Masked Images Are Counterfactual Samples for Robust Fine-tuning
(Appendix)

Yao Xiao Ziyi Tang Pengxu Wei Cong Liu Liang Lin
Sun Yat-sen University
{xiaoy99,tangzy27}@mail2.sysu.edu.cn {weipx3, liucong3}@mail.sysu.edu.cn
linliang@ieee.org

A. Experiment Details

A.1. Training Routines

For fine-tuning on ImageNet via vanilla fine-tuning or our approach, we use the AdamW optimizer [8] with $\beta_1 = 0.9, \beta_2 = 0.999$, weight decay of 0.1 and gradient clipping at $\ell_2$-norm 1. We use a batch size of 512, and fine-tune for 10 epochs. The learning rate is set to $3 \times 10^{-5}$ for all parameters and follows a cosine-annealing schedule [7] with 500 warm-up steps. For both training and testing, we resize and center-crop the images to the size of $224 \times 224$, and no data augmentation is applied. Besides, different from WiSE-FT [16], we do not use label smoothing.

A.2. Validation of CAM-based Object Masking

In Sec. 4.2, to verify that our CAM-based object masking can effectively mask the patches that cover the main object, we report the average object masking rate and IoU during training with different CAM score thresholds. Since we do not have the ground truth of the masks of main objects for ImageNet, we approximate it by the prediction of Mask2Former [2], a segmentation model pre-trained on COCO [6] (the specific model is reported in Appendix B). We select three super-classes defined in Restricted ImageNet [13] that can be recognized by the segmentation model, i.e., Dog, Cat and Bird, which cover 144 ImageNet classes in total. For each training image of these classes, we obtain the pixel-level segmentation mask $M_{seg}$ corresponding to the super-class, and compare it with our patch-level CAM-based mask, which is translated to a pixel-level mask $M_{CAM}$ according to the correspondence between patches and pixels.

The metrics in Tab. 2 in the main text are defined as follows. Formally, a mask $M$ of an image $I$ is defined as a subset of the pixels. Let $n(\cdot)$ denote the number of pixels in a mask or an image. Then, the metrics are defined as:

- Image masking rate: $\frac{n(M_{CAM})}{n(I)}$;
- Object masking rate: $\frac{n(M_{CAM} \cap M_{seg})}{n(M_{seg})}$;
- IoU: $\frac{n(M_{CAM} \cap M_{seg})}{n(M_{CAM} \cup M_{seg})}$.

A.3. WiSE-KD

In Sec. 4.4, we consider using the WiSE-FT [16] model as a teacher model, and add the vanilla knowledge distillation loss [5] to our training objective, i.e.,

$$L = L_{CE}(g(f(x), y) + \gamma L_{KL}(g(f(x), g_{c}(f_{c}(x)))) + \beta L_{MSE}(\hat{f}_{cf}(x), f_{cf}(x))),$$

where $L_{KL}$ is the Kullback-Leibler divergence loss, and $f_{c}$ and $g_{c}$ are the encoder and classification head of the ensemble model produced by WiSE-FT, correspondingly. We set $\gamma = 1$, and use the WiSE-FT model with $\alpha = 0.5$. The temperature of the vanilla knowledge distillation is 10.

B. Use of Existing Assets

Datasets. In this paper, we utilize the following existing benchmark datasets without modification or repackaging:

- ObjectNet [1] (https://objectnet.dev/)
In our experiments, we select the hyper-parameters based on validation accuracy on ImageNet, and use the other datasets solely for robustness evaluation. For ObjectNet, we follow the official guidance to remove the red borders of the images before other preprocessing steps in evaluation.

**Code and pre-trained model weights.** The experiments in this paper are based on the code and pre-trained model weights provided by the following packages or GitHub repositories:

- Model Soup [15] (https://github.com/mlfoundations/model-soups/issues/1): we use the pre-trained weights of uniform soup provided by the authors in an issue.

**References**


