

LSDL Lecture 05

Masked Image Modeling

Ildus Sadrdinov, 14.10.24

Self-supervised pre-training

generative

generative
pre-text tasks

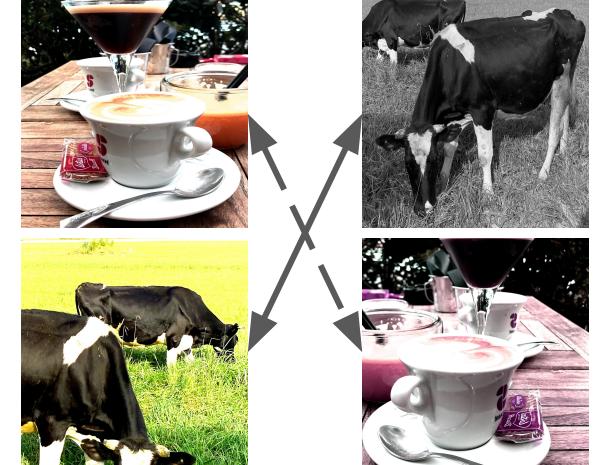


discriminative

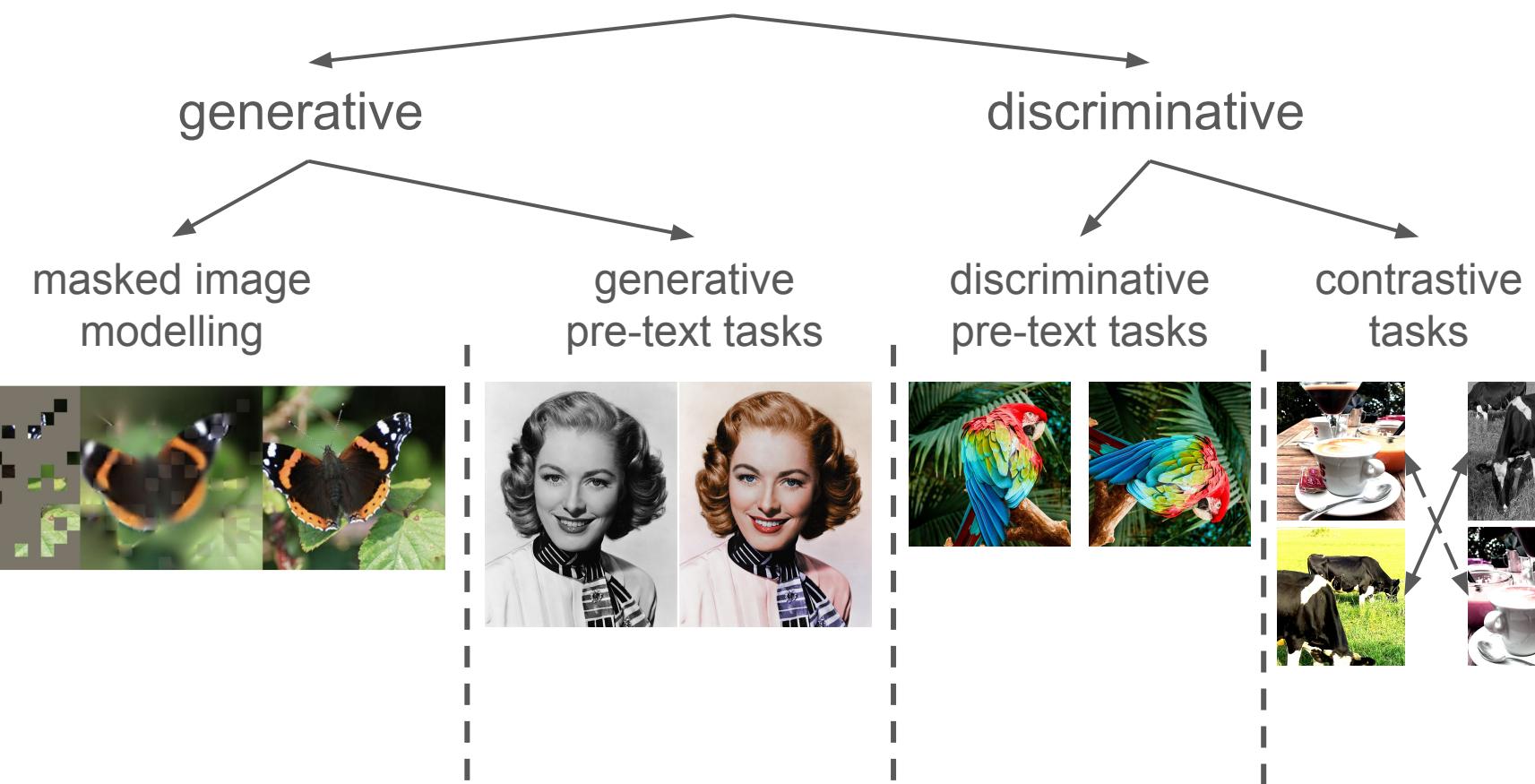
discriminative
pre-text tasks



contrastive
tasks



Self-supervised pre-training



Plan

- **Self-supervised transformers**
 - Self-distillation with no labels (DINO)
- **Masked Image Modeling**
 - BEiT, MAE
 - MaskFeat
- **Improving Contrastive Learning**
 - Dense CL
 - Supervised CL
 - Augmentations in CL (DiLo, LooC)

Plan

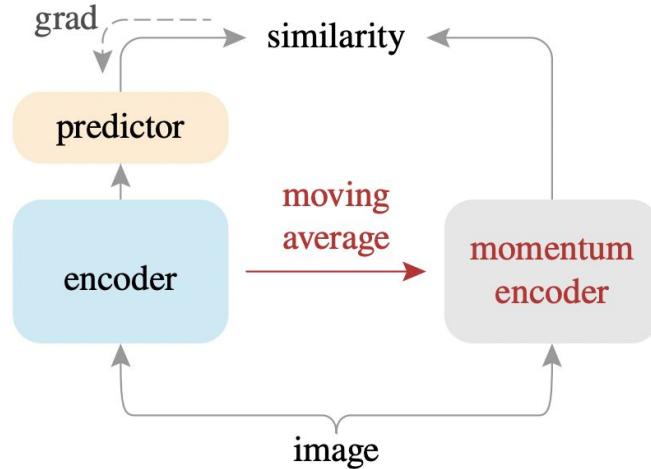
- **Self-supervised transformers**
 - Self-distillation with no labels (DINO)
- **Masked Image Modeling**
 - BEiT, MAE
 - MaskFeat
- **Improving Contrastive Learning**
 - Dense CL
 - Supervised CL
 - Augmentations in CL (DiLo, LooC)

Emerging Properties in Self-Supervised Vision Transformers

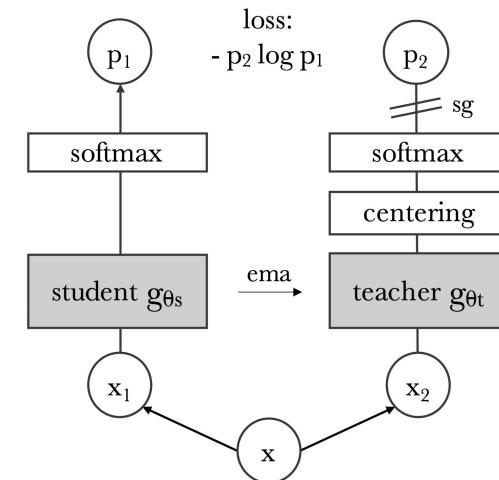
FAIR

- Self-supervised ViT features contain explicit information about **semantic segmentation** (opposed to supervised ViTs and ConvNets)
- Self-supervised ViT features are **excellent k-NN classifiers**
- Importance of **momentum encoder, multi-crop training, small patches**
- **DINO** – self-DIstillation with **NO** labels (contrastive w/o negative examples)

DINO: scheme



BYOL



DINO

$$\mathcal{L} = - \sum_{k=1}^K p_k^t \log p_k^s$$

[Caron et al., 2021](#)

DINO: avoiding collapse

centering

$$g_{\theta_t}(x) \leftarrow g_{\theta_t}(x) - c$$

$$c \leftarrow mc + (1 - m) \frac{1}{B} \sum_{i=1}^B g_{\theta_t}(x_i)$$

sharpening centering both

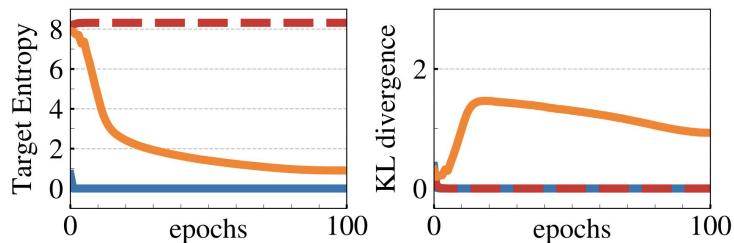


Figure 7: **Collapse study.** (left): evolution of the teacher's target entropy along training epochs; (right): evolution of KL divergence between teacher and student outputs.

sharpening

$$p_t(x)^{(i)} = \frac{\exp(g_{\theta_t}(x)^{(i)}/\tau_t)}{\sum_{k=1}^K \exp(g_{\theta_t}(x)^{(k)}/\tau_t)}, 0 < \tau_t < 1$$

$$H(P_t, P_s) = h(P_t) + D_{KL}(P_t | P_s)$$

DINO: algorithm

Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

DINO: multi-crop (proposed in SwAV)

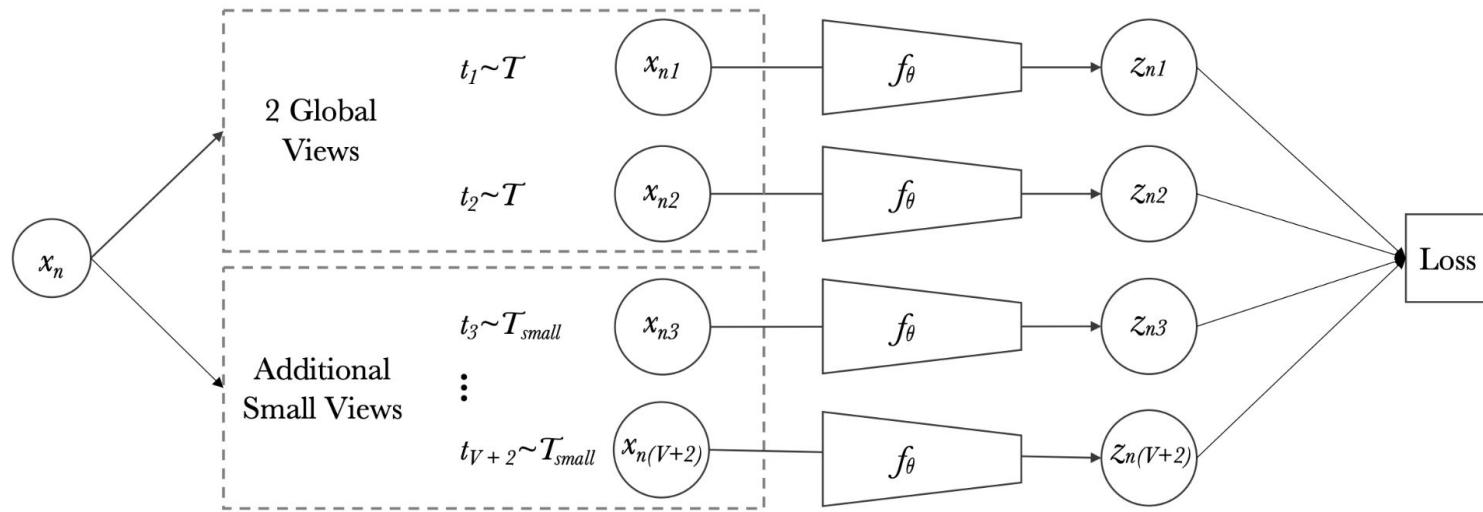


Figure 5: **Multi-crop:** the image x_n is transformed into $V + 2$ views: two global views and V small resolution zoomed views.

