Self-supervised Pre-training with Masked Image Modeling

Ildus Sadrtdinov

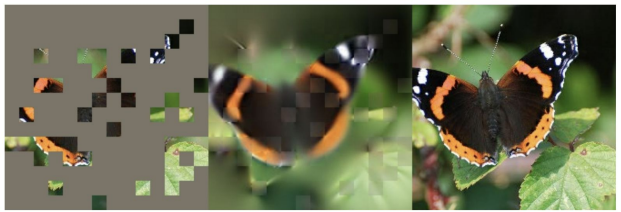
Bayesian seminar

31.03.23

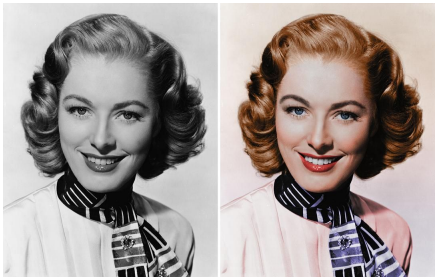
Self-supervised pre-training

generative

masked image modelling



generative pre-text tasks



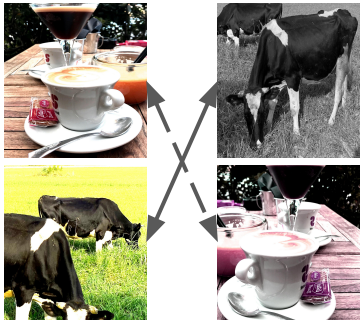
- colorization
- inpainting

discriminative

discriminative pre-text tasks

- rotation angle prediction
- jigsaw puzzle

contrastive tasks



Contrastive learning

Main idea of contrastive learning

X

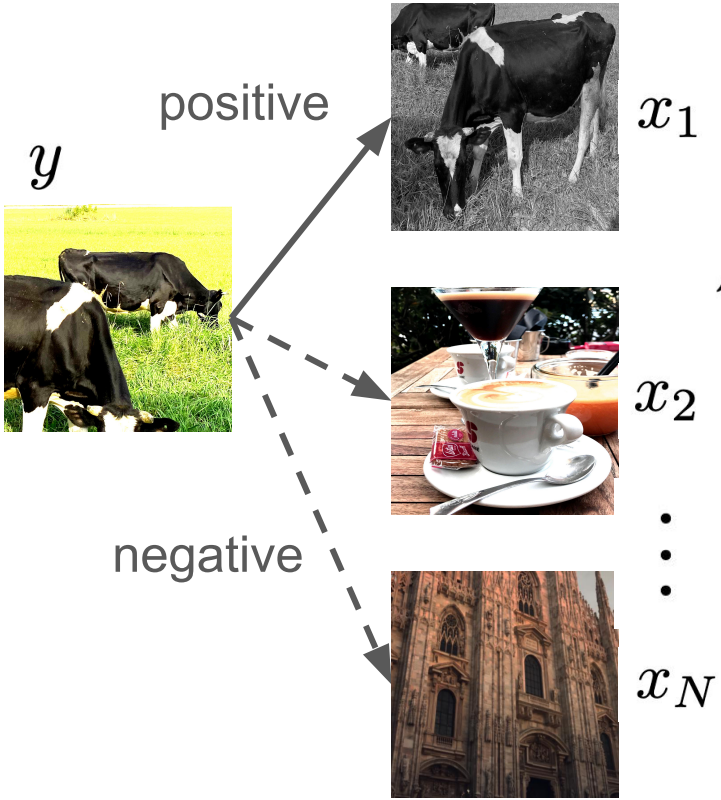
Y

$$I\Big(f_\theta(X), f_\theta(Y)\Big) \to \max_\theta$$

f_θ – our neural network with weights θ

$$I(X;Y) = \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy$$

InfoNCE loss and negative examples



$$\mathcal{L}_{NCE}(\theta) = \mathbb{E}_{p(x_{1:N},y)}\left[-\log \frac{e^{f_\theta(x_1,y)}}{\sum_{n=1}^{N} e^{f_\theta(x_n,y)}} \right] \to \min_\theta$$

Noise **C**ontrastive **E**stimation

Oord et al., 2018

InfoNCE is a lower bound for mutual information

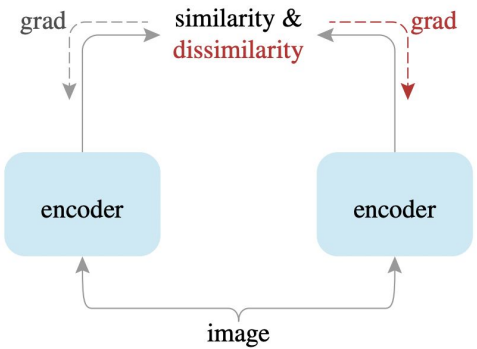
$$I(X_1; Y) \geq \log N - \mathcal{L}_{NCE}$$

$$\mathcal{L}_{NCE}(\theta) = \mathbb{E}_{p(x_{1:N}, y)} \left[-\log \frac{e^{f_\theta(x_1, y)}}{\sum_{n=1}^{N} e^{f_\theta(x_n, y)}} \right]$$

