Closest Neighbors are Harmful for Lightweight Masked Auto-encoders

Supplementary Material

7. Repelling Random Patches vs. Repelling Closest Neighbor Patches

We are fully aware of the trend of employing randomly asymmetrical masks to the input of the MAE training [18, 39]. We further evaluate the impact of incorporating the random asymmetry while applying the proposed repelling learning scheme (Eq. 2 and Eq. 3). Specifically, the inputs are two sets of randomly selected patches with the same sparsity, we compute the same repelling loss based on Eq. 2 and Eq. 3 after the default encoding and decoding.

Table 8. Performance comparison between random asymmetrical masks and the proposed NoR-MAE.

ViT Model	Method	ImageNet-100 Accuracy (%)
ViT-Tiny	Random Asymmetry	72.15
	Neighbor Repelling (Proposed)	**78.26**
ViT-Small	Random Asymmetry	82.04
	Neighbor Repelling (Proposed)	**84.28**

As shown in Table 8, repelling the random information cannot fully resolve the poor learnability of the lightweight model. The superior accuracy of the proposed Neighbor-Repelling scheme further proves the necessity of understanding the semantic differences between different local information (neighbor-based centroid).

8. Minimize the Distance Between Patches and their Closest Neighbors

Although the proposed method highlights the necessity of repelling the closest neighbor, the similarity between patches and their closest neighbors should be ignored. Based on the setup of the closest neighborhood patches (CNP) introduced in Section 3.1, we perform the MAE training scheme by minimizing the distance between the unmasked patches and their CNP. Mathematically, we modify Eq. 2 to Neighbor Alignment Loss (NoA):

$$\mathcal{L}_{\text{NoA}} = \frac{1}{NDK} \sum_{N,D,K} (\bar{Z} - \bar{Z}_{\text{CNP}})^2 \tag{4}$$

And the total loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_2(\bar{Z}, \text{Ground Truth}) + \lambda_{\text{NoA}} \cdot \mathcal{L}_{\text{NoA}} \tag{5}$$

Which is equivalent to minimizing the Mean Square distance between patches and their CNP. As shown in Table 9, the performance boost with Neighbor Alignment (NoA) is unsatisfactory compared to the proposed Neighbor Repelling. Given that only 20% of the patches are extremely similar (similarity ≥ 0.99) to their closest neighbor patches (CNP) (Figure 3), minimizing the distance between each patch and their CNP ignores the dissimilarity of the CNP, which leads to the sub-optimal performance.

Table 9. Comparison between Neighbor Repelling and Neighbor Alignment for MAE training.

ViT Model	Method	ImageNet-100 Accuracy (%)
ViT-Tiny	MAE [20]	71.04
	Neighbor Alignment (NoA)	72.52
	Neighbor Repelling (NoR)	**78.26**

Table 10. Performance and training cost comparison between NoR-MAE and the knowledge distillation-based MAE training (DMAE) with the ImageNet-1K dataset (based on RTX 6000 GPU).

ViT Model	Method	Teacher	Total GPU Memory (GB)	Time per Epoch	FT Acc. (%)
ViT-Tiny	MAE [20]	N/A	102 (1×)	35 min (4 GPU)	66.60
	DMAE [3]	ViT-B	152 (1.49 ×)	43 min (4 GPU)	70.00
	NoR-MAE (This work)	**N/A**	108 (1.08×)	36 min (4 GPU)	**70.24**
ViT-Small	MAE [20]	N/A	144 (1 ×)	45 min (4 GPU)	79.00
	DMAE [3]	ViT-B	218 (1.5×)	1 hr 2 min (4 GPU)	79.30
	NoR-MAE (This work)	**N/A**	152 (1.06×)	47 min (4 GPU)	**80.13**

9. Training Cost of NoR-MAE

We profiled and compare the training cost of the vanilla MAE, distillation-based DMAE, and the proposed NoR-MAE algorithm. As shown in Table 10, the proposed method outperforms DMAE [3] with better accuracy and $1.4\times$ training cost reduction (memory and training time). Compared to the vanilla MAE [20], the proposed NoR-MAE algorithm improve the performance of the lightweight vision transformer with **only $\leq$8% memory and training time overhead**.

Table 11. Training cost comparison between the proposed NoR-MAE and vanilla MAE on a cloud GPU server.

ViT-Tiny	Batch Size	Total GPU Memory	Time / Epoch	Cost / Epoch
MAE (75% Spars.)	4096	102 GB	25 min 20 sec (1.0×)	2.69 USD
MAE (50% Spars.)	4096	127 GB	33 min 07 sec (1.30×)	3.50 USD
NoR-MAE (75% Spars.)	4096	108 GB	27 min 10 sec (1.08×)	2.90 USD

To solidify resource usage, **we report the model on a leased cloud server** with 2×H100 GPUs ($6.38/hour), as shown in Table 11 above.

Figure 9. Improved segmentation quality with the proposed NoR-MAE algorithm.

10. NoR-MAE for Segmentation

In addition to the quantitative segmentation results reported in Section 4, we validate the downstream performance of the proposed NoR-MAE with qualitative demonstration with the ADE20K dataset. We follow the default settings of MAE [20] and employ the UperNet [35] and fine-tune the NoR-MAE-pretrained ViT-Small model for 100 epochs.

Compared to the vanilla MAE [20], the proposed NoR-MAE exhibits clearer segmentation among multiple objects within the given scene, as shown in Figure 9.

11. Experimental Setup

Pretraining on the ImageNet dataset We follow the standard pre-training protocal from the vanilla MAE to initate the pre-training of the proposed NoR-MAE on ImageNet-1K and ImageNet-100 datasets. Specifically, we choose the default base learning rate as 1.5e-4 with weight decay = 0.05. The vision transformer models are trained by 200 and 400 epochs (include 40 epochs of warmup) with batch size of 4096.

End-to-end Fine-tuning The end-to-end fine-tuning follows the standard supervised learning of supervised ViT training. We use the base learning rate as 1e-3, with the weight decay of 0.05. Following the setup of the vanilla MAE [20], we employ label smoothing, mixup, and cutmix for supervised fine-tuning.

Downstream Fine-tuning We follow the fine-tuning settings in [13] and fine-tune the pre-trained model for 10,000 steps with SGD and batch size of 64. The learning rate is set to 0.1 with no weight decay. The input samples are resized to 224×224 to maintain the dimensionality as the pre-trained model.