DINO: comparison to contrastive methods (300ep, ViT-S/16)

	Method	Mom.	SK	MC	Loss	Pred.	k -NN	Lin.
1	DINO	✓	✗	✓	CE	✗	72.8	76.1
2		✗	✗	✓	CE	✗	0.1	0.1
3		✓	✓	✓	CE	✗	72.2	76.0
4		✓	✗	✗	CE	✗	67.9	72.5
5		✓	✗	✓	MSE	✗	52.6	62.4
6		✓	✗	✓	CE	✓	71.8	75.6
7	BYOL	✓	✗	✗	MSE	✓	66.6	71.4
8	MoCov2	✓	✗	✗	INCE	✗	62.0	71.6
9	SwAV	✗	✓	✓	CE	✗	64.7	71.8

SK: Sinkhorn-Knopp, MC: Multi-Crop, Pred.: Predictor

CE: Cross-Entropy, MSE: Mean Square Error, INCE: InfoNCE

DINO: results

Method	Arch.	Param.	im/s	Linear	k -NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	75.3	65.7
DINO	RN50	23	1237	75.3	67.5
Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [30]	ViT-S	21	1007	71.4	66.6
MoCov2* [15]	ViT-S	21	1007	72.7	64.4
SwAV* [10]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	77.0	74.5

Method	Arch.	Param.	im/s	Linear	k -NN
<i>Comparison across architectures</i>					
SCLR [12]	RN50w4	375	117	76.8	69.3
SwAV [10]	RN50w2	93	384	77.3	67.3
BYOL [30]	RN50w2	93	384	77.4	–
DINO	ViT-B/16	85	312	78.2	76.1
SwAV [10]	RN50w5	586	76	78.5	67.1
BYOL [30]	RN50w4	375	117	78.6	–
BYOL [30]	RN200w2	250	123	79.6	73.9
DINO	ViT-S/8	21	180	79.7	78.3
SCLRv2 [13]	RN152w3+SK	794	46	79.8	73.1
DINO	ViT-B/8	85	63	80.1	77.4

DINO: kNN for image retrieval & copy detection

Table 3: **Image retrieval.** We compare the performance in retrieval of off-the-shelf features pretrained with supervision or with DINO on ImageNet and Google Landmarks v2 (GLDv2) dataset. We report mAP on revisited Oxford and Paris. Pretraining with DINO on a landmark dataset performs particularly well. For reference, we also report the best retrieval method with off-the-shelf features [57].

Pretrain	Arch.	Pretrain	\mathcal{R}_{Ox}		\mathcal{R}_{Par}	
			M	H	M	H
Sup. [57]	RN101+R-MAC	ImNet	49.8	18.5	74.0	52.1
Sup.	ViT-S/16	ImNet	33.5	8.9	63.0	37.2
DINO	ResNet-50	ImNet	35.4	11.1	55.9	27.5
DINO	ViT-S/16	ImNet	41.8	13.7	63.1	34.4
DINO	ViT-S/16	GLDv2	51.5	24.3	75.3	51.6

Table 4: **Copy detection.** We report the mAP performance in copy detection on Copydays “strong” subset [21]. For reference, we also report the performance of the multigrain model [5], trained specifically for particular object retrieval.

Method	Arch.	Dim.	Resolution	mAP
Multigrain [5]	ResNet-50	2048	224^2	75.1
Multigrain [5]	ResNet-50	2048	largest side 800	82.5
Supervised [69]	ViT-B/16	1536	224^2	76.4
DINO	ViT-B/16	1536	224^2	81.7
DINO	ViT-B/8	1536	320^2	85.5

DINO: attention maps to segmentation masks

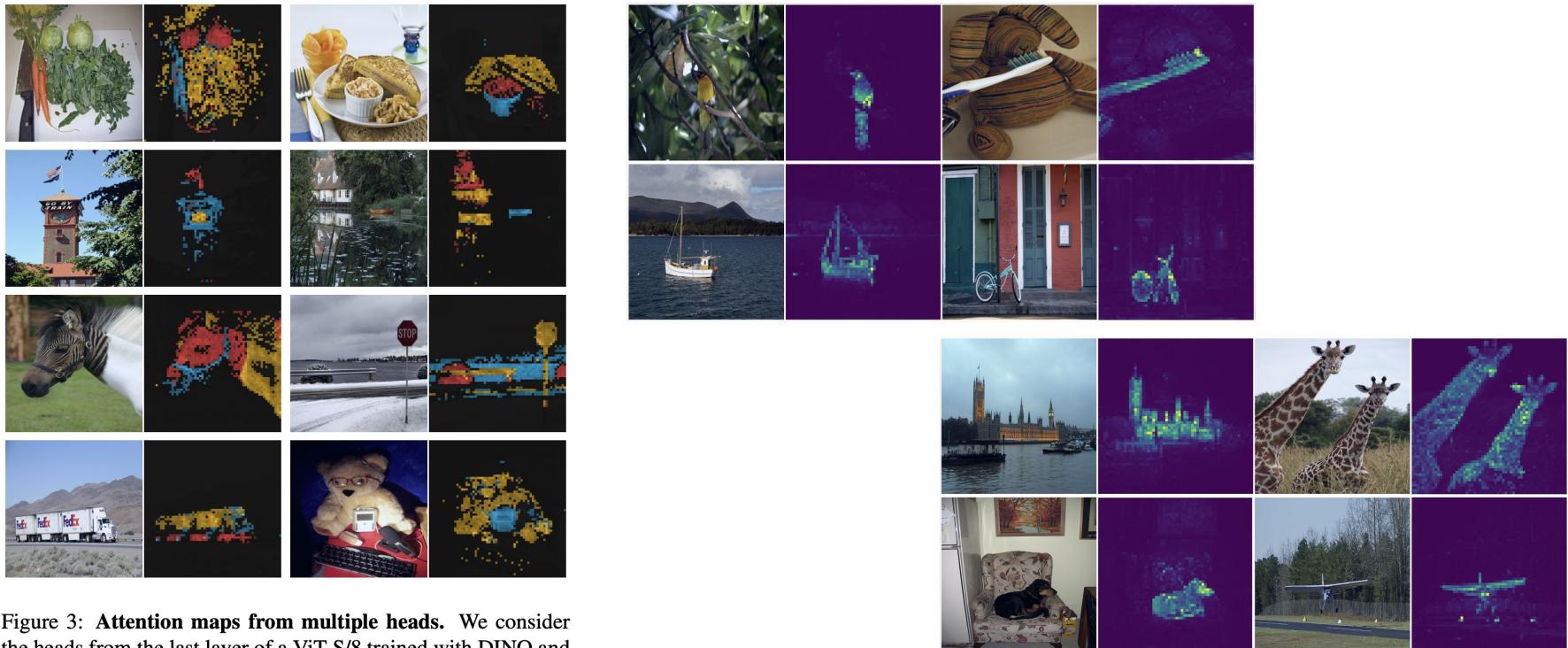


Figure 3: **Attention maps from multiple heads.** We consider the heads from the last layer of a ViT-S/8 trained with DINO and display the self-attention for [CLS] token query. Different heads, materialized by different colors, focus on different locations that represents different objects or parts (more examples in Appendix).

DINO: attention maps to segmentation masks

	Random	Supervised	DINO
ViT-S/16	22.0	27.3	45.9
ViT-S/8	21.8	23.7	44.7

Supervised



DINO



Figure 4: Segmentations from supervised versus DINO. We visualize masks obtained by thresholding the self-attention maps to keep 60% of the mass. On top, we show the resulting masks for a ViT-S/8 trained with supervision and DINO. We show the best head for both models. The table at the bottom compares the Jaccard similarity between the ground truth and these masks on the validation images of PASCAL VOC12 dataset.

DINO: ablations

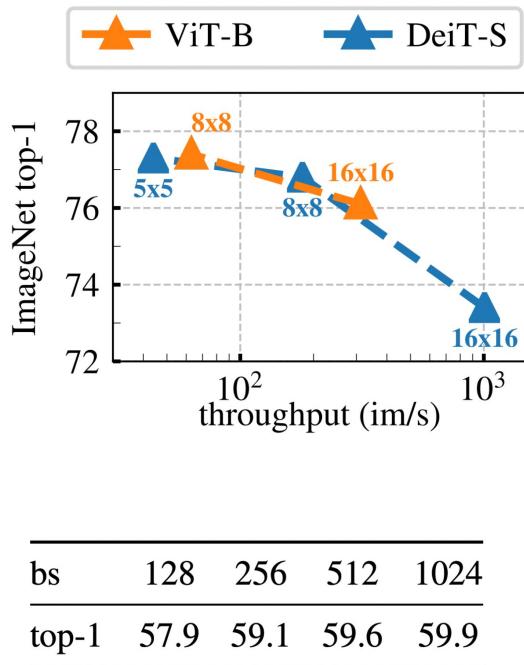


Figure 5: **Effect of Patch Size.** k -NN evaluation as a function of the throughputs for different input patch sizes with ViT-B and ViT-S. Models are trained for 300 epochs.

Table 8: **Time and memory requirements.** We show total running time and peak memory per GPU (“mem.”) when running ViT-S/16 DINO models on two 8-GPU machines. We report top-1 ImageNet val acc with linear evaluation for several variants of multi-crop, each having a different level of compute requirement.

multi-crop	100 epochs		300 epochs		
	top-1	time	top-1	time	mem.
2×224^2	67.8	15.3h	72.5	45.9h	9.3G
$2 \times 224^2 + 2 \times 96^2$	71.5	17.0h	74.5	51.0h	10.5G
$2 \times 224^2 + 6 \times 96^2$	73.8	20.3h	75.9	60.9h	12.9G
$2 \times 224^2 + 10 \times 96^2$	74.6	24.2h	76.1	72.6h	15.4G

Table 9: **Effect of batch sizes.** Top-1 with k -NN for models trained for 100 epochs without multi-crop.

Plan

- **Transformers for images**
 - ViT, DEiT
 - Self-distillation with no labels (DINO)
- **Masked Image Modeling**
 - BEiT, MAE
 - MaskFeat
- **Improving Contrastive Learning**
 - Dense CL
 - Supervised CL
 - Augmentations in CL (DiLo, LooC)

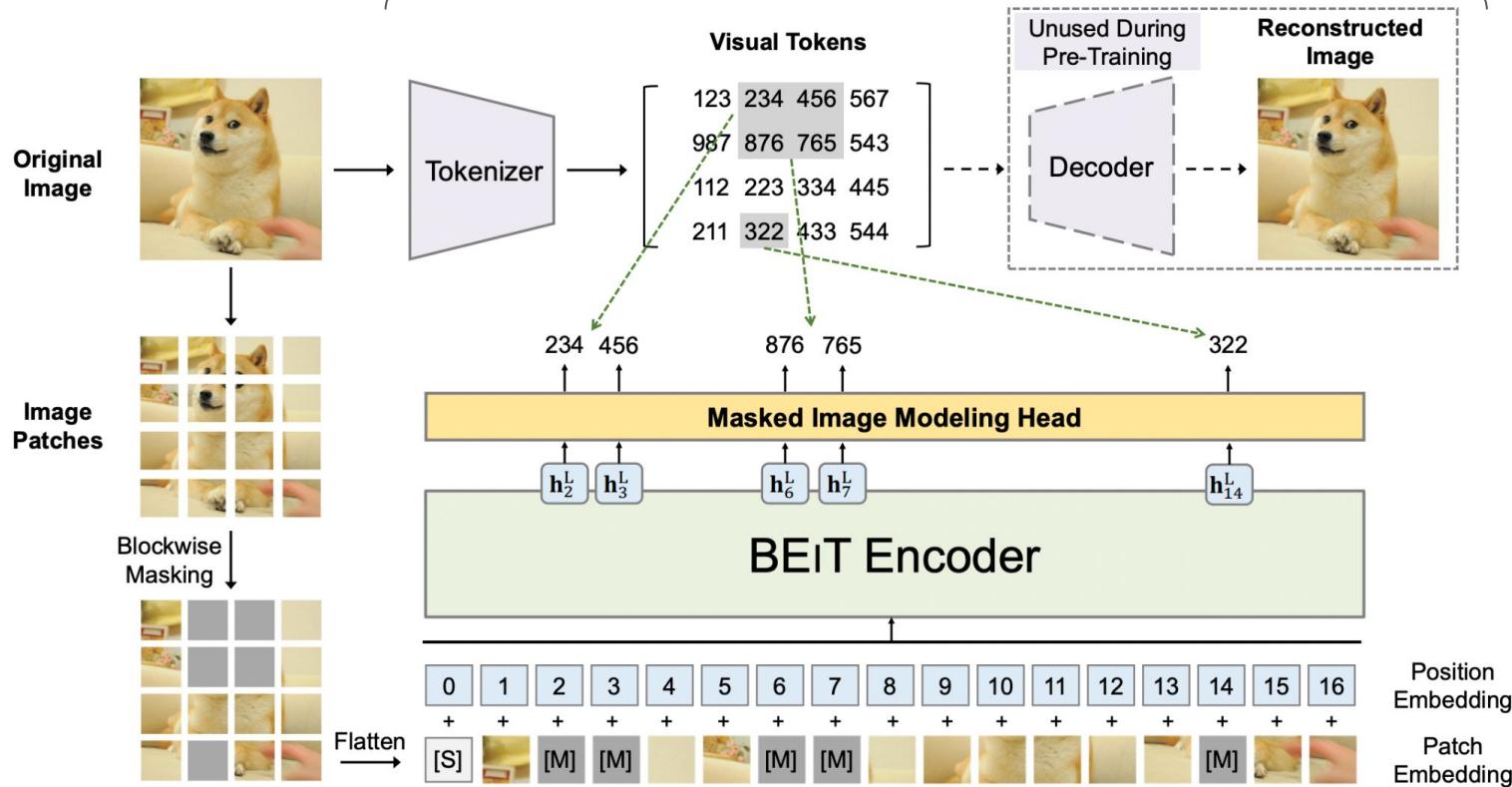
BEiT

Bidirectional Encoder representation from Image Transformers
Microsoft Research

- Masked Image Modeling (MIM) task
- Masking mechanism is similar to BERT
- Image is tokenized using dVAE and used as a target