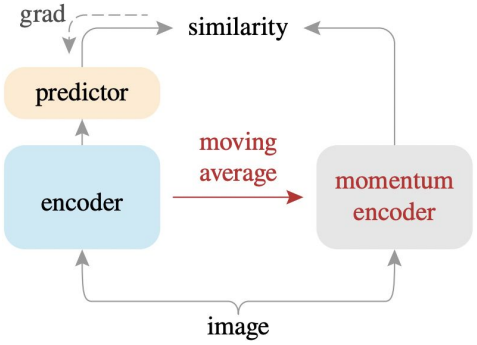
Oord et al., 2018 Poole et al., 2019

Contrastive learning for images

	With negative examples	No negative examples
Single (siamese) encoder	SimCLR, Chen et al., 2020	SimSiam, Chen and He, 2020
Momentum encoder	MoCo, He et al., 2019	BYOL, Grill et al., 2020



SimCLR

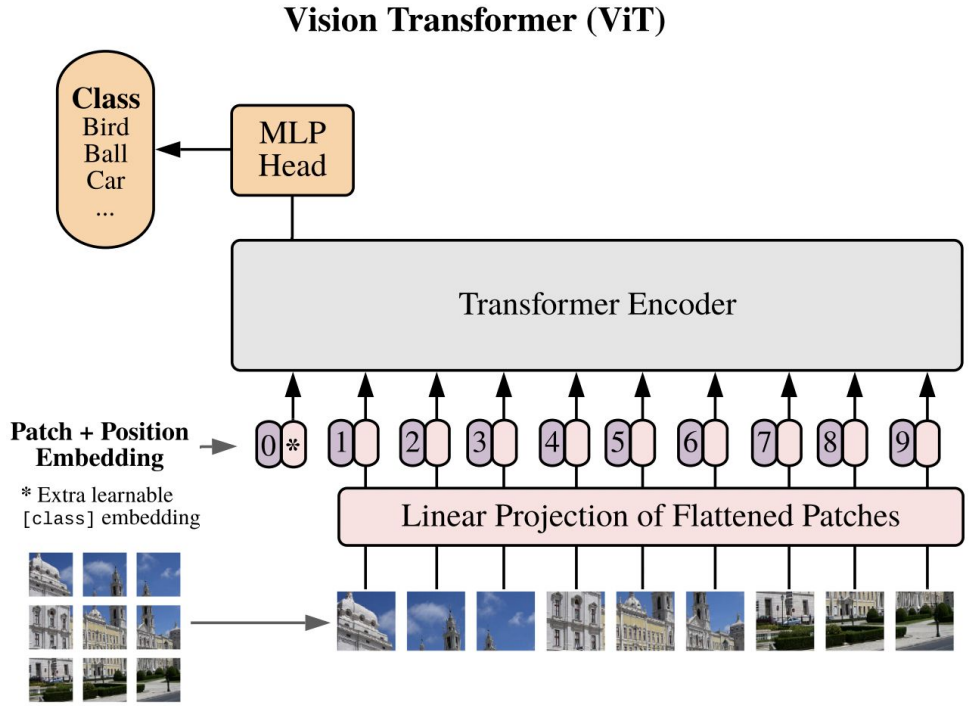


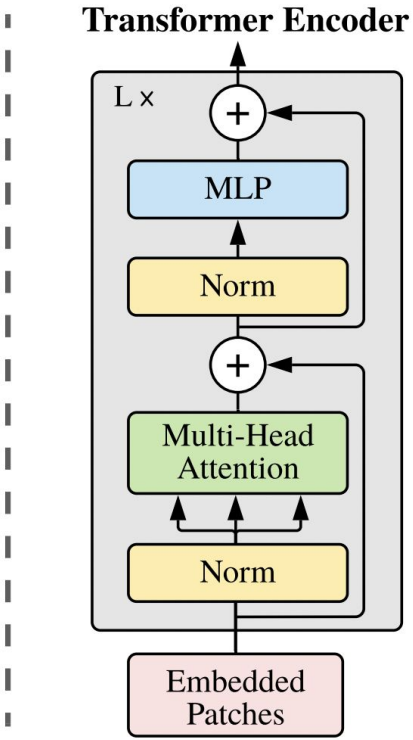
BYOL

- Clustering-based (SwaV, Caron et al., 2020)

