FreeKD: Knowledge Distillation via Semantic Frequency Prompt

Supplementary Material

A. Implementation Details

A.1. Object Detection

We train the student with our FreeKD loss $\mathcal{L}_{\text{FreeKD}}$, regression KD loss, and task loss for the object detection task. We set FreeKD loss weight to 1 and regression loss weight to 1 on *Faster RCNN* students. For other detection frameworks, we simply adjust the loss weight of FreeKD to keep a similar amount of loss value as *Faster RCNN*. Concretely, the loss weights μ of $\mathcal{L}_{\text{FreeKD}}$ in Eq.14 on *RetinaNet*, *FCOS*, and *RepPoints* are 5, 10, and 10.

A.2. Semantic Segmentation

For the segmentation task, we train the models with standard data augmentations including random flipping, random scaling in the range of $[0.5, 2]$, and a crop with size 512×512. The student is supervised by the FreeKD loss and task loss. Specifically, the loss weights μ of $\mathcal{L}_{\text{FreeKD}}$ on *PSPNet-R18* and *DeepLabV3-R18* are 5 and 5, respectively.

A.3. Distill on Segment Anything Model

Segment Anything Model (SAM) [23] is a large-scale segmentation model characterized by the prompt proposed by Meta. To conduct distillation experiments on the SAM, we build a training framework based on the official code base[3] and refer to the distillation pipeline in MobileSAM[4]. The pipeline is divided into two stages: first, distill the image encoder, and then fine-tune the mask decoder with the image encoder frozen. During FreeKD distillation, we utilize a full SA-1B dataset consisting of around 10 million images to train the student model SAM-ViT-Tiny. The goal is to distill the student image encoder, with the officially released SAM-ViT-H model serving as the teacher. The image encoder is distilled for $10,000$ steps with 1024×1024 input size, and then the mask decoder is fine-tuned for $10,000$ steps. To speed up the training process, we simplify the finetune mask decoder process appropriately (e.g., only one round of interaction), and other settings strictly follow the original SAM for reproduction. We run all the experiments on 8 A100 GPUs.

B. DWT meets Spatial-based Method

To investigate the feasibility of directly applying other spatial-based distillation methods to frequency domain distillation, we conduct experiments using CWD [37] and PKD [1] as examples. For CWD, we minimize the Kullback–Leibler (KL) divergence between the channel-wise probability map of the high-frequency bands in the teacher and student. To apply PKD to the frequency domain, we try to reduce the representation gap between teacher and student via normalizing the high-frequency features with Pearson correlation. The results are listed in Table 10. We find that transferring the distillation method from the spatial domain to the frequency domain further improves the accuracy of the student model. Meanwhile, we notice that the convergence speed and training speed of frequency domain distillation are both higher than spatial domain distillation. This is because neural networks initially learn low-frequency information and later focus on high-frequency context, which allows them to mimic the high-frequency components of the teacher model as a form of previewing.

Table 10. **The performance of spatial-based distillation methods via frequency domain on COCO val set.** Evaluate models with average precision (AP).

Method	Spatial Domain	Frequency Domain
T: RepPoints-X101	44.2	-
S: RepPoints-R50	38.6	-
CWD [37] ICCV21	41.8	42.0
PKD [1] NeurIPS22	42.0	42.1
FreeKD	-	**42.4**

C. Confusion Matrix with FreeKD on COCO

We compute the confusion matrix of our method and make a comparison with the vanilla student in Figure 6. We utilize RepPoints-R50 student and RepPoints-X101 teacher as an example. The normalized values on the diagonal of the confusion matrix represent the ratio of predictions that match the ground truth predictions. Our method achieves a higher ratio of matching in most cases (e.g. 20% improvement on the toaster), which further validates that our method could transfer more knowledge from the frequency domain.

D. More Visualizations

D.1. Visualization of Feature Maps

We visualize the features of student and teacher models in the first output (stride 4) of FPN in Figure 7. The models used to extract the feature are RepPoints-R50 (student) and RepPoints-X101 (teacher) trained on COCO dataset. Following DiffKD [20], we average the feature map along

[3] https://github.com/facebookresearch/segment-anything

[4] https://github.com/ChaoningZhang/MobileSAM

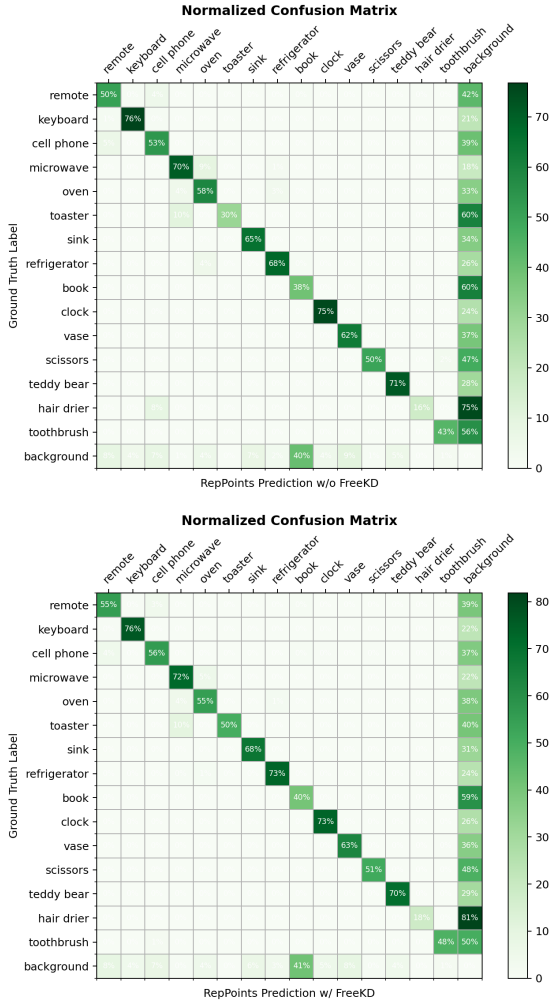

Figure 6. **Confusion matrix between the original student predictions and the student distilled by FreeKD predictions.**

the channel axis and perform softmax on the spatial axis to measure the saliency of each pixel. Formally, with a given feature map $\boldsymbol{X} \in \mathbb{R}^{C\times HW}$, we first average the channels and get $\boldsymbol{X'} \in \mathbb{R}^{HW}$, where

$$X'_i = \frac{1}{HW}\sum_{i=1}^{HW}(\boldsymbol{X}_{:,i}), \tag{15}$$

Then we generate the attention map for visualization as

$$\boldsymbol{V} = H \cdot W \cdot softmax(\boldsymbol{X'}/\tau), \tag{16}$$

where τ is the temperature factor for controlling the softness of distribution, and we set $\tau = 0.5$.

D.2. Visualization of Frequency PoIs

We visualize the more two PoIs (masks) generated by frequency prompt in the high-frequency band HH in Figure 8.

Figure 7. **Visualizations of the distilled student features and teacher features on COCO dataset.** We utilize RepPoints-R50 student and RepPoints-X101 teacher as an example.

Figure 8. **Visualizations of Frequency PoIs in high-frequency band `HH` on COCO dataset.** We employ the RepPoints-X101 teacher to generate the Frequency Prompt.