BEiT: scheme

dVAE



BEiT: scheme

- 40% of patches are masked
- Masked patches are replaced with a learnable embedding
- Blockwise masking with a random aspect ratio and minimum 16 patches

Algorithm 1 Blockwise Masking

Input: $N (= h \times w)$ image patches

Output: Masked positions \mathcal{M}

$\mathcal{M} \leftarrow \{\}$

repeat

$s \leftarrow \text{Rand}(16, 0.4N - |\mathcal{M}|)$ \triangleright *Block size*

$r \leftarrow \text{Rand}(0.3, \frac{1}{0.3})$ \triangleright *Aspect ratio of block*

$a \leftarrow \sqrt{s \cdot r}; b \leftarrow \sqrt{s/r}$

$t \leftarrow \text{Rand}(0, h - a); l \leftarrow \text{Rand}(0, w - b)$

$\mathcal{M} \leftarrow \mathcal{M} \cup \{(i, j) : i \in [t, t + a], j \in [l, l + b]\}$

until $|\mathcal{M}| > 0.4N$ \triangleright *Masking ratio is 40%*

return \mathcal{M}

BEiT: results

Models	Model Size	Resolution	ImageNet
<i>Training from scratch (i.e., random initialization)</i>			
ViT ₃₈₄ -B [DBK ⁺ 20]	86M	384 ²	77.9
ViT ₃₈₄ -L [DBK ⁺ 20]	307M	384 ²	76.5
DeiT-B [TCD ⁺ 20]	86M	224 ²	81.8
DeiT ₃₈₄ -B [TCD ⁺ 20]	86M	384 ²	83.1
<i>Supervised Pre-Training on ImageNet-22K (using labeled data)</i>			
ViT ₃₈₄ -B [DBK ⁺ 20]	86M	384 ²	84.0
ViT ₃₈₄ -L [DBK ⁺ 20]	307M	384 ²	85.2
<i>Self-Supervised Pre-Training on ImageNet-1K (without labeled data)</i>			
iGPT-1.36B [†] [CRC ⁺ 20]	1.36B	224 ²	66.5
ViT ₃₈₄ -B-JFT300M [‡] [DBK ⁺ 20]	86M	384 ²	79.9
MoCo v3-B [CXH21]	86M	224 ²	83.2
MoCo v3-L [CXH21]	307M	224 ²	84.1
DINO-B [CTM ⁺ 21]	86M	224 ²	82.8
BEiT-B (ours)	86M	224 ²	83.2
BEiT ₃₈₄ -B (ours)	86M	384 ²	84.6
BEiT-L (ours)	307M	224 ²	85.2
BEiT ₃₈₄ -L (ours)	307M	384 ²	86.3

Table 1: Top-1 accuracy on ImageNet-1K. We evaluate base- (“-B”) and large-size (“-L”) models at resolutions 224×224 and 384×384 . [†]: iGPT-1.36B contains 1.36 billion parameters, while others are base-size models. [‡]: ViT₃₈₄-B-JFT300M is pretrained with the “masked patch prediction” task on Google’s in-house 300M images, while others use ImageNet.

Models	ADE20K
Supervised Pre-Training on ImageNet	45.3
DINO [CTM ⁺ 21]	44.1
BEiT (ours)	45.6
BEiT + Intermediate Fine-Tuning (ours)	47.7

Table 3: Results of semantic segmentation on ADE20K. We use SETR-PUP [ZLZ⁺20] as the task layer and report results of single-scale inference.

BEiT: ablations

Models	ImageNet	ADE20K
BEiT (300 Epochs)	82.86	44.65
– Blockwise masking	82.77	42.93
– Visual tokens (i.e., recover masked pixels)	81.04	41.38
– Visual tokens – Blockwise masking	80.50	37.09
+ Recover 100% visual tokens	82.59	40.93
– Masking + Recover 100% visual tokens	81.67	36.73
Pretrain longer (800 epochs)	83.19	45.58

Table 4: Ablation studies for BEiT pre-training on image classification and semantic segmentation.

- Directly using pixel-level auto-encoding pushes the model to focus on short-range dependencies and high-frequency details
- BEiT overcomes the above issue by predicting discrete visual tokens, which summarizes the details to high-level abstractions.

BEiT: attention maps

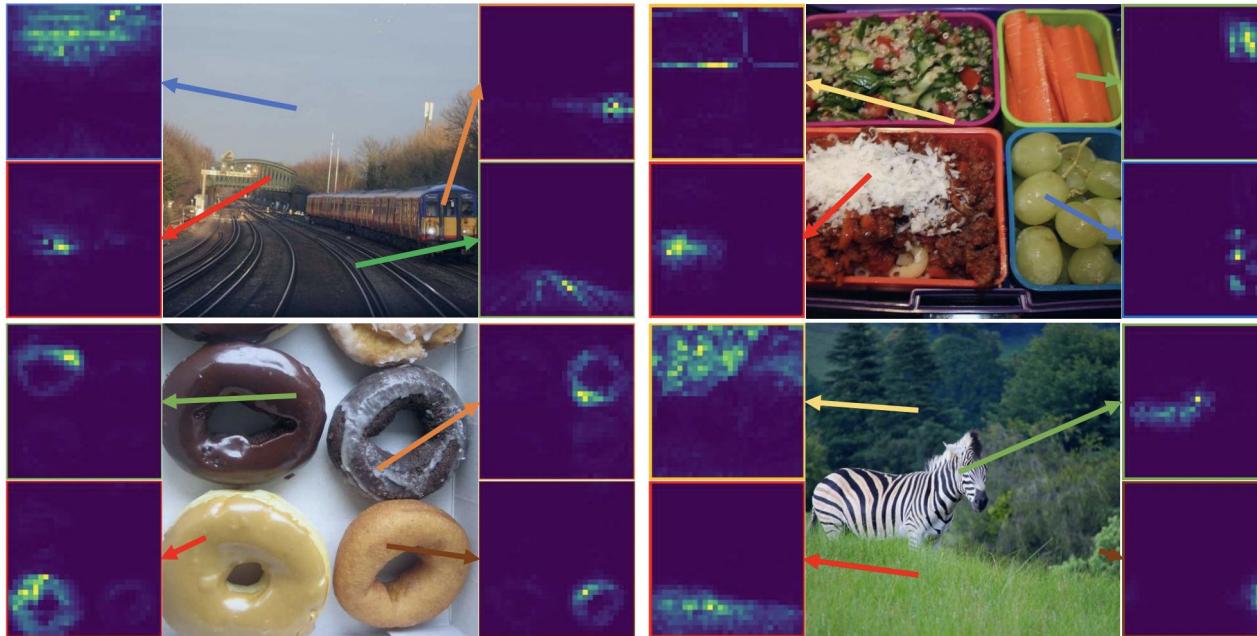


Figure 2: Self-attention map for different reference points. The self-attention mechanism in BEiT is able to separate objects, although self-supervised pre-training does not use manual annotations.

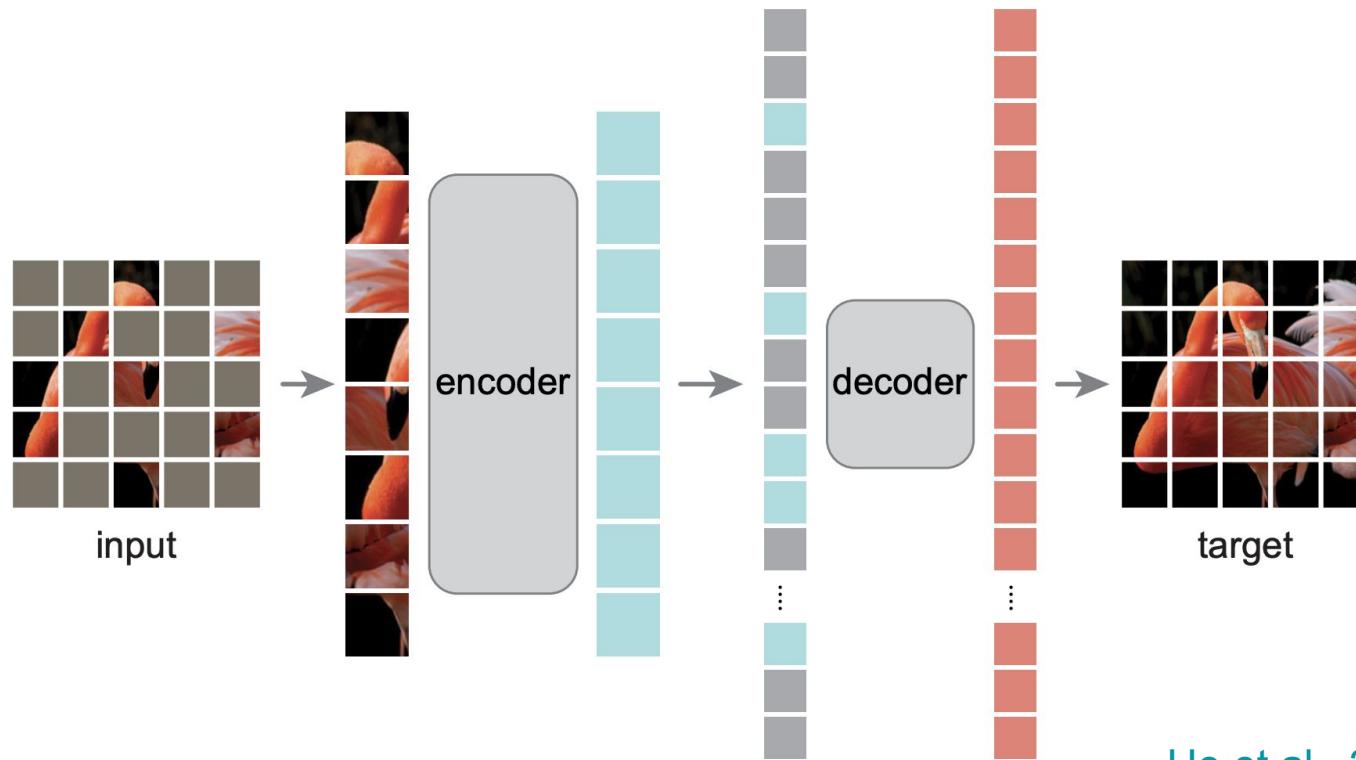
[Bao et al., 2021](#)

MAE

Masked AutoEncoders Are Scalable Vision Learners, FAIR

- Encoder-decoder architecture
- Encoder applied only to non-masked patches
- Mask high proportion (75%) of the input image
- Reconstruct images in the pixel space (i.e. not tokens as in BEiT), loss computed over masked patches only
- Decoder is flexible, smaller than encoder and takes both encodings of non-masked patches and mask tokens