Transformers for vision

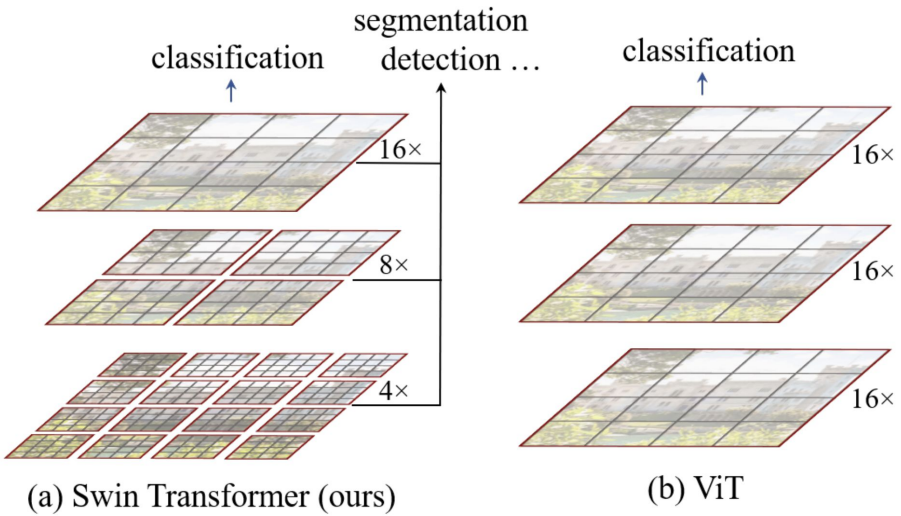
Vision Transformer (ViT)



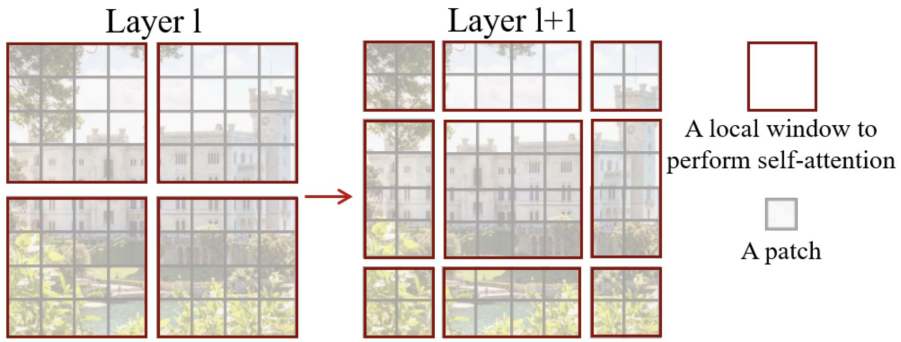


Dosovitskiy et al., 2021

Swin (**S**hifted **win**dows) Transformer



hierarchical attention mapping



shifted windows

[Liu et al., 2021]

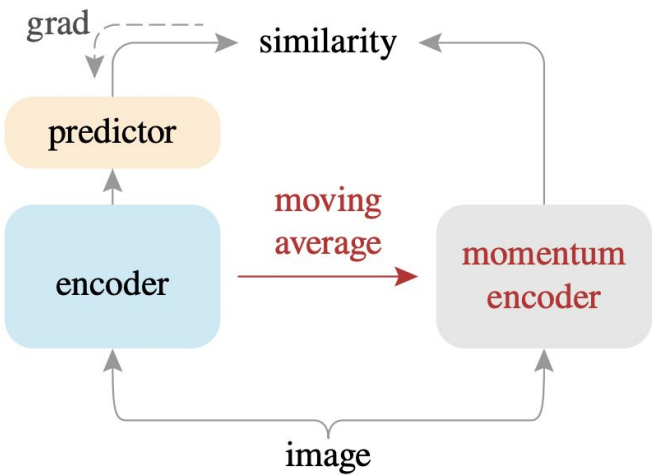
Emerging Properties in Self-Supervised Vision Transformers

Meta AI

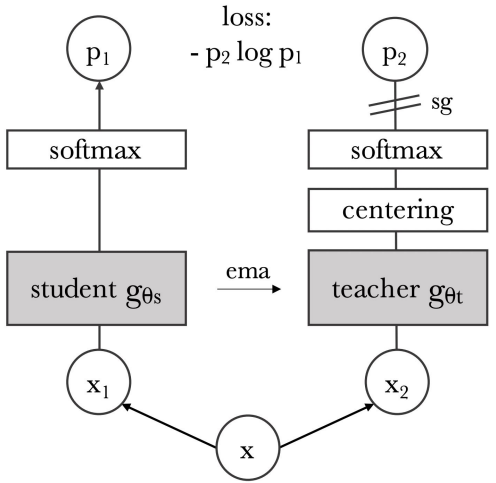
- **DINO** – self-**DI**stillation with **NO** labels (contrastive w/o negative examples)
- Self-supervised ViT features contain explicit information about **semantic segmentation** (opposed to supervised ViTs and ConvNets)
- Self-supervised ViT features are **excellent k-NN classifiers**

Caron et al., 2021

DINO: scheme



BYOL



DINO

[Caron et al., 2021]

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
Comparison across architectures					
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

Caron et al., 2021

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].

