

# EXTREME MASKING FOR LEARNING INSTANCE AND DISTRIBUTED VISUAL REPRESENTATIONS

**Anonymous authors**

Paper under double-blind review

## A DISCUSSIONS ON CAiT-STYLE ARCHITECTURE

ExtreMA follows the CaiT-style transformer architecture Touvron et al. (2021), where the class token is appended later in the attention blocks. We find that such design is critical for ExtreMA to stabilize learning, whereas the conventional ViT class token design failed to converge properly. Additionally, we also investigate a third option on using average pooling across tokens to aggregate the holistic representation. In Figure 1, we plot the training loss and the kNN classification accuracy for different class token designs. The ViT class token design leads to unstable optimization, and average pooling finds a representation shortcut. The CaiT-style architecture works as desired.

It remains as a limitation of this work to fully understand the training dynamics for the class token design. We hypothesize that the problem originates from the Siamese networks processing input sequences with very different lengths. This makes the learning of the class token representation harder, when it is processed throughout the network.

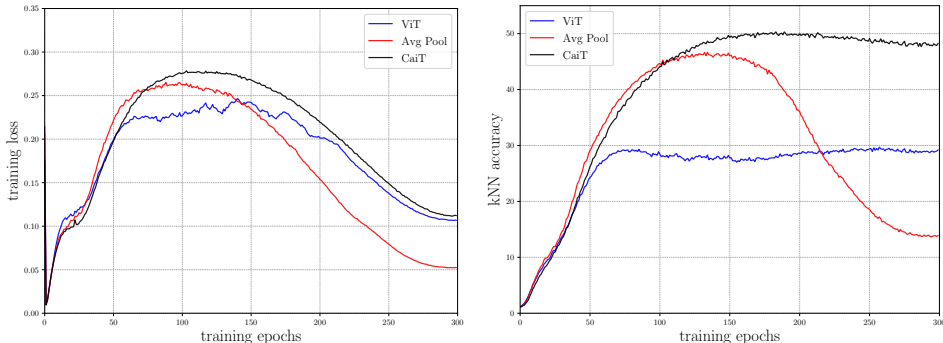


Figure 1: Training loss and kNN accuracy curves for three class token designs: ViT, average pooling, and CaiT. The network is trained with the ExtreMA objective with masking ratio 90% of 8 crops, using the ViT-small architecture.

## B DETECTION TRANSFER RESULTS

We adopt the Mask-RCNN framework for object detection and instance segmentation using the ViT-base architecture. We fine-tune the model on MSCOCO for 12 epochs and evaluate the performance on the validation set. The results are summarized in Table 1. ExtreMA outperforms DINO/MoCo-v3/BEiT while using a lot less compute. ExtreMA outperforms MAE with the same number of pretraining epochs, but underforms MAE if MAE is trained longer. When pretrained on the ImageNet22k dataset, ExtreMA improves the performance by about 1% AP.

## C ADDITIONAL COMPARISONS OF LOCALITY PROPERTIES

We compare the performance on localization with other works, MAE / DINO / MoCo-v3. We use the [cls] token representation from these models. In Figure 2, we find that DINO performs favorably well, and that MAE / MoCo-v3 degrades the performance notably. MAE does not supervise an instance

Table 1: Object detection and instance segmentation transfer on COCO.

methods	epochs	object detection			instance segmentation		
		AP	AP <sub>50</sub>	AP <sub>75</sub>	AP	AP <sub>50</sub>	AP <sub>75</sub>
DINO	300	46.8	68.6	50.9	41.5	65.3	44.5
MoCo-v3	300	45.5	67.1	49.4	40.5	63.7	43.4
BEiT	800	42.1	63.3	46.0	37.8	60.1	40.6
MAE	300	45.4	66.4	49.6	40.6	63.4	43.7
MAE	1600	48.4	69.4	53.1	42.6	66.1	45.9
ExtreMA (1k)	300	47.5	68.9	51.9	42.0	65.6	65.1
ExtreMA (22k)	30	<b>48.5</b>	<b>69.8</b>	<b>53.1</b>	<b>42.7</b>	<b>66.5</b>	<b>46.0</b>

representation in the formulation, and hence its instance representation is weaker. MoCo-v3 suffers from the heavy use of spatial cropping augmentation, and DINO improves by using small local crops for localization.