MAE: scheme



MAE: reconstruction

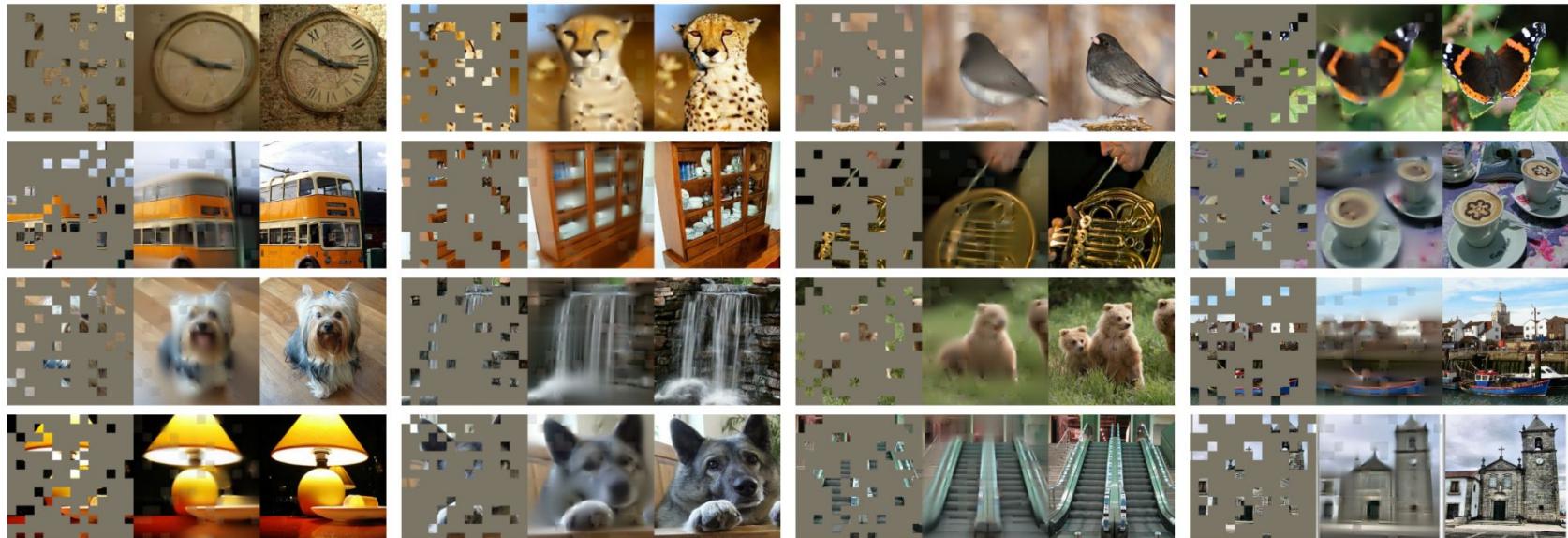
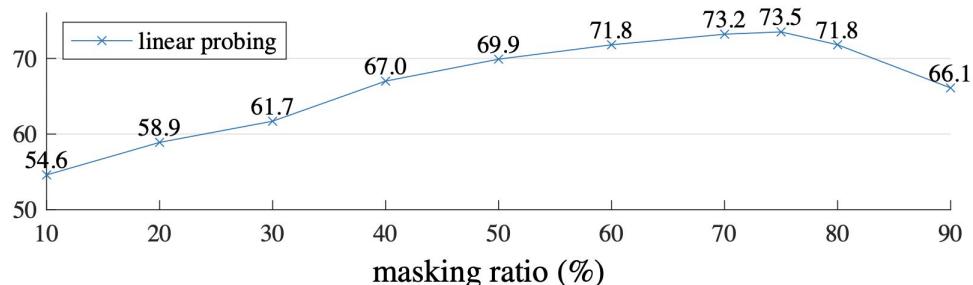
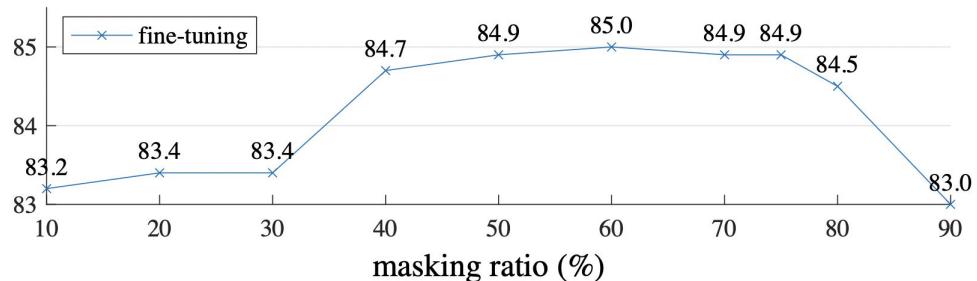
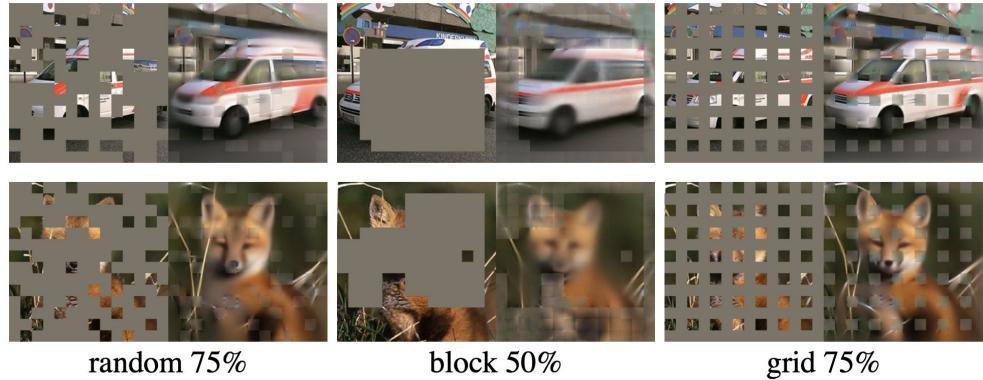


Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction[†] (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix.

[†]*As no loss is computed on visible patches, the model output on visible patches is qualitatively worse. One can simply overlay the output with the visible patches to improve visual quality. We intentionally opt not to do this, so we can more comprehensively demonstrate the method's behavior.*

MAE: masking



MAE: ablations

blocks	ft	lin
1	84.8	65.5
2	84.9	70.0
4	84.9	71.9
8	84.9	73.5
12	84.4	73.3

(a) **Decoder depth.** A deep decoder can improve linear probing accuracy.

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	85.4	73.9
PCA	84.6	72.3
dVAE token	85.3	71.6

(d) **Reconstruction target.** Pixels as reconstruction targets are effective.

dim	ft	lin
128	84.9	69.1
256	84.8	71.3
512	84.9	73.5
768	84.4	73.1
1024	84.3	73.1

(b) **Decoder width.** The decoder can be narrower than the encoder (1024-d).

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	84.9	73.5
crop + color jit	84.3	71.9

(e) **Data augmentation.** Our MAE works with minimal or no augmentation.

case	ft	lin	FLOPs
encoder w/ [M]	84.2	59.6	3.3×
encoder w/o [M]	84.9	73.5	1×

(c) **Mask token.** An encoder without mask tokens is more accurate and faster (Table 2).

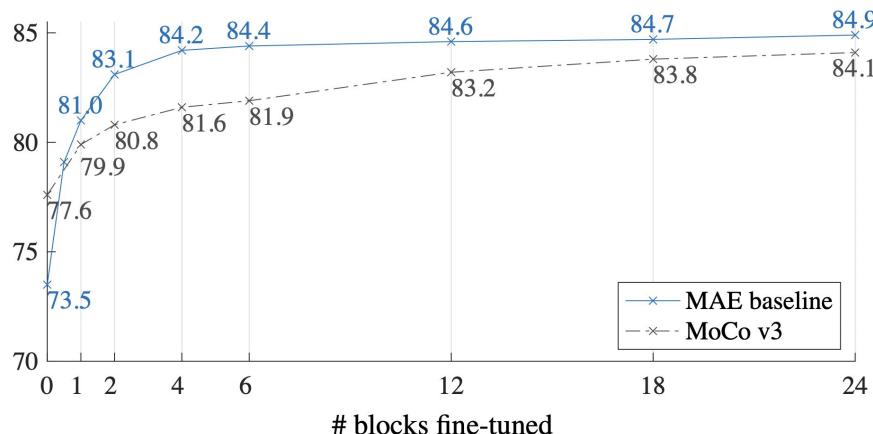
case	ratio	ft	lin
random	75	84.9	73.5
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

(f) **Mask sampling.** Random sampling works the best. See Figure 6 for visualizations.

Table 1. **MAE ablation experiments** with ViT-L/16 on ImageNet-1K. We report fine-tuning (ft) and linear probing (lin) accuracy (%). If not specified, the default is: the decoder has depth 8 and width 512, the reconstruction target is unnormalized pixels, the data augmentation is random resized cropping, the masking ratio is 75%, and the pre-training length is 800 epochs. Default settings are marked in gray .

MAE: results

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H ₄₄₈
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	<u>83.6</u>	<u>85.9</u>	<u>86.9</u>	87.8



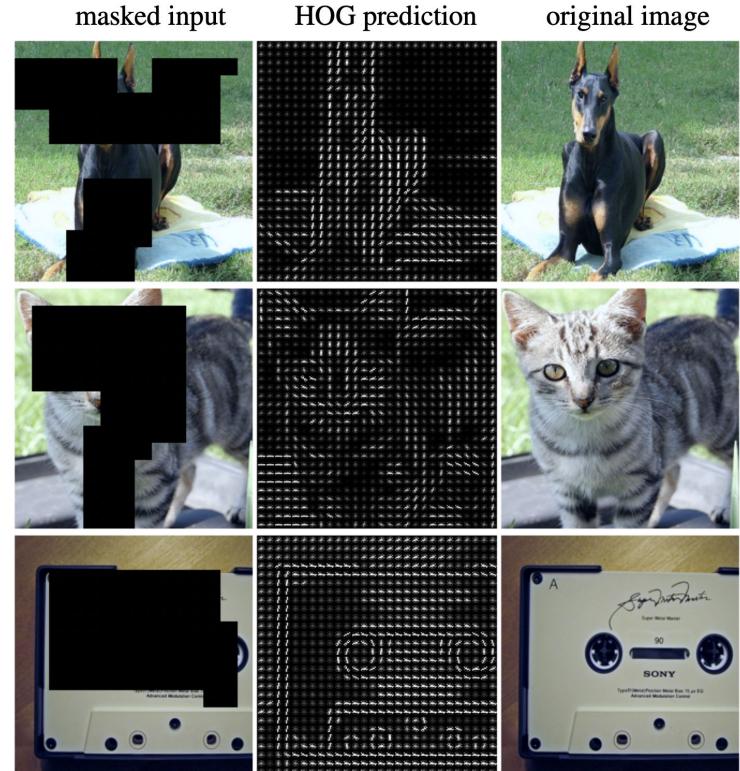
method	pre-train data	AP ^{box}		AP ^{mask}	
		ViT-B	ViT-L	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3	42.9	43.9
MoCo v3	IN1K	47.9	49.3	42.7	44.0
BEiT	IN1K+DALLE	49.8	53.3	44.4	47.1
MAE	IN1K	50.3	53.3	44.9	47.2

Table 4. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline. All entries are based on our implementation. Self-supervised entries use IN1K data *without* labels. Mask AP follows a similar trend as box AP.

MaskFeat

Masked Feature Prediction, FAIR

- Pre-training of video models
- Use features of masked patches (i.e. HOG) as a target for masked modelling



MaskFeat: scheme & features

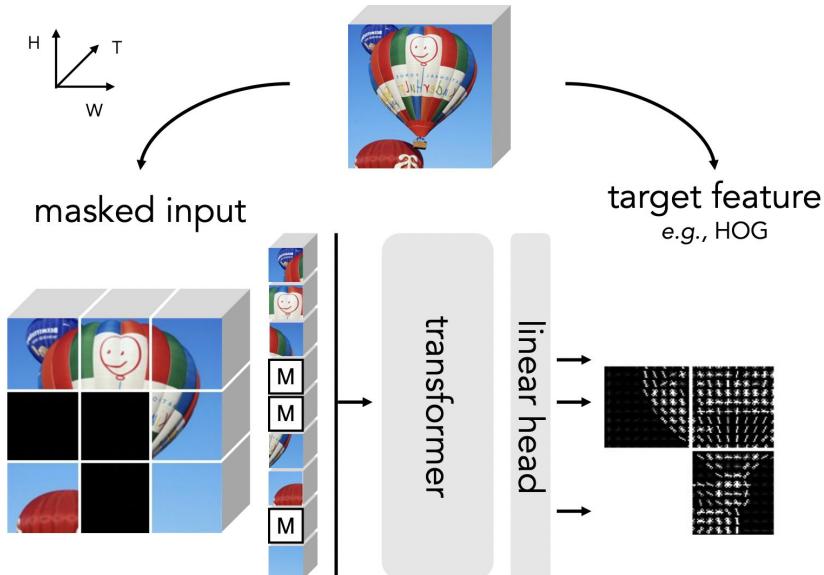
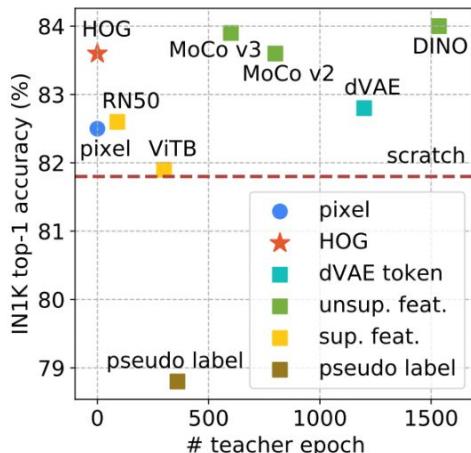


Figure 2. **MaskFeat pre-training.** We randomly replace the input space-time cubes of a video with a [MASK] token and directly regress features (e.g. HOG) of the masked regions. After pre-training, the Transformer is fine-tuned on end tasks.

feature type	one-stage	variant	top-1
scratch	-	MViTv2-S [46]	81.1
pixel	✓	RGB	80.7
image descriptor	✓	HOG [18]	82.2
dVAE	✗	DALL-E [58]	81.7
unsupervised feature	✗	DINO [9], ViT-B	82.5
supervised feature	✗	MViT-B [26]	81.9

Table 1. **Comparing target features for MaskFeat (video).** All variants are pre-trained for 300 epochs on MViTv2-S, 16×4 with MaskFeat. We report fine-tuning top-1 on K400. Default is gray .