			$\mathcal{R}$Ox		$\mathcal{R}$Par	
Pretrain	Arch.	Pretrain	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**

Caron et al., 2021

DINO: attention maps to segmentation masks

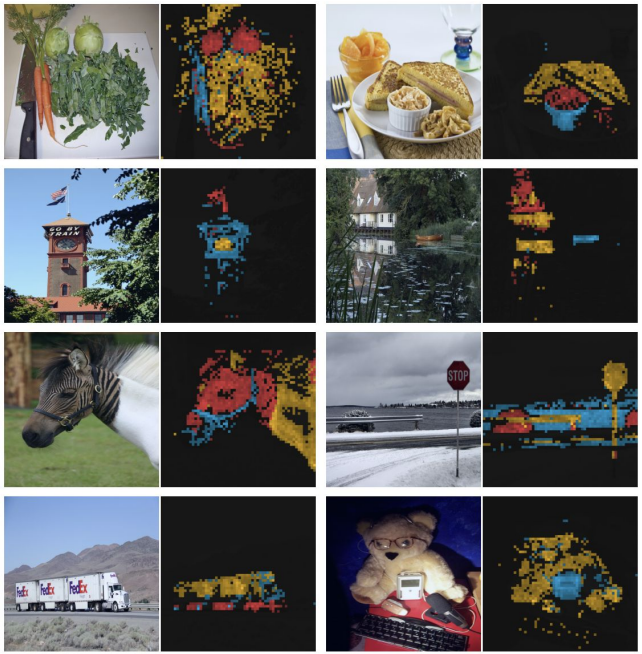
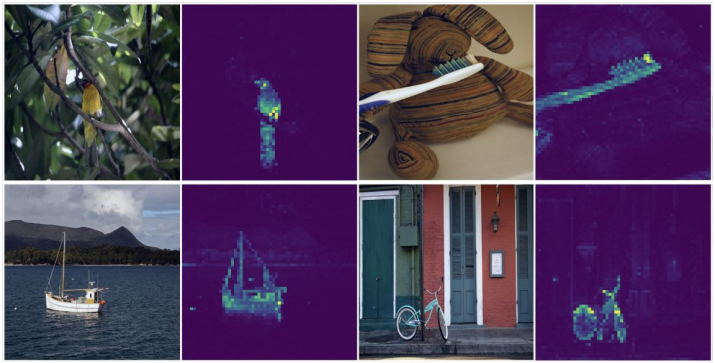
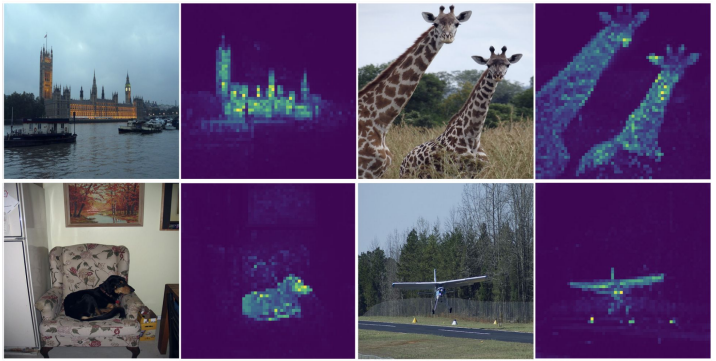


Figure 3: **Attention maps from multiple heads.** We consider the heads from the last layer of a ViT-S/8 trained with DINO and display the self-attention for `[CLS]` token query. Different heads, materialized by different colors, focus on different locations that represents different objects or parts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





Caron et al., 2021

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

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.

Supervised

DINO

Caron et al., 2021

Masked Image Modeling (MIM)