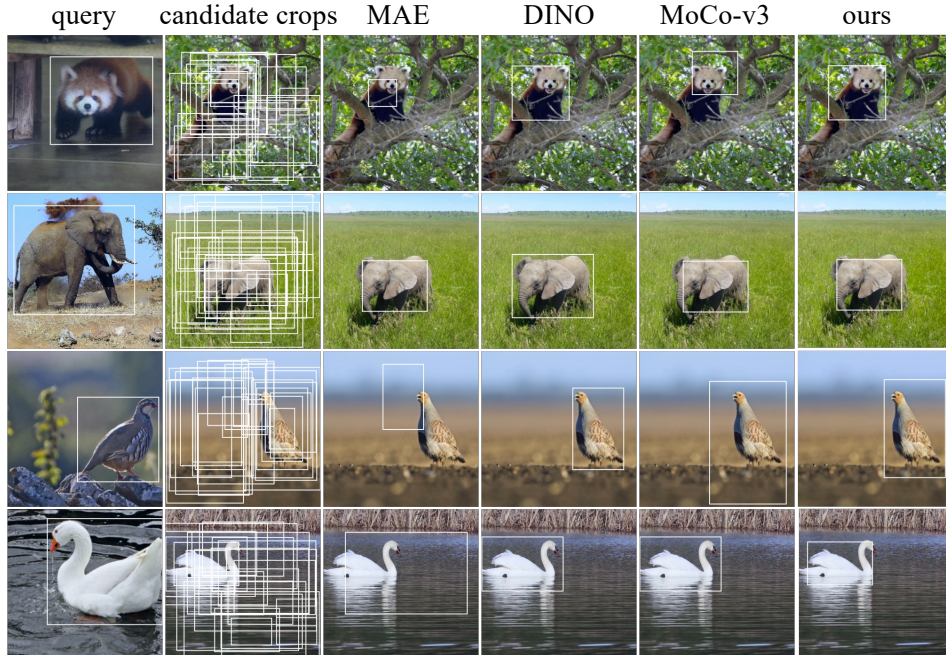


Figure 2: State-of-the-art comparisons with other models for localization.

## D DETAILS OF EVALUATION PROTOCOLS

The evaluation protocols for end-to-end finetuning and linear probing largely follow BEiT and MAE. The hyper-parameter configurations are detailed in Table 2 and Table 3. We finetune ViT-Small models for 200 epochs and ViT-Base models for 100 epochs. We use a base learning rate 1e-3 and layer decay 0.75 for ImageNet1k pretrained models, and a slightly smaller learning rate 5e-4 and a smaller layer decay 0.65 for ImageNet22k pretrained models. The linear probing configuration is adopted consistently for all reported entries.

Table 2: End-to-end fine-tuning protocol.

config	value
optimizer	AdamW
base learning rate	1e-3
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2=0.9, 0.999$
layer-wise lr decay	0.75
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	200 (S), 100 (B)
augmentation	RandAug (9, 0.5)
label smoothing	0.1
mixup	0.8
cutmix	1.0
drop path	0.1

Table 3: Linear probing protocol.

config	value
optimizer	LARS
base learning rate	0.1
weight decay	0
optimizer momentum	0.9
batch size	4096
learning rate schedule	cosine decay
warmup epochs	10
training epochs	90
augmentation	RandomResizedCrop

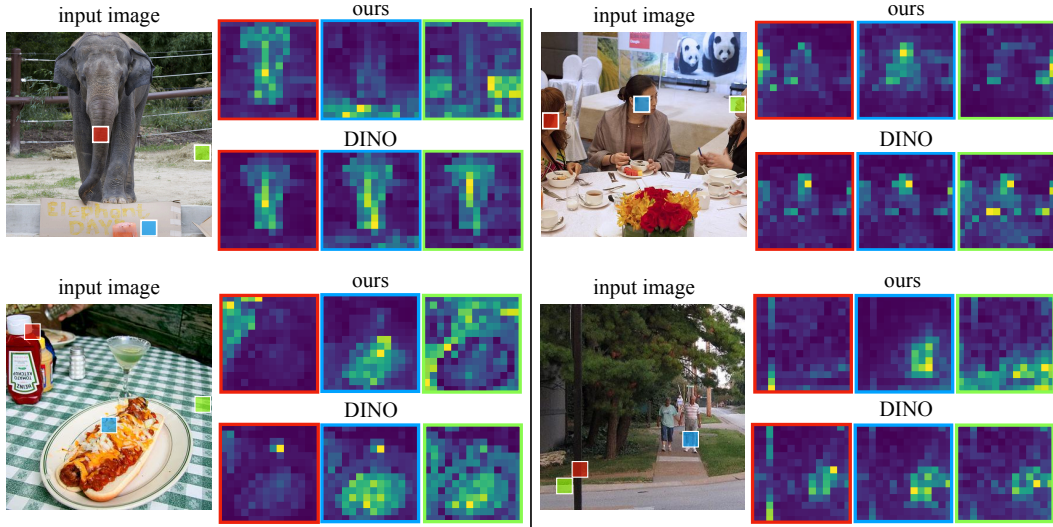


Figure 3: **Attention maps on the last layer of the ViT encoder.** We average the responses for 12 attention heads for visualization. Our model produces diverse and distributed attention maps, whereas DINO Caron et al. (2020) mainly attends to the foreground object, ignoring the others. The border color of the attention map corresponds to the colored query in the input image.