MaskFeat: results

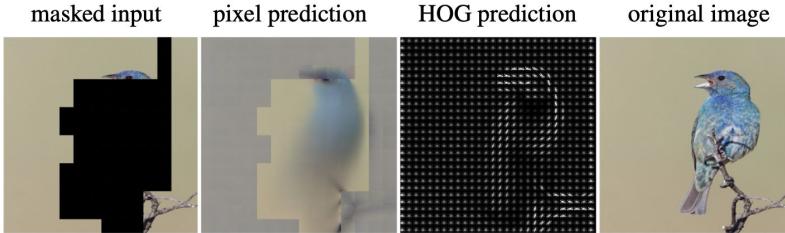


feature type	one-stage	variant	arch.	param.	epoch [†]	top-1
scratch	-	DeiT [63]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5
image descriptor	✓	HOG [18]	-	-	-	83.6
dVAE token	✗	DALL-E [58]	dVAE	54	1199	82.8
unsupervised feature	✗	MoCo v2 [15]	ResNet50	23	800	83.6
unsupervised feature	✗	MoCo v3 [17]	ViT-B	85	600	83.9
unsupervised feature	✗	DINO [9]	ViT-B	85	1535	84.0
supervised feature	✗	pytorch [53]	ResNet50	23	90	82.6
supervised feature	✗	DeiT [63]	ViT-B	85	300	81.9
pseudo-label	✗	Token Labeling [42]	NFNet-F6	438	360	78.8

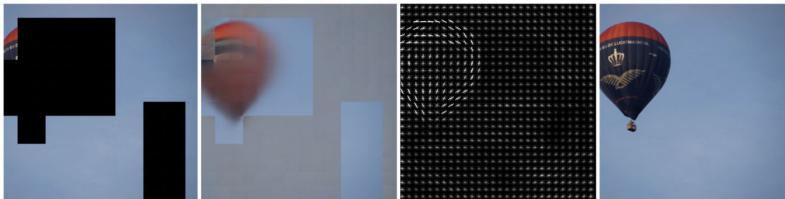
Table 2. **Comparing target features for MaskFeat (image).** For all targets, ViT-B is pre-trained with MaskFeat for 300 epochs on IN-1K. We report 100-epoch fine-tuning accuracy on IN-1K. For two-stage targets, we report the *teacher* architecture, number of parameters (M), and effective epoch[†] on IN-1K. The default entry is marked in gray. The plot on the left visualizes the acc/epoch trade-off of the table.

[†] Different teachers use different training strategies. dVAE is pre-trained on an external 250M dataset, while self-supervised methods require multi-view training. To measure the cost in a unified way, we normalize the number of epochs by the cost of one epoch on IN-1K training set with *one 224² view*.

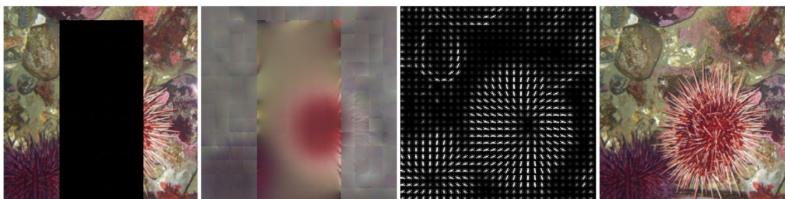
MaskFeat: results



Both two predictions make good sense given a small visible region at the bird's head.



Pixel with **color ambiguity**: Though pixel prediction makes a sensible guess on the balloon, the loss penalty is large because of unmatched color (red vs. black).



Pixel with **texture ambiguity**: Pixel prediction is blurry in texture-rich area because of ambiguity, while HOG successfully characterizes major edge directions.

masking	frame	tube	cube
top-1	81.0 (-1.2)	81.9 (-0.3)	82.2

Table 6. **Masking strategy.** Varying the strategy of masking in spatiotemporal data. The default entry is highlighted in gray .

pre-train	extra data	extra model	ViT-B	ViT-L
scratch [63]	-	-	81.8	81.5
supervised ₃₈₄ [23]	IN-21K	-	84.0	85.2
MoCo v3 [17]	-	momentum ViT	83.2	84.1
DINO [9]	-	momentum ViT	82.8	-
BEiT [2]	DALL-E	dVAE	83.2	85.2
MaskFeat (w/ HOG)	-	-	84.0	85.7

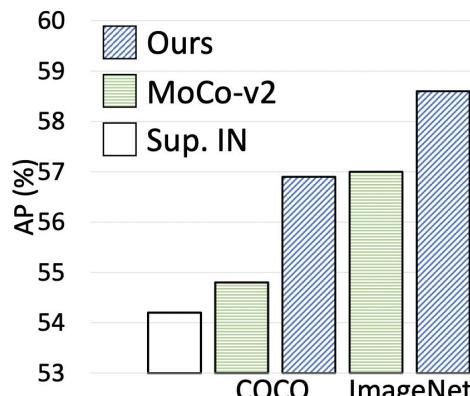
Table 7. **Comparison with previous work on IN-1K.** All entries are pre-trained on IN-1K train split, except supervised₃₈₄ using IN-21K. MoCo v3 and DINO use momentum encoder. BEiT uses 250M DALL-E data to pre-train dVAE. All entries are trained and evaluated at image size 224² except supervised₃₈₄ at 384².

Plan

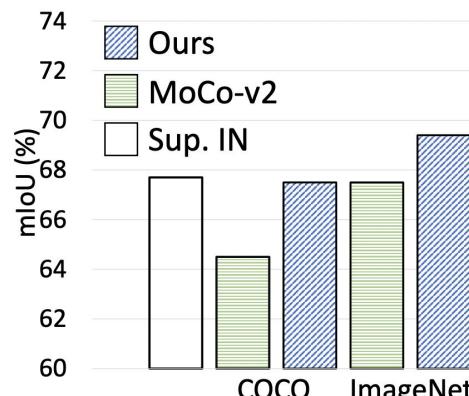
- **Transformers for images**
 - ViT, DEiT
 - Self-distillation with no labels (DINO)
- **Masked Image Modeling**
 - BEiT, MAE
 - MaskFeat
- **Improving Contrastive Learning**
 - Dense CL
 - Supervised CL
 - Augmentations in CL (DiLo, LooC)

Dense Contrastive Learning

- **Global contrastive learning** works good for classification tasks
- Do **local contrastive learning** for dense tasks (i.e. detection & segmentation)



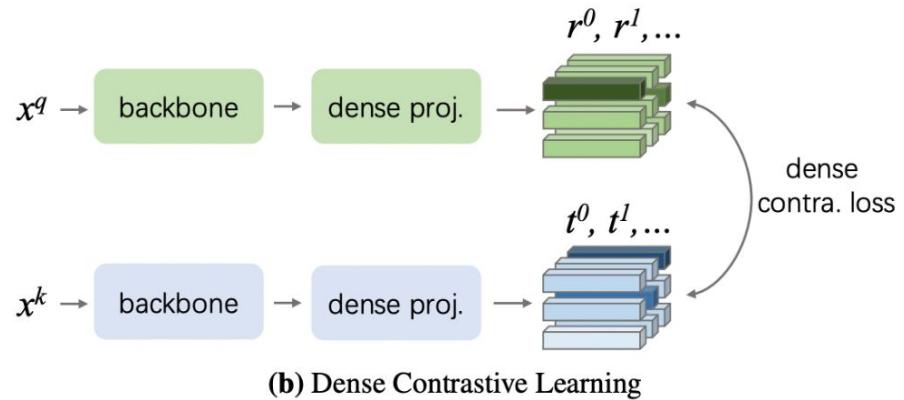
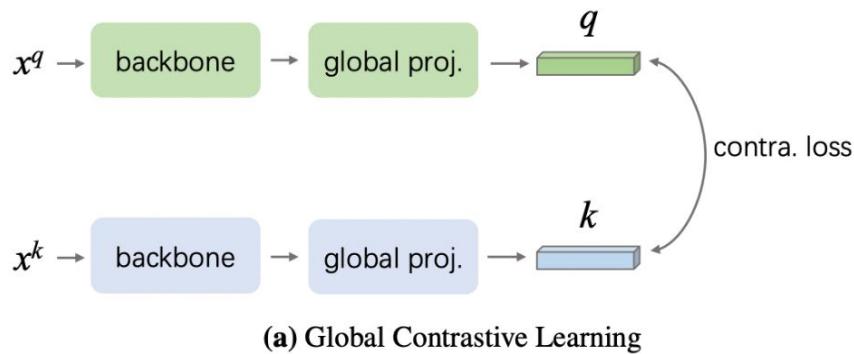
(a) Object Detection



(b) Semantic Segmentation

[Want et al., 2020](#)

Dense Contrastive Learning: scheme



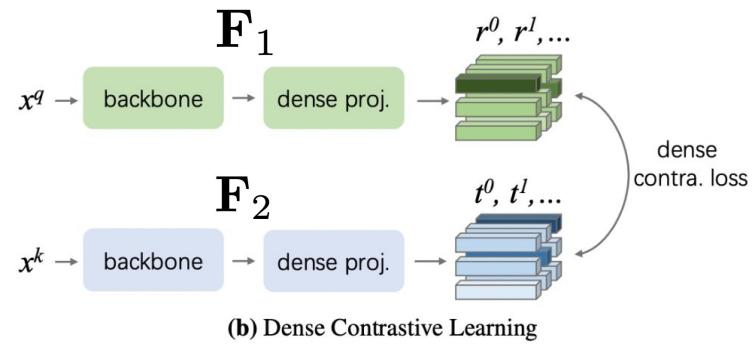
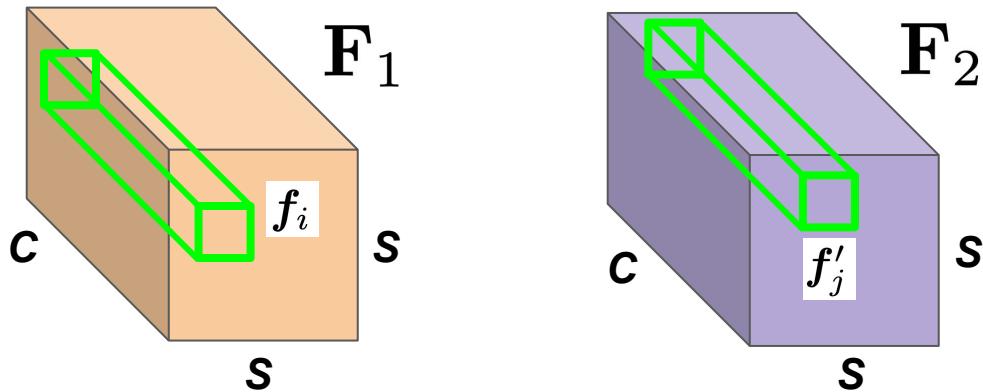
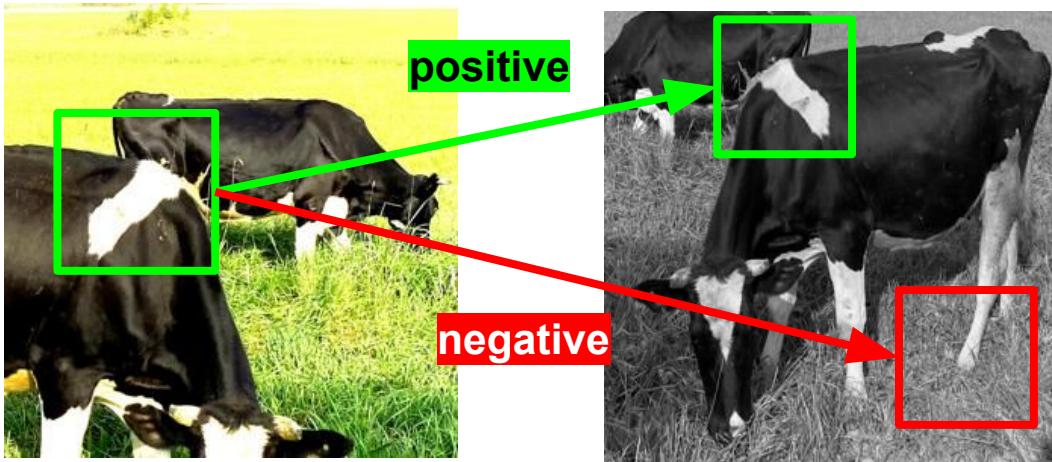
$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\exp(q \cdot k_+) + \sum_{k_-} \exp(q \cdot k_- / \tau)}$$

$$\mathcal{L}_r = \frac{1}{S^2} \sum_s -\log \frac{\exp(r^s \cdot t^s_+ / \tau)}{\exp(r^s \cdot t^s_+) + \sum_{t^s_-} \exp(r^s \cdot t^s_- / \tau)}$$

