Supplementary Material

The implementation details are introduced first. Then we show more visualization results for the feature distributions shown in Fig. 4 and mean attention distances shown in Fig. 5. Finally, we present more experimental results.

A. Implementation Details

A.1. Pre-training on ImageNet

Data Augmentation. The data augmentations consist of random cropping (with a scale of 0.25-1.0), resizing to 224×224, random horizontal flip, gaussian blur, and color jittering. The local augmentations for the multi-crop strategy consist of random cropping (with a scale of 0.05-0.25), resizing to 96×96, random horizontal flip, gaussian blur, and color jittering. Two global views and eight local views are used in all pre-taining experiments.

Training. During the pre-training procedure, we follow the most hyper-parameters setting of DINO [5]. Without a specific statement, the default batch size is 256. The SGD and AdamW [35] optimizers are used for ResNet and ViT, respectively. The learning rate is linearly warmed up to its base value during the first 10 epochs. And the base learning rate is set to 0.1 and 0.0003 for ResNet and ViT, respectively. After the warm-up procedure, the learning rate is decayed with a cosine schedule [34]. The weight decay is set to $1e-4$ and 0.04 for ResNet and ViT, respectively. For the temperatures, τ is set to 0.1, and a linear warm-up from 0.04 to 0.07 is set to τ' during the first 30 epochs. Following DINO [5], the centering operation is applied to the output of the momentum encoders to avoid collapse. λ_1 and λ_2 in Eq. (9) are set to 1 and 0.1 for the larger and smaller models, respectively. While for model pairs with the same backbone, λ_1 and λ_2 are set to 1.

A.2. k-NN and Linear Probing on ImageNet

After pre-training, the k-NN and linear probing are employed to evaluate the representation performance. For k-NN, it is implemented based on DINO [5]. We report the best result among $k = 10, 20, 100, 200$.

For linear probing, a linear classifier added to the frozen backbone is trained [21]. The linear classifier is trained with the SGD optimizer and a batch size of 2048 for 100 epochs on the ImageNet training set. The learning rate is linearly warmed up to its base value during the first 10 epochs. And the base learning rate is set to 0.08 and 0.008 for ResNet and ViT, respectively. After the warm-up stage, the learning rate is decayed with a cosine schedule [34]. Weight decay is not used. The input resolution is 224×224 during training and testing.

A.3. Semi-Supervised Learning on ImageNet

We use the 1% and 10% subsets of the ImageNet [41] training set for fine-tuning, which follows the semi-supervised protocol in [8]. The same splits of 1% and 10% of ImageNet training set in [9] are used. Models are fine-tuned with 1024 batch size for 60 epochs and 30 epochs on 1% and 10% subsets, respectively. The SGD optimizer is adopted. The learning rate is linearly warmed up to its base value during the first 5 epochs. And the best base learning rate is searched for each model. After the warm-up procedure, the learning rate is decayed with a cosine schedule [34]. The input resolution is 224×224 during training and testing.

A.4. Fine-tuning on Cifar10/Cifar100

Most settings keep the same with the experiments of semi-supervised learning on ImageNet except for the training epochs. On Cifar10/100, the warming-up epoch is set to 10, and the training epoch is set to 100.

A.5. Fine-tuning on COCO

Following [15], The C4-based Mask R-CNN [22] detector is used for objection detection and instance segmentation on COCO. The model is trained for 180k iterations, i.e., the $2\times$ schedule. The SGD optimizer is adopted. The initial learning rate is set to 0.02. The scale of images for training is set as [600, 800] and 800 at inference. We report $\mathrm{AP^b}$, $\mathrm{AP^b_{50}}$, and $\mathrm{AP^b_{75}}$ for object detection and $\mathrm{AP^s}$, $\mathrm{AP^s_{50}}$, and $\mathrm{AP^s_{75}}$ for instance segmentation.