## E ADDITIONAL VISUALIZATIONS

We provide additional visualizations on the generative aspects of our model in Fig. 4, and attention maps of the distributed representations in Fig. 3. Both visualizations reveal properties of the distributed representations. These representations maintain accurate correspondences with the input tokens, while inferring meaningful semantic relationships among tokens.

## REFERENCES

- Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.
- Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021.



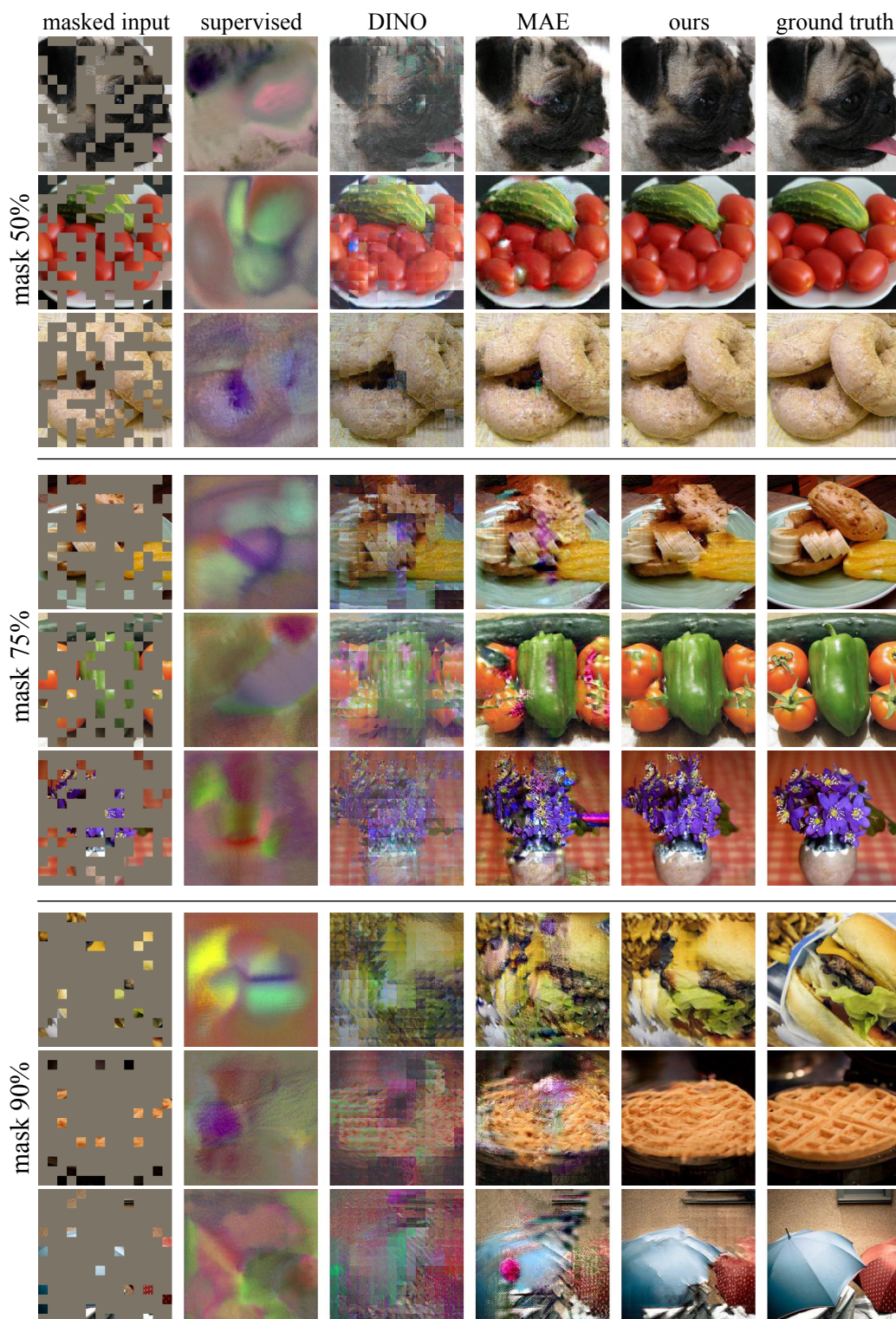


Figure 4: Additional examples of inpainting at various masking ratios.