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_q + \lambda\mathcal{L}_r$$

[Want et al., 2020](#)

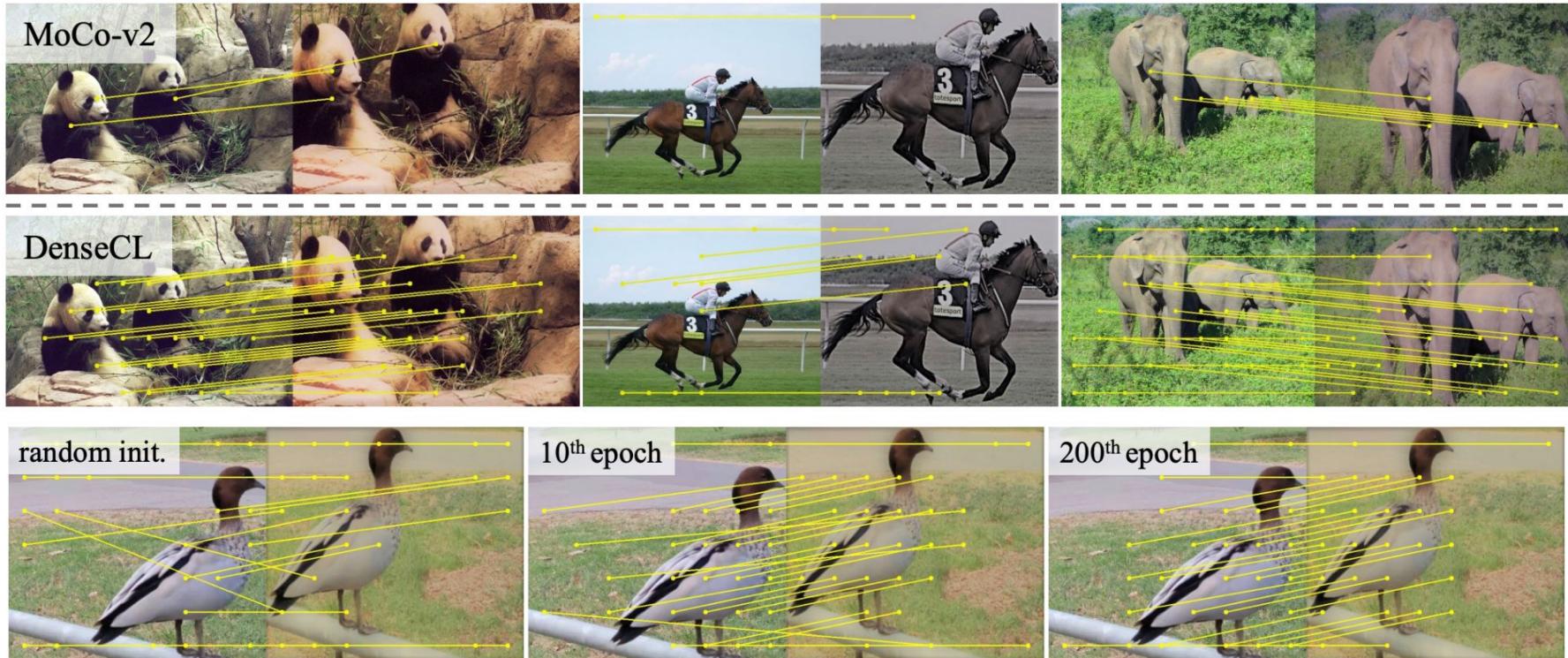
Dense Contrastive Learning: scheme



$$c_i = \arg \max_j sim(f_i, f'_j)$$

[Want et al., 2020](#)

Dense Contrastive Learning: scheme



Dense Contrastive Learning: results

pre-train	AP ^b	AP ^b ₅₀	AP ^b ₇₅	AP ^m	AP ^m ₅₀	AP ^m ₇₅
random init.	32.8	50.9	35.3	29.9	47.9	32.0
super. IN	39.7	59.5	43.3	35.9	56.6	38.6
MoCo-v2 CC	38.5	58.1	42.1	34.8	55.3	37.3
DenseCL CC	39.6	59.3	43.3	35.7	56.5	38.4
SimCLR IN	38.5	58.0	42.0	34.8	55.2	37.2
BYOL IN	38.4	57.9	41.9	34.9	55.3	37.5
MoCo-v2 IN	39.8	59.8	43.6	36.1	56.9	38.7
DenseCL IN	40.3	59.9	44.3	36.4	57.0	39.2

Table 2 – Object detection and instance segmentation fine-tuned on COCO. ‘CC’ and ‘IN’ indicate the pre-training mod-

pre-train	AP ^b	AP ^b ₅₀	AP ^b ₇₅	AP ^m	AP ^m ₅₀	AP ^m ₇₅
random init.	20.6	34.0	21.5	18.9	31.7	19.8
super. IN	23.6	37.7	25.4	21.8	35.4	23.2
MoCo-v2 CC	22.8	36.4	24.2	20.9	34.6	21.9
DenseCL CC	24.1	38.1	25.6	21.9	36.0	23.0
MoCo-v2 IN	23.8	37.5	25.6	21.8	35.4	23.2
DenseCL IN	24.8	38.8	26.8	22.6	36.8	23.9

Table 3 – Semi-supervised object detection and instance segmentation fine-tuned on COCO. During the fine-tuning,

time/epoch	COCO	ImageNet
MoCo-v2	1'45"	16'48"
DenseCL	1'46"	16'54"

Table 9 – Pre-training time comparison. The training time per epoch is reported. We measure the results on the same 8-GPU machine. The training time overhead introduced by DenseCL is less than 1%.

[Want et al., 2020](#)

Dense Contrastive Learning: ablations

λ	Detection			Classification mAP
	AP	AP ₅₀	AP ₇₅	
0.0	54.7	81.0	60.6	82.6
0.1	55.2	81.4	61.4	82.9
0.3	56.2	81.5	62.6	83.3
0.5	56.7	81.7	63.0	82.9
0.7	56.8	81.9	63.1	81.0
0.9	55.5	80.9	61.3	77.8
1.0*	53.5	79.5	58.8	68.9

Table 5 – Ablation study of weight λ . $\lambda = 0$ is the MoCo-v2 baseline. $\lambda = 0.5$ shows the best trade-off between detection and classification. '*' indicates training with warm-up, as discussed in Section 3.4.

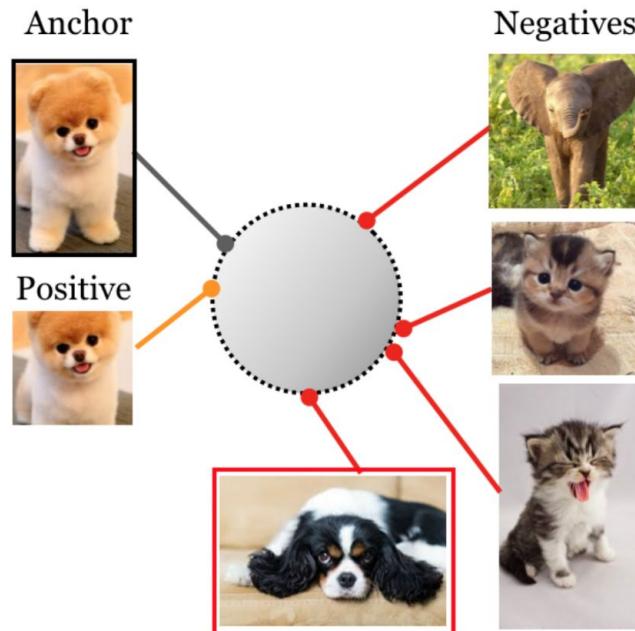
strategy	Detection			Classification mAP
	AP	AP ₅₀	AP ₇₅	
random	56.0	81.3	62.0	81.7
max-sim Θ	56.0	81.5	62.1	81.8
max-sim F	56.7	81.7	63.0	82.9

Table 6 – Ablation study of matching strategy. To extract the dense correspondence according to the backbone features F_1 and F_2 shows the best results.

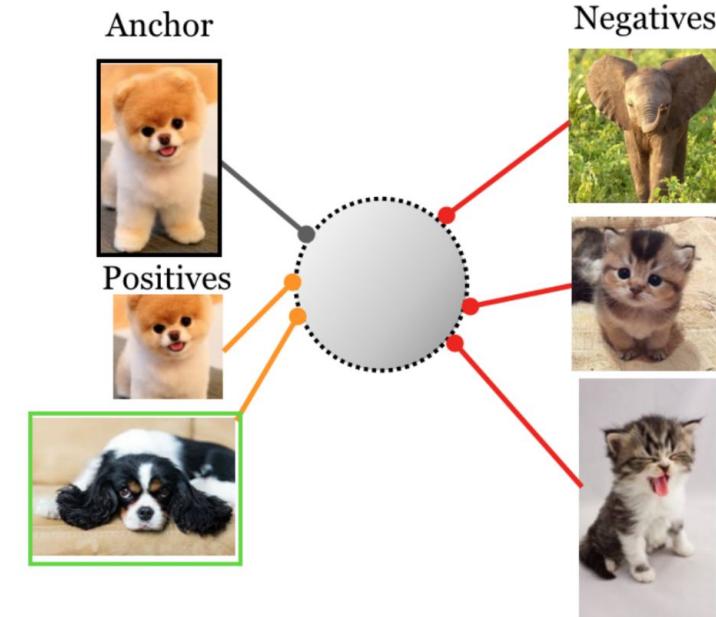
grid size	Detection			Classification mAP
	AP	AP ₅₀	AP ₇₅	
1	54.6	80.8	60.5	82.2
3	55.6	81.3	61.5	81.6
5	56.1	81.4	62.2	82.6
7	56.7	81.7	63.0	82.9
9	56.7	82.1	63.2	82.9

Table 7 – Ablation study of grid size S . The results increase as the S gets larger. We use grid size being 7 in other experiments, as the performance becomes stable when the S grows beyond 7.

Supervised Contrastive Learning



Self Supervised Contrastive



Supervised Contrastive

Supervised Contrastive Learning

$$\mathcal{L}^{self} = \sum_{i \in I} \mathcal{L}_i^{self} = - \sum_{i \in I} \log \frac{\exp(z_i \cdot z_{j(i)}/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)}$$

$$\mathcal{L}_{out}^{sup} = \sum_{i \in I} \mathcal{L}_{out,i}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)}$$

$$\mathcal{L}_{in}^{sup} = \sum_{i \in I} \mathcal{L}_{in,i}^{sup} = \sum_{i \in I} -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)} \right\}$$

Supervised Contrastive Learning

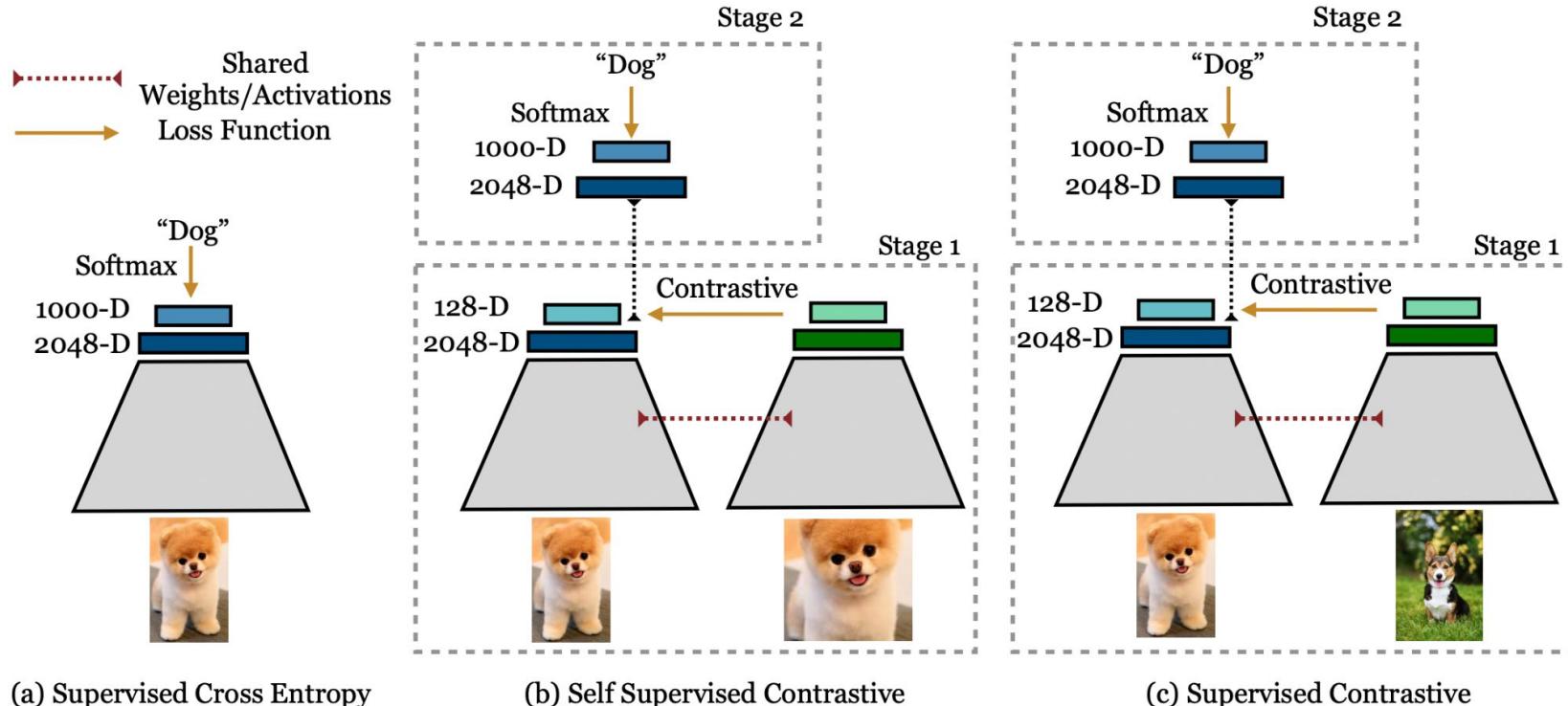
$$\mathcal{L}^{self} = \sum_{i \in I} \mathcal{L}_i^{self} = - \sum_{i \in I} \log \frac{\exp(z_i \cdot z_{j(i)}/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)}$$