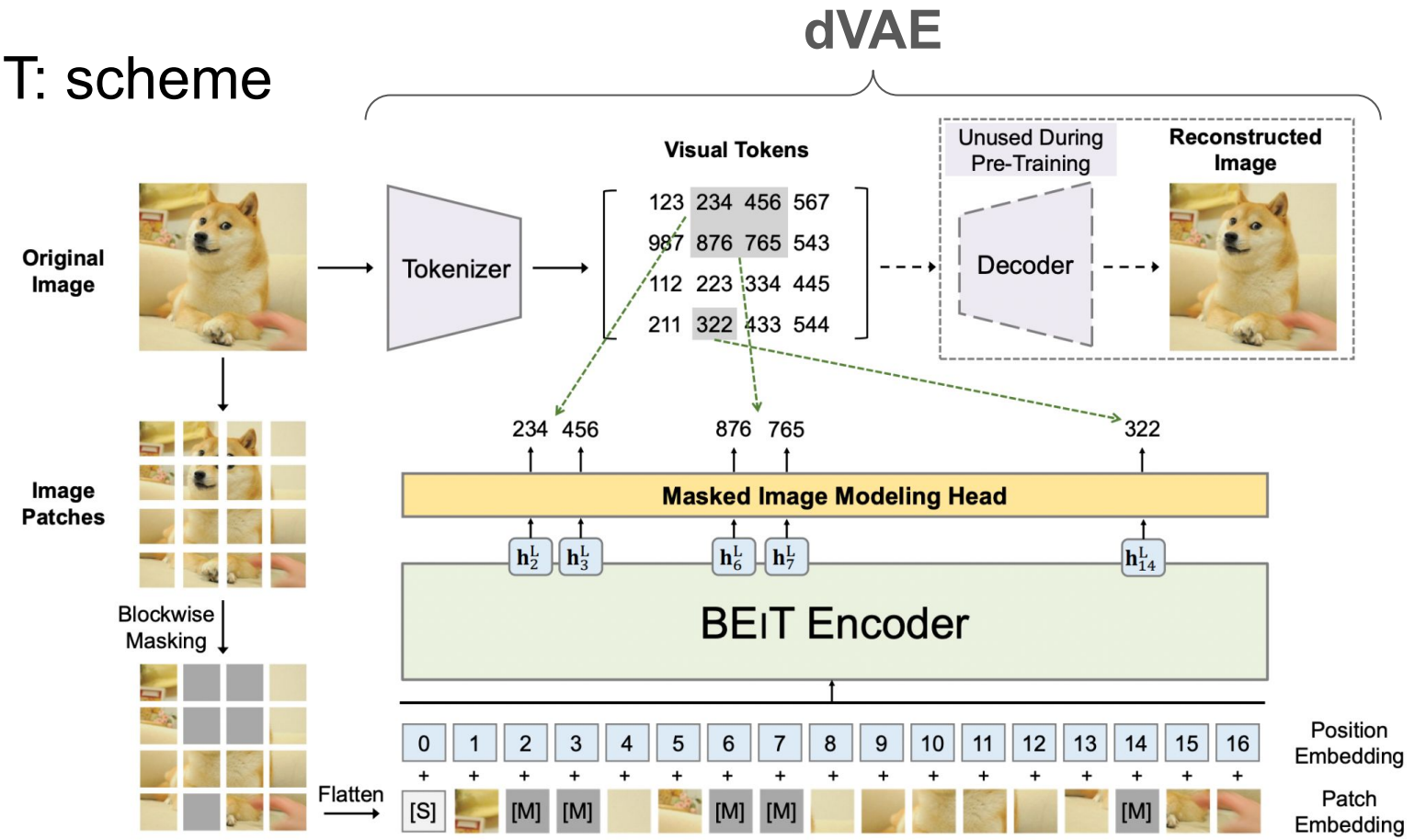
BEiT

Bidirectional **E**ncoder representation from **I**mage **T**ransformers
Microsoft Research

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

Bao et al., 2021

BEiT: scheme



Bao et al., 2021

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
  $s \leftarrow \text{Rand}(16, 0.4N - |\mathcal{M}|)$ ▷ *Block size*
  $r \leftarrow \text{Rand}(0.3, \frac{1}{0.3})$ ▷ *Aspect ratio of block*
  $a \leftarrow \sqrt{s \cdot r}; b \leftarrow \sqrt{s/r}$
  $t \leftarrow \text{Rand}(0, h - a)$; $l \leftarrow \text{Rand}(0, w - b)$
  $\mathcal{M} \leftarrow \mathcal{M} \bigcup \{(i, j) : i \in [t, t + a), j \in [l, l + b)\}$
until $|\mathcal{M}| > 0.4N$ ▷ *Masking ratio is* 40%
return $\mathcal{M}$

Bao et al., 2021

BEiT: results

Models	Model Size	Resolution	ImageNet
Training from scratch (i.e., random initialization)			
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
Supervised Pre-Training on ImageNet-22K (using labeled data)			
ViT$_{384}$-B [DBK+20]	86M	384^2	84.0
ViT$_{384}$-L [DBK+20]	307M	384^2	85.2
Self-Supervised Pre-Training on ImageNet-1K (without labeled data)			
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. ‡: ViT$_{384}$-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.

Bao et al., 2021

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.