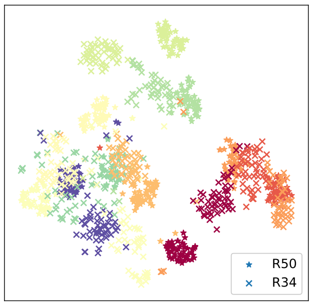
B. More Visualizations

B.1. T-SNE Visualization of Feature Distribution

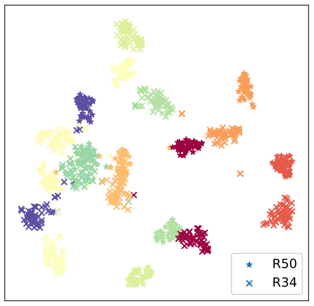
Following Fig. 4, more model pairs, including two ResNets and two ViTs are visualized. As shown in Fig. S1, the model pairs (R50-R34 and ViT-S-ViT-T) trained by MOKD also show different feature distributions, demonstrating that MOKD does not make models more similar. In addition, we can see that the feature distributions of R34 and ViT-T get better when trained with MOKD.

B.2. Mean Attention Distances

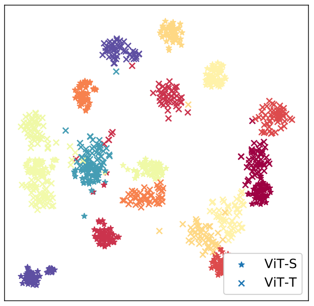
In Fig. 5, we show the mean attention distances [13] of ViT-S, R50, and R101 trained independently (by DINO) and trained by MOKD. For ResNets, when trained with ViTs, i.e., R50-ViT-S, R5O-ViT-B, R101-ViT-S, and R101-ViT-B, the ResNet models trained by MOKD turns to be more "global". However, this phenomenon is not shown in two ResNet model pairs trained by MOKD, i.e., R50-R34, R50-R18, R101-R34, and R101-R18. As shown in Fig. S2, we show more mean attention distances of ViTs. Compared with the ViT models trained independently (as shown in Fig. S2(a)(e)), the mean attention distances on deep layers



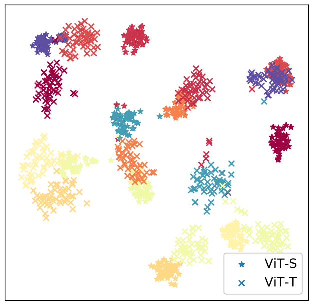
(a) Independent (R50-R34).



(b) MOKD (R50-R34).

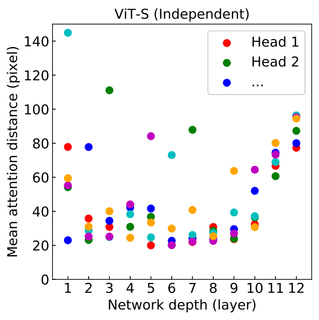


(c) Independent (ViT-S-ViT-T).

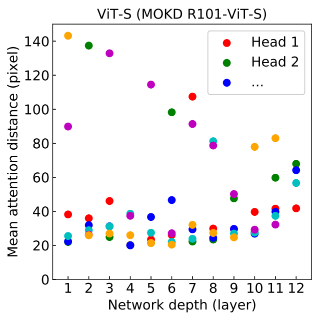


(d) MOKD (ViT-S-ViT-T).

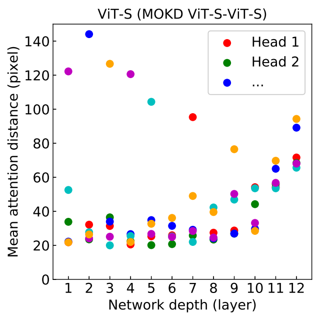
Figure S1. T-SNE [46] visualization of feature distributions on ImageNet100. Ten categories (shown in different colors) are randomly selected for better visualization. Different color denotes different category and different marker denotes different model.



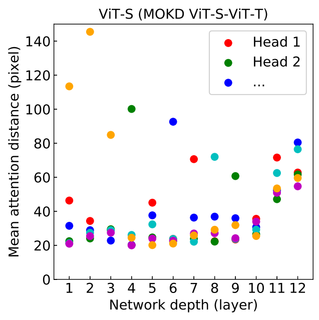
(a) ViT-S (Independent).