$$\mathcal{L}_{in}^{sup} \leq \mathcal{L}_{out}^{sup}$$

$$\mathcal{L}_{out}^{sup} = \sum_{i \in I} \mathcal{L}_{out,i}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)}$$
$$\mathcal{L}_{in}^{sup} = \sum_{i \in I} \mathcal{L}_{in,i}^{sup} = \sum_{i \in I} -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)} \right\}$$

Loss	Top-1
\mathcal{L}_{out}^{sup}	78.7%
\mathcal{L}_{in}^{sup}	67.4%

Supervised Contrastive Learning



Supervised Contrastive Learning: results

Loss	Architecture	Augmentation	Top-1	Top-5
Cross-Entropy (baseline)	ResNet-50	MixUp [61]	77.4	93.6
Cross-Entropy (baseline)	ResNet-50	CutMix [60]	78.6	94.1
Cross-Entropy (baseline)	ResNet-50	AutoAugment [5]	78.2	92.9
Cross-Entropy (our impl.)	ResNet-50	AutoAugment [30]	77.6	95.3
SupCon	ResNet-50	AutoAugment [5]	78.7	94.3
Cross-Entropy (baseline)	ResNet-200	AutoAugment [5]	80.6	95.3
Cross-Entropy (our impl.)	ResNet-200	Stacked RandAugment [49]	80.9	95.2
SupCon	ResNet-200	Stacked RandAugment [49]	81.4	95.9
SupCon	ResNet-101	Stacked RandAugment [49]	80.2	94.7

Supervised Contrastive Learning: ablations

Loss	Architecture	rel. mCE	mCE
Cross-Entropy (baselines)	AlexNet [28]	100.0	100.0
	VGG-19+BN [44]	122.9	81.6
	ResNet-18 [17]	103.9	84.7
Cross-Entropy (our implementation)	ResNet-50	96.2	68.6
	ResNet-200	69.1	52.4
Supervised Contrastive	ResNet-50	94.6	67.2
	ResNet-200	66.5	50.6

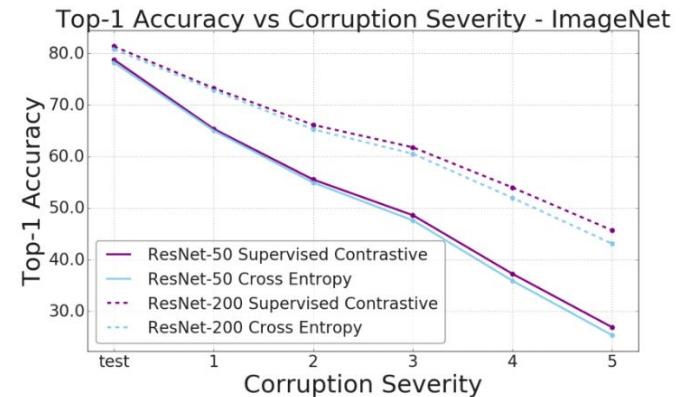
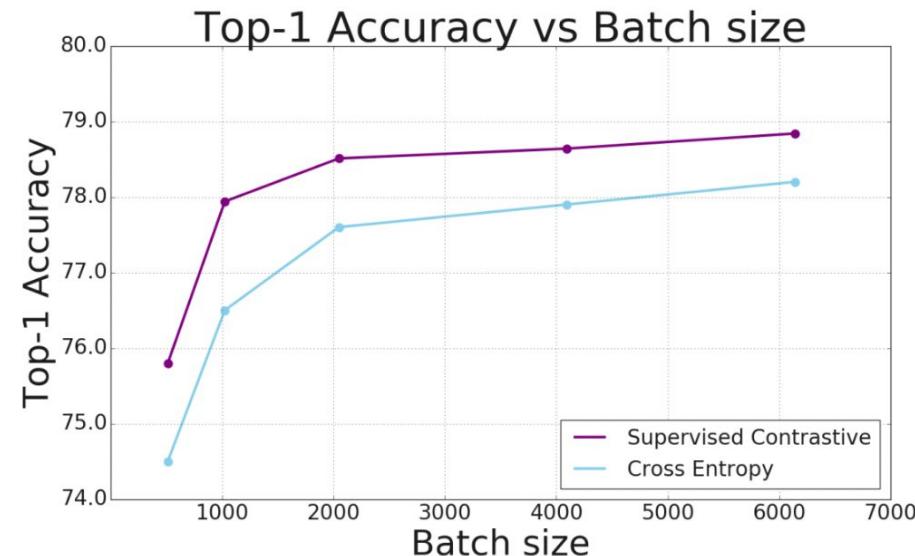
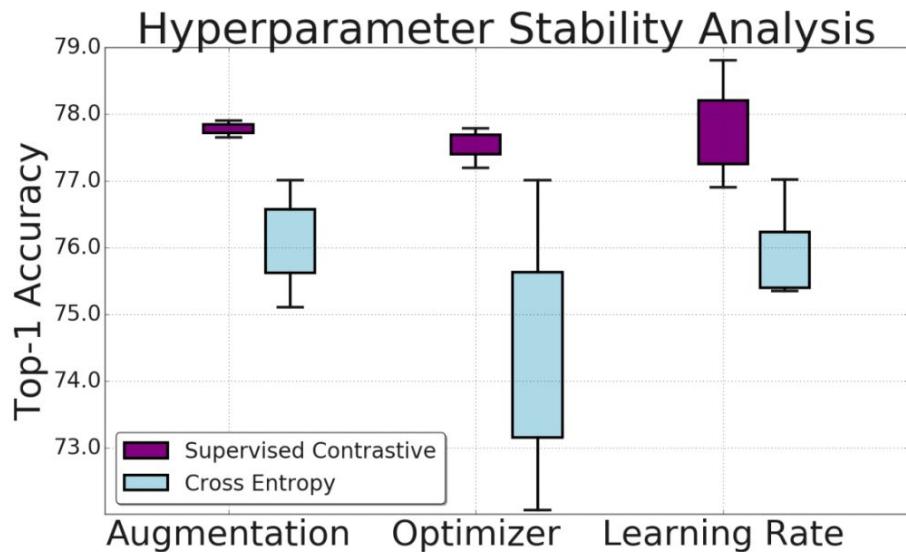


Figure 3: Training with supervised contrastive loss makes models more robust to corruptions in images. **Left:** Robustness as measured by Mean Corruption Error (mCE) and relative mCE over the ImageNet-C dataset [19] (lower is better). **Right:** Mean Accuracy as a function of corruption severity averaged over all various corruptions. (higher is better).

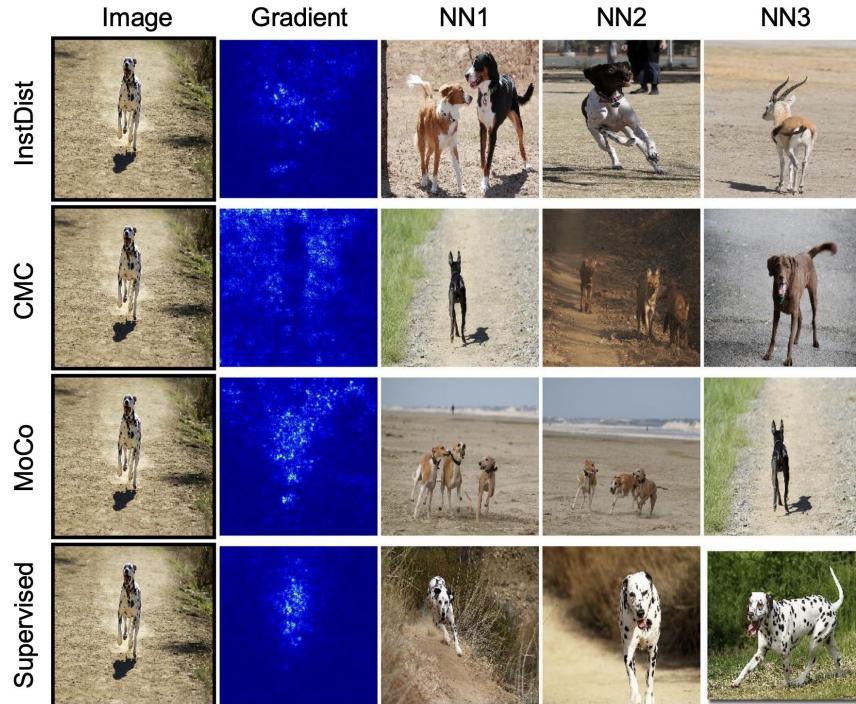
Supervised Contrastive Learning: ablations



DiLo

Distilling Localization, Microsoft Research

- Self-supervised ConvNets are bad at localizing object in the image
- Main idea: force invariance to backgrounds
- Use saliency masks to cut object from the image and swap background



DiLo: saliency masks

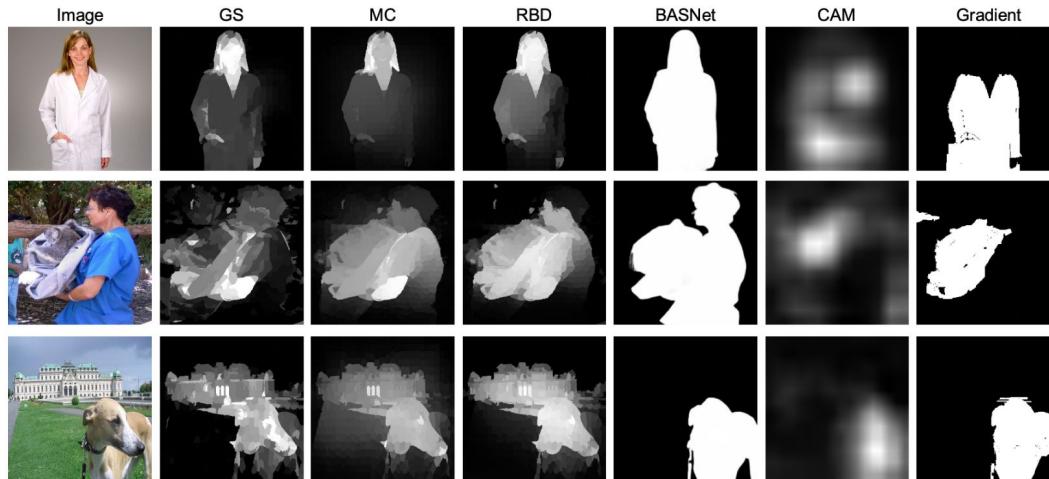
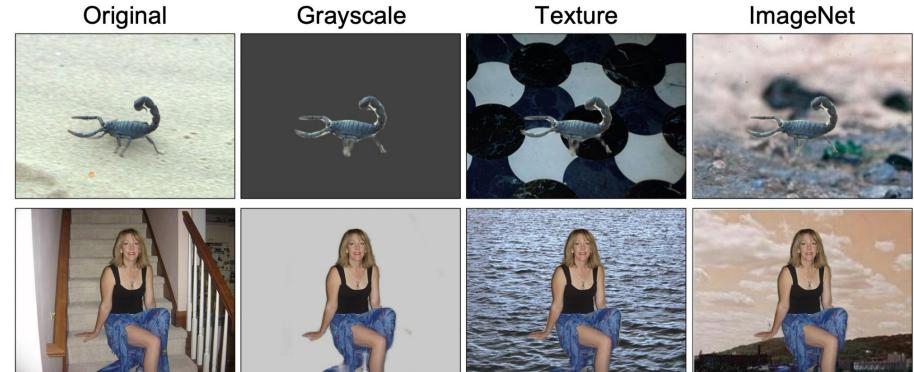