Bao et al., 2021

BEiT: attention maps

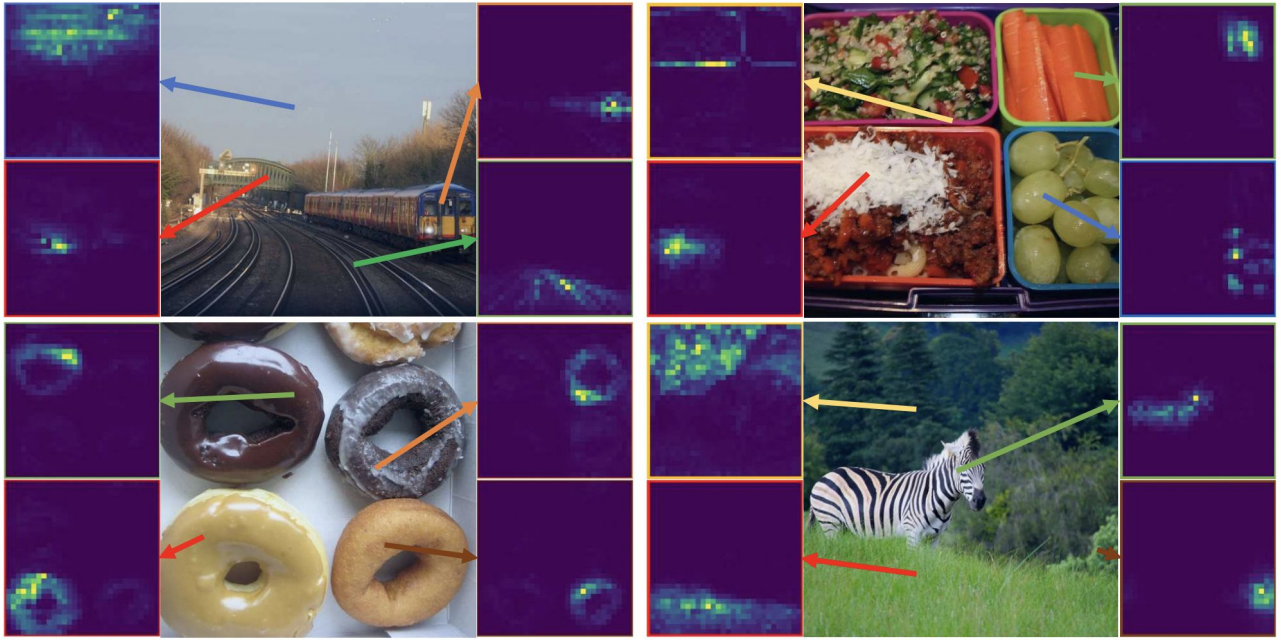


Figure 2: Self-attention map for different reference points. The self-attention mechanism in BEIT is able to separate objects, although self-supervised pre-training does not use manual annotations.