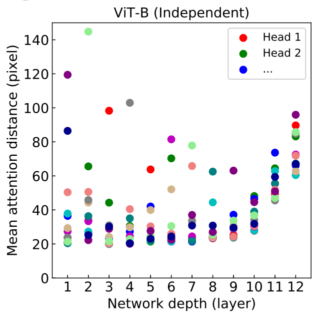
(b) ViT-S (MOKD with R101).



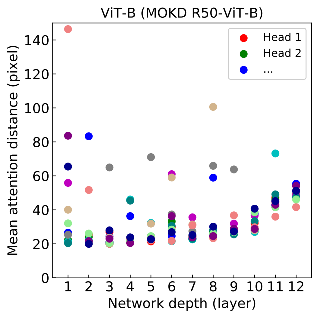
(c) ViT-S (MOKD with ViT-S).



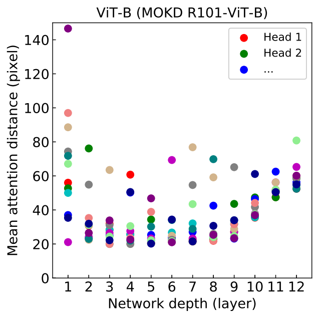
(d) ViT-S (MOKD with ViT-T).



(e) ViT-B (Independent).



(f) ViT-B (MOKD with R50).



(g) ViT-B (MOKD with R101).

Figure S2. Mean attention distances [13] of different models.

(layer 10-12) of the ViT models trained with ResNet models (as shown in Fig. S2(b)(f)(g)) decrease, which indicates that the ViT-S model trained by MOKD turns to be more "local" on deep layers. This phenomenon is not shown in two ViT models trained by MOKD, as shown in Fig. S2(c)(d). These observations show that two heterogeneous models absorb knowledge from each other: ViT model learns more locality while CNN model learns more global information.

C. More Experiments

C.1. Results on Other Convnets

We conduct experiments on EfficientNet-B0 [42] and MobileNet-v3-Large [26]. Following DisCo [16], R50 and R101 are selected as the larger models. We pre-train MOKD with 256 batch size for 100 epochs on ImageNet. As shown in Tab. S1, MOKD brings consistent improvements over different model pairs. It achieves the best performance for EfficientNet-B0 [42] and MobileNet-v3-Large [26].

C.2. Influence of T-Head Depth

T-Head is added for cross-attention feature search in MOKD. As shown in Tab. S2, increasing T-Head depth (number of transformer blocks) improves the performance of MOKD. However, extra computation costs brought by T-Head should be controlled. Thus, T-Head depth should be small and set to 3 in this study.

Method	R50	Eff-b0	R50	Mob-v3	R101	Eff-b0	R101	Mob-v3
SEED [15]	67.4	61.3	67.4	55.2	70.3	63.0	70.3	59.9
ReKD [58]	67.6	63.4	67.6	56.7	69.7	65.0	69.7	59.6
DisCo [16]	67.4	66.5	67.4	64.4	69.1	68.9	69.1	65.7
MOKD	72.5	**69.2**	72.2	**66.0**	74.9	**70.1**	74.7	**67.2**

Table S1. Results on other convnets.

Depth	1	2	3
R50, ViT-S	87.4, 83.3	88.1, 84.1	88.3, 84.6

Table S2. Influence of T-Head depth.

Method	Backbones	Time (h)	Mem.(G)	LP
DINO [5]	R50/ViT-S	97+94	6.4+6.3	72.1/73.8
MOKD	R50-ViT-S	122	13.5	74.1/74.4

Table S3. Training Time and Memory Requirement.

C.3. Training Time and Memory Requirement

We show the total training time and peak memory per GPU (“mem.”) when training ViT-S model pairs on an 8 V100 GPU machine. We pre-train DINO [5] and MOKD with 256 batch size for 100 epochs on ImageNet. From Tab. S3, we can tell that the total memory requirement and training time of training two models independently via DINO [5] are comparable to those of MOKD.