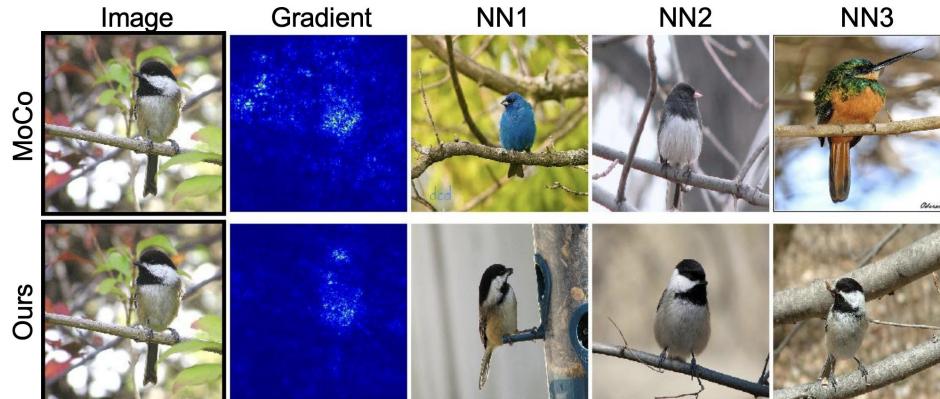
Figure 3: Examples of saliency estimations methods. We show 6 saliency estimations, including traditional methods (GS (Wei et al. 2012), MC (Jiang et al. 2013a), RBD (Zhu et al. 2014)), a network predicted saliency BASNet (Qin et al. 2019), and class-specific methods visualized from a pretrained network (CAM (Zhou et al. 2016), Gradient (Simonyan, Vedaldi, and Zisserman 2013)).

Zhao et al., 2020

DiLo: ablations

Table 2: Ablation studies for investigating copy-and-pasting augmentations: (a) on various saliency estimation methods (b) on controlling the ratio of using copy-and-pasting augmentation (c) on various background images (d) on blending options.

(a)					(b)			(c)			(d)		
Saliency	F_β	MAE	Acc	Δ	Aug Ratio	Linear	Δ	Background	Linear	Δ	Blending	Linear	Δ
MoCo	-	-	60.6	-	MoCo	60.6	-	MoCo	60.6	-	MoCo	60.6	-
GS	0.557	0.173	62.7	+2.1	30%	62.8	+2.2	Texture	60.6	+0.0	No blend	62.4	+1.8
MC	0.627	0.186	62.1	+1.5	50%	62.2	+1.6	Imagenet	62.1	+1.5	Gaussian	62.5	+1.9
RBD	0.630	0.144	62.8	+2.2	70%	61.6	+1.0	Grayscale	62.8	+2.2	Mix	62.8	+2.2
BASNet	0.805	0.056	65.0	+4.4	100%	47.6	-13.0						



DiLo: results

Table 3: Distilling localization on various contrastive representation learning models for ImageNet classification.

Methods	Original	DiLo-RBD	DiLo-BasNet
InstDist	56.5	59.3	62.9
CMC	63.4	65.0	66.9
MoCo-v1	60.6	62.8	65.0
MoCo-v2	67.5	67.9	69.2

Table 4: Transfer learning for object detection on VOC 0712. We present the gap to ImageNet supervised pre-training in the brackets for reference. All numbers are the averages of three independent runs.

Method	AP	AP_{50}	AP_{75}
Supervised	53.5	-	81.3
MoCo	55.9 (+2.4)	81.5 (+0.2)	62.6 (+3.8)
DiLo-RBD	56.5 (+3.0)	81.9 (+0.6)	63.3 (+4.5)
DiLo-BasNet	56.9 (+3.4)	82.1 (+0.8)	64.1 (+5.3)

Table 5: Transfer learning for object detection and instance segmentation on COCO. Model is finetuned with Mask-RCNN ResNet50-FPN pipeline and 1x schedule.

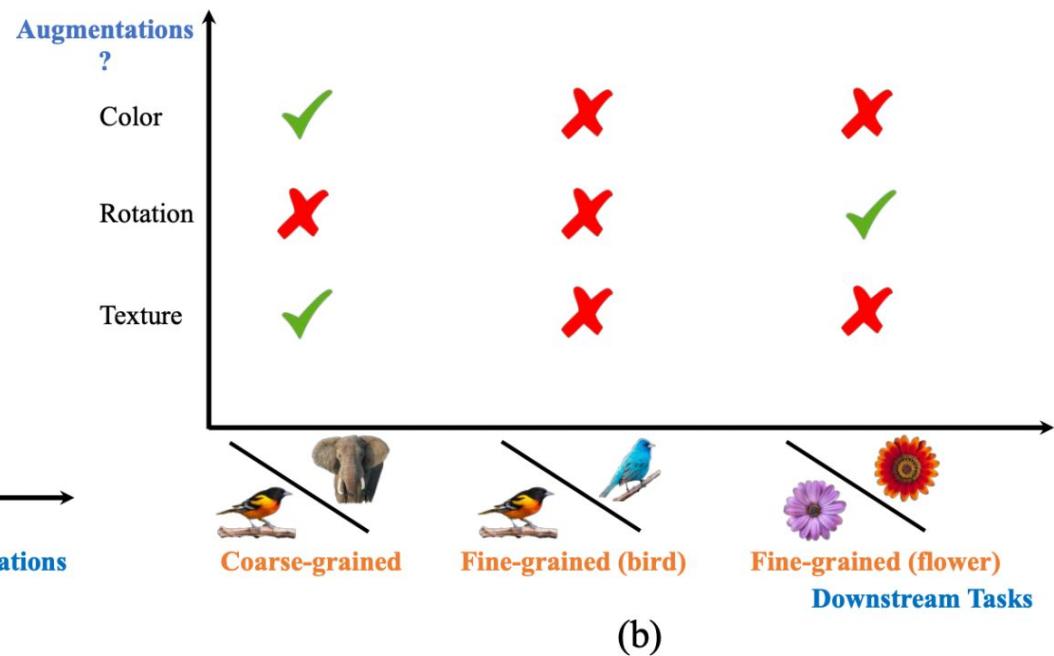
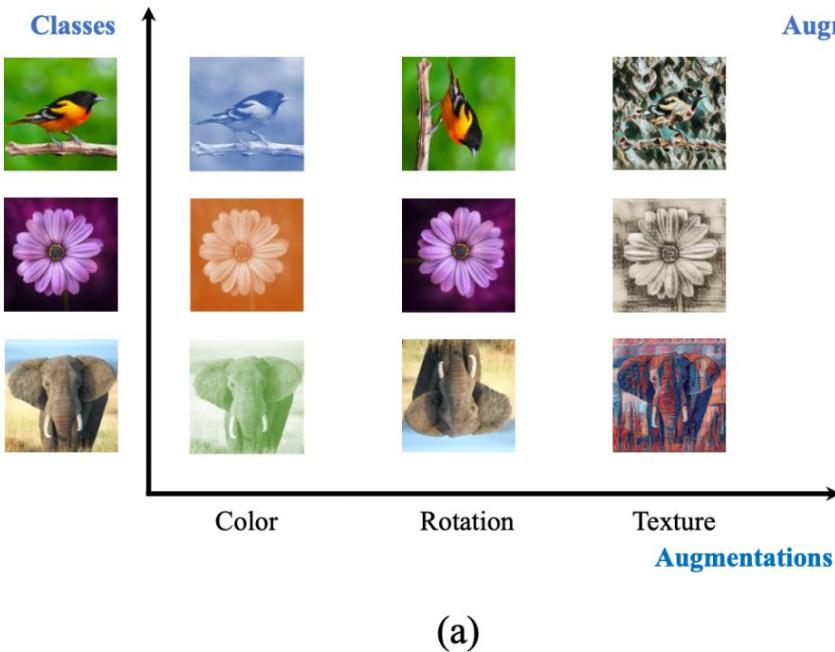
Method	AP^{bb}	AP_{50}^{bb}	AP_{75}^{bb}	AP^{mk}	AP_{50}^{mk}	AP_{75}^{mk}
Supervised	39.7	59.5	43.3	35.9	56.6	38.6
MoCo	39.4	59.1	42.9	35.6	56.2	38.0
DiLo-RBD	39.8	59.5	43.3	36.0	56.7	38.6
DiLo-BasNet	40.1	60.0	44.0	36.3	56.8	39.0

What should not be contrastive in contrastive learning

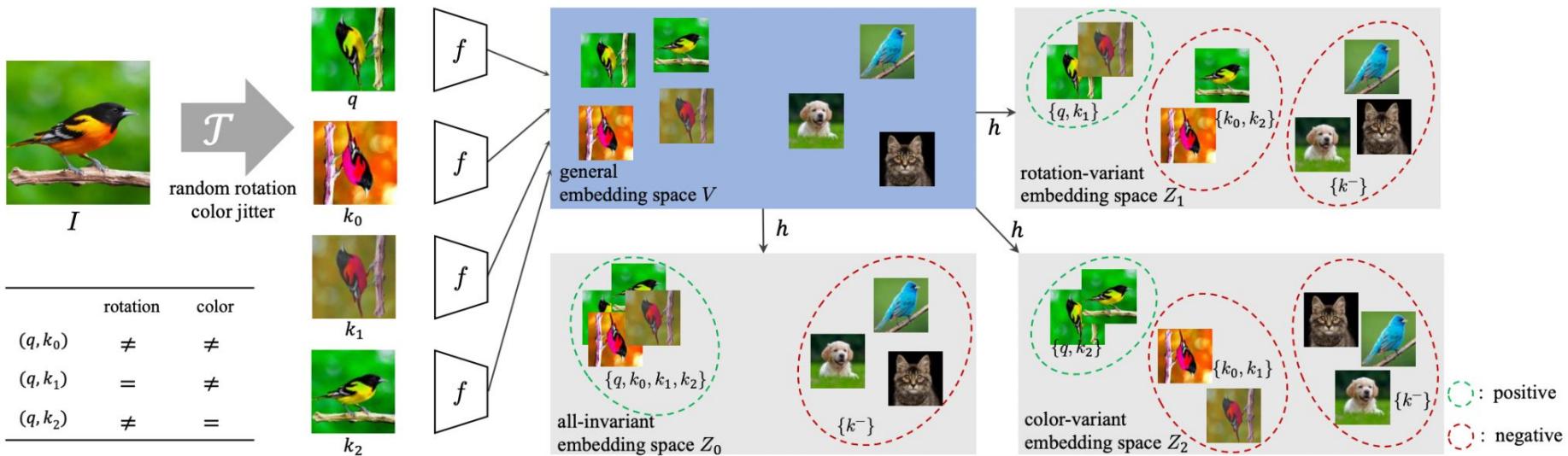
Leave-one-out Contrastive Learning (LooC)

- Contrastive models are trained to force invariance over augmentations
- This invariance may be beneficial to one task and harmful to the other task
- Main idea: multi-head network with different embedding spaces, each forcing invariance over a single augmentation type

LooC: augmentation invariance



LooC: scheme



LooC: results

model	Rotation Acc.	IN-100	
		top-1	top-5
Supervised	72.3	83.7	95.7
MoCo	61.1	81.0	95.2
MoCo + Rotation	43.3	79.4	94.1
MoCo + Rotation (same for q and k)	45.5	78.1	94.3
LooC + Rotation [ours]	65.2	80.2	95.5

model	Augmentation		iNat-1k		CUB-200		Flowers-102		IN-100	
	Color	Rotation	top-1	top-5	top-1	top-5	5-shot	10-shot	top-1	top-5
MoCo	✓		36.2	62.0	36.7	64.7	67.9 (± 0.5)	77.3 (± 0.1)	81.0	95.2
LooC	✓		41.2	67.0	40.1	69.7	68.2 (± 0.6)	77.6 (± 0.1)	81.1	95.3
		✓	40.0	65.4	38.8	67.0	70.1 (± 0.4)	79.3 (± 0.1)	80.2	95.5
	✓	✓	44.0	69.3	39.6	69.2	70.9 (± 0.3)	80.8 (± 0.2)	79.2	94.7
LooC++	✓	✓	46.1	71.5	39.3	69.3	68.1 (± 0.4)	78.8 (± 0.2)	81.2	95.2