Bao et al., 2021

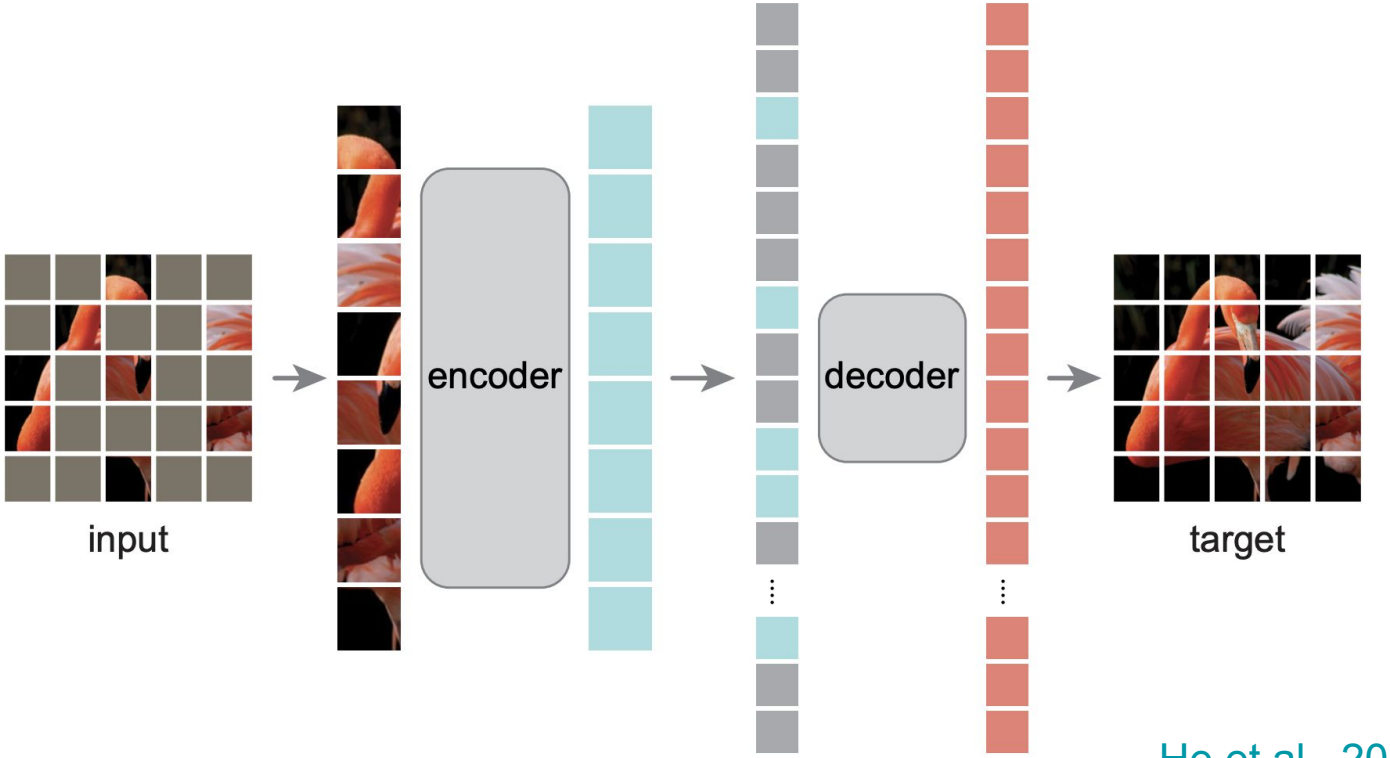
MAE

Masked **A**uto**E**ncoders Are Scalable Vision Learners, Meta AI

- 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

He et al., 2021

MAE: scheme



He et al., 2021

MAE: reconstruction

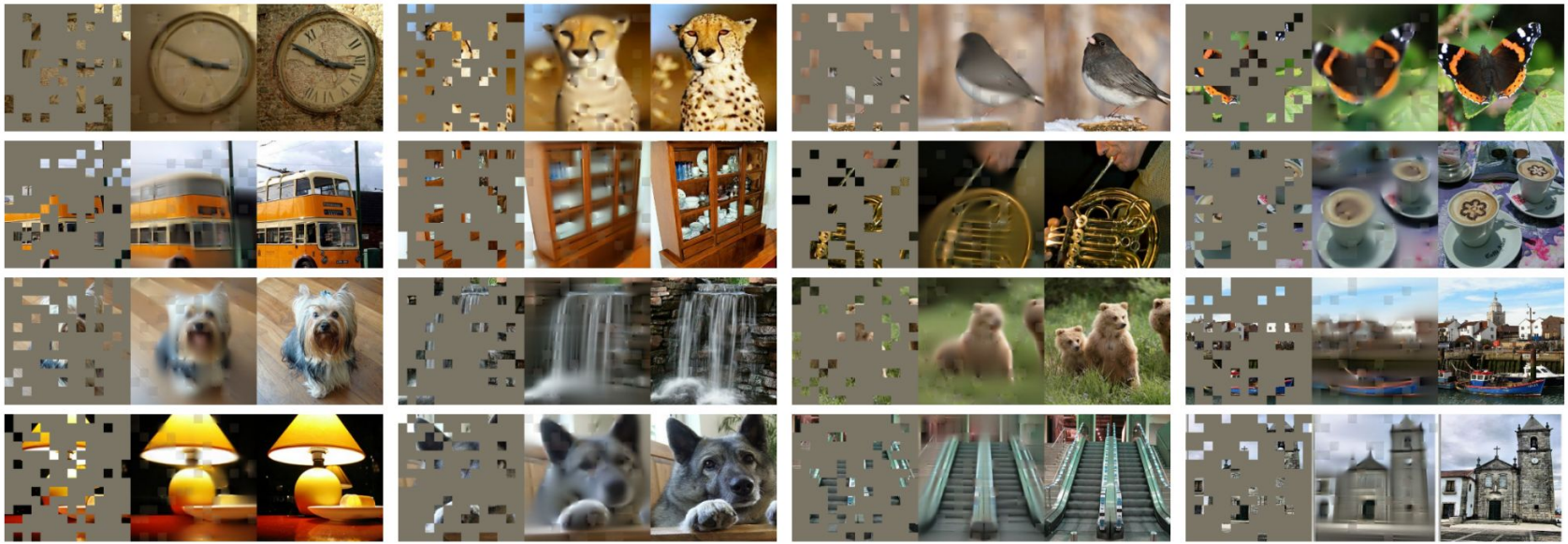
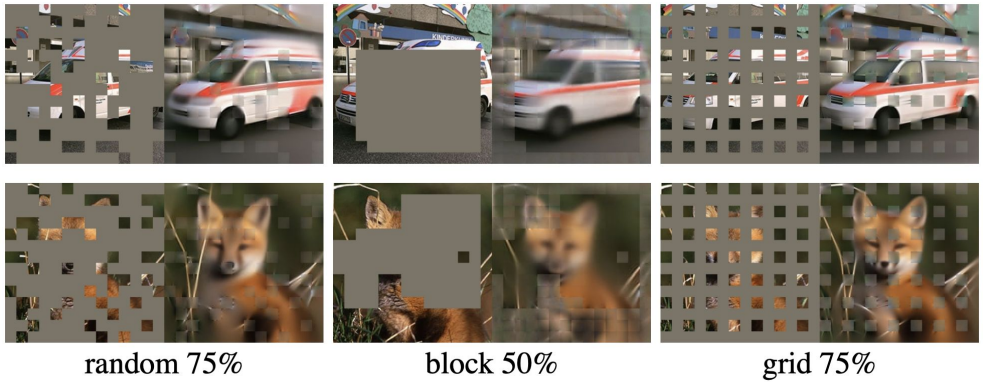


