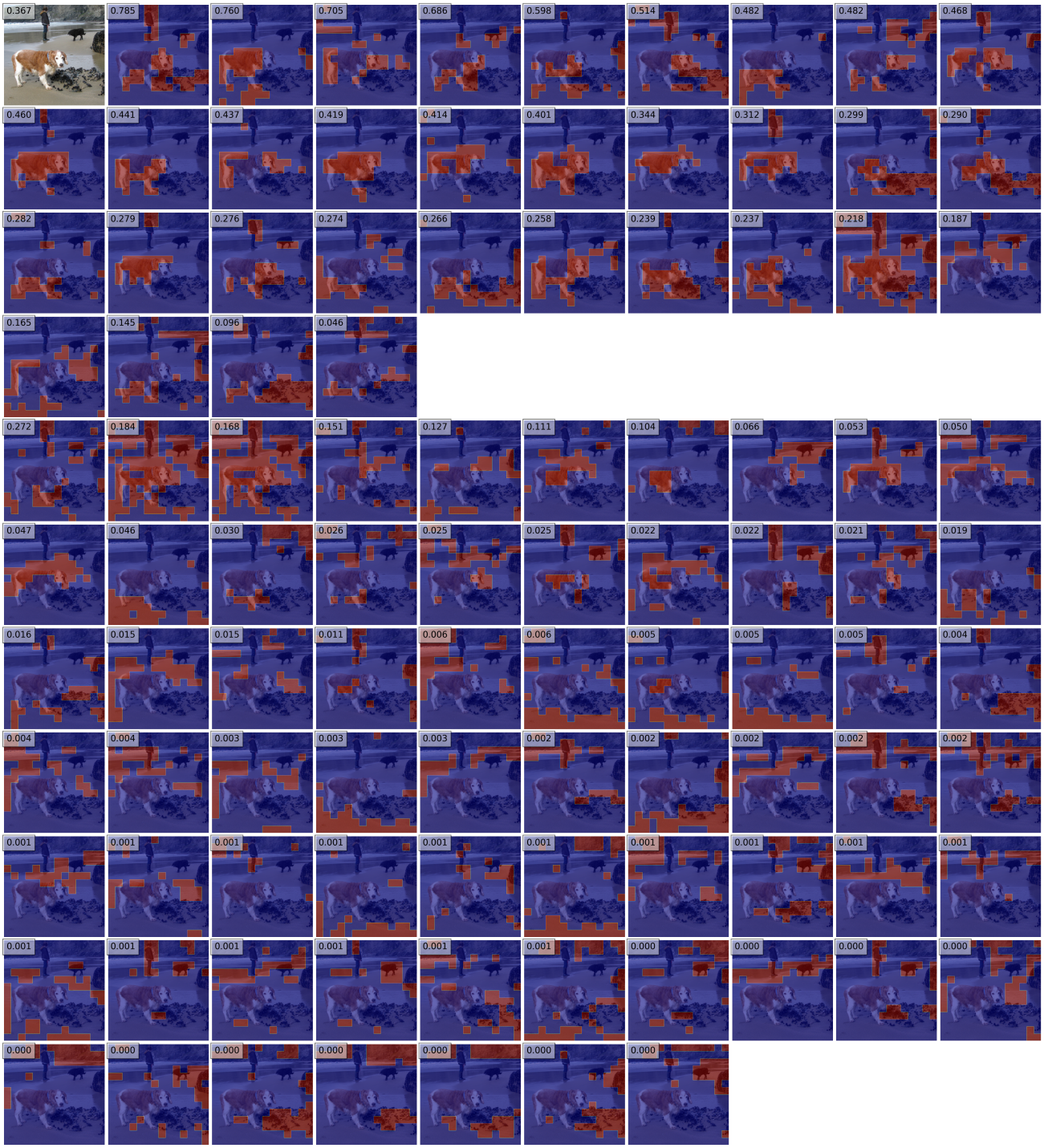

Figure 9. Visualizing the learned clusters with an image (id: 10933) in the IN1K validation set. We use the PaCa-Small network (Table 1). This image is correctly classified by the model. The 100 clusters learned at the third stage are used. The left-top image is the input image with the original predicted probability for the ground-truth class shown in the left-top box. The first 4 rows show the masked images in the positive group. It is interesting to see that many masked images can lead to higher predicted probabilities for the ground-truth class. The remaining rows show the masked images in the negative group. Although the first several images in the negative group have the predicted probabilities larger than some images in the positive group, the ground-truth class is not the top-1.

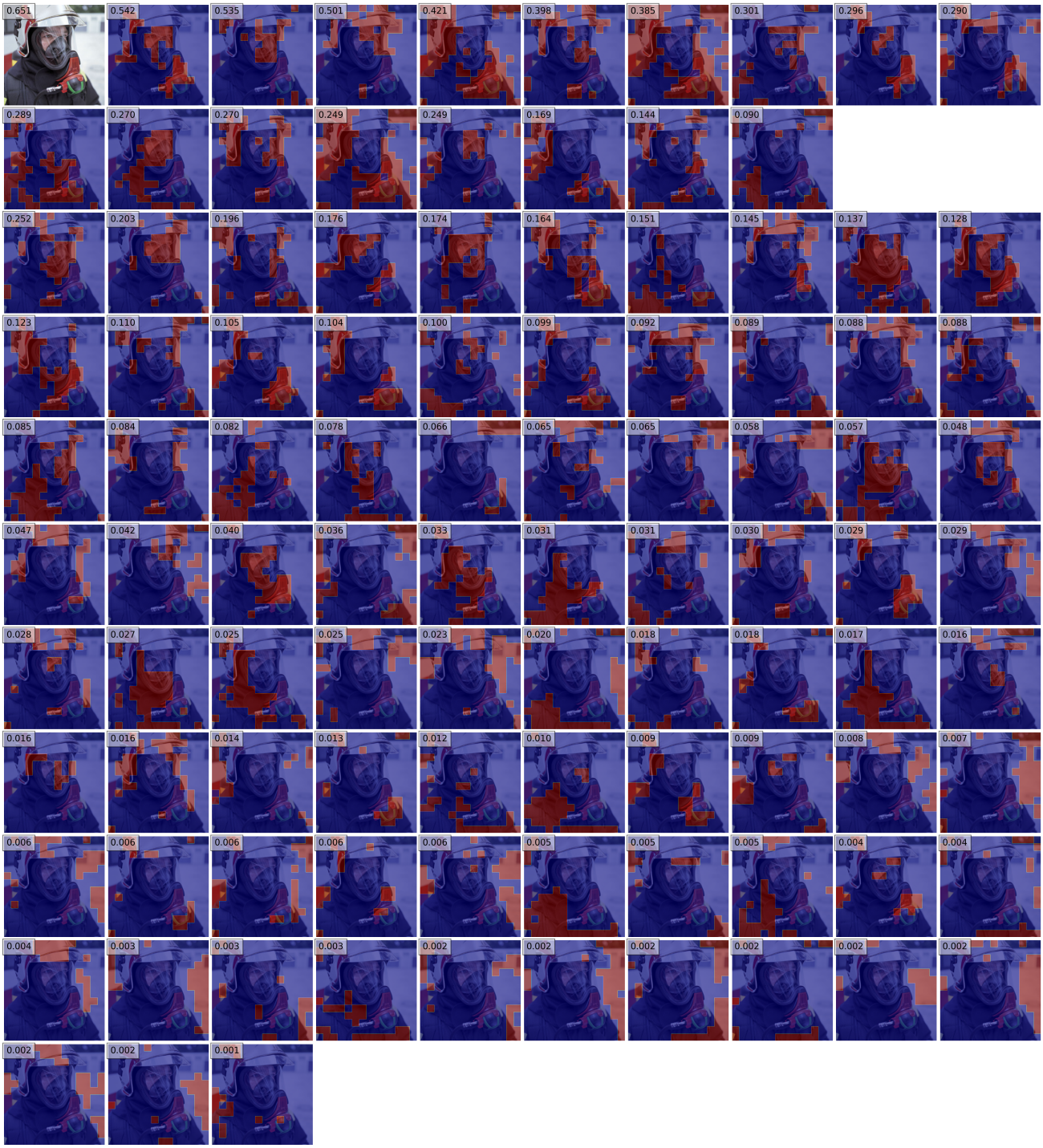

Figure 10. Visualizing the learned clusters with an image (id: 34561) in the IN1K validation set. We use the PaCa-Small network (Table 1). This image is correctly classified by the model. The 100 clusters learned at the third stage are used. The left-top image is the input image with the original predicted probability for the ground-truth class shown in the left-top box. The first 2 rows show the masked images in the positive group. For this examples, all the masked images have smaller predicted probabilities than the original unmaimage. The remaining rows show the masked images in the negative group. Although the first several images in the negative group have the predicted probabilities larger than some images in the positive group, the ground-truth class is not the top-1.