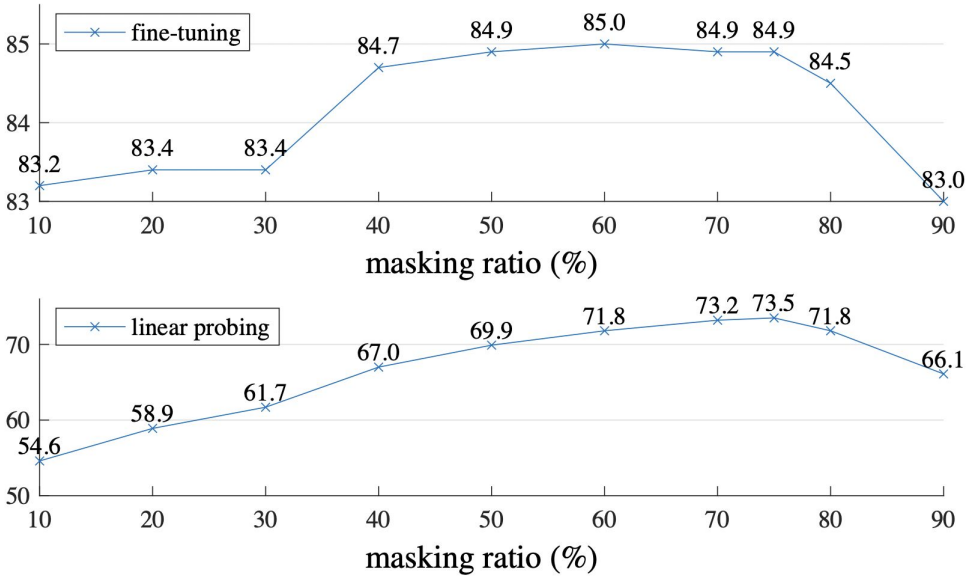
Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction† (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix.

†*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.*

He et al., 2021

MAE: masking





He et al., 2021

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.

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	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	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.

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	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

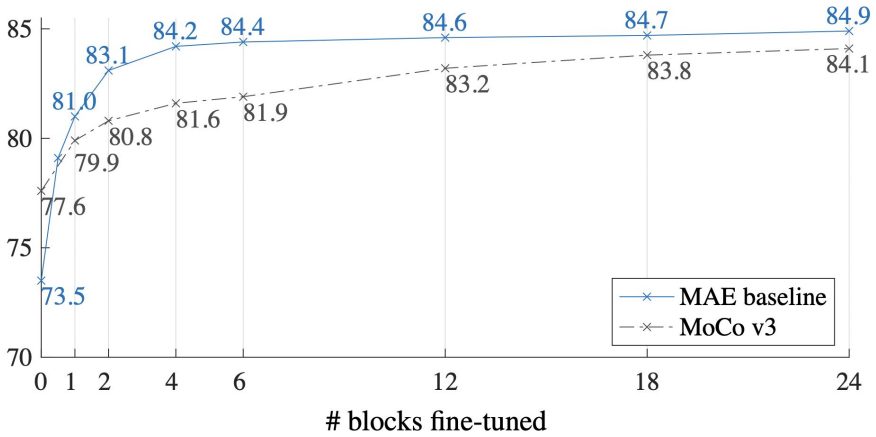
(f) **Mask sampling**. Random sampling works the best. See Figure 6 for visualizations.

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.

He et al., 2021

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	83.6	85.9	86.9	**87.8**



		AP^{box}		AP^{mask}	
method	pre-train data	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.

He et al., 2021

SimMIM

A Simple Framework for Masked Image Modeling, Microsoft Research

- Parallel work with MAE (both accepted to CVPR 2022)
- Takes all patches as inputs (even masked, contrary to MAE)
- Works with hierarchical transformers, e.g. Swin (advantage over MAE)

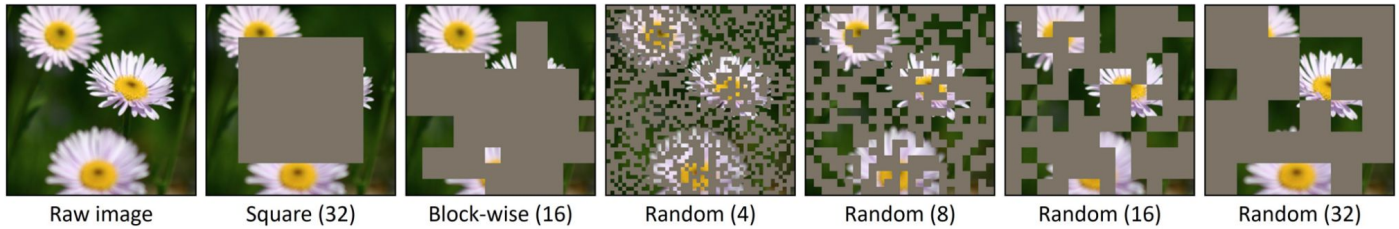


Figure 2. Illustration of masking area generated by different masking strategies using a same mask ratio of 0.6: square masking [38], block-wise masking [1] apply on 16-sized patches, and our simple random masking strategy on different patch sizes (e.g., 4, 8, 16 and 32).

Xie et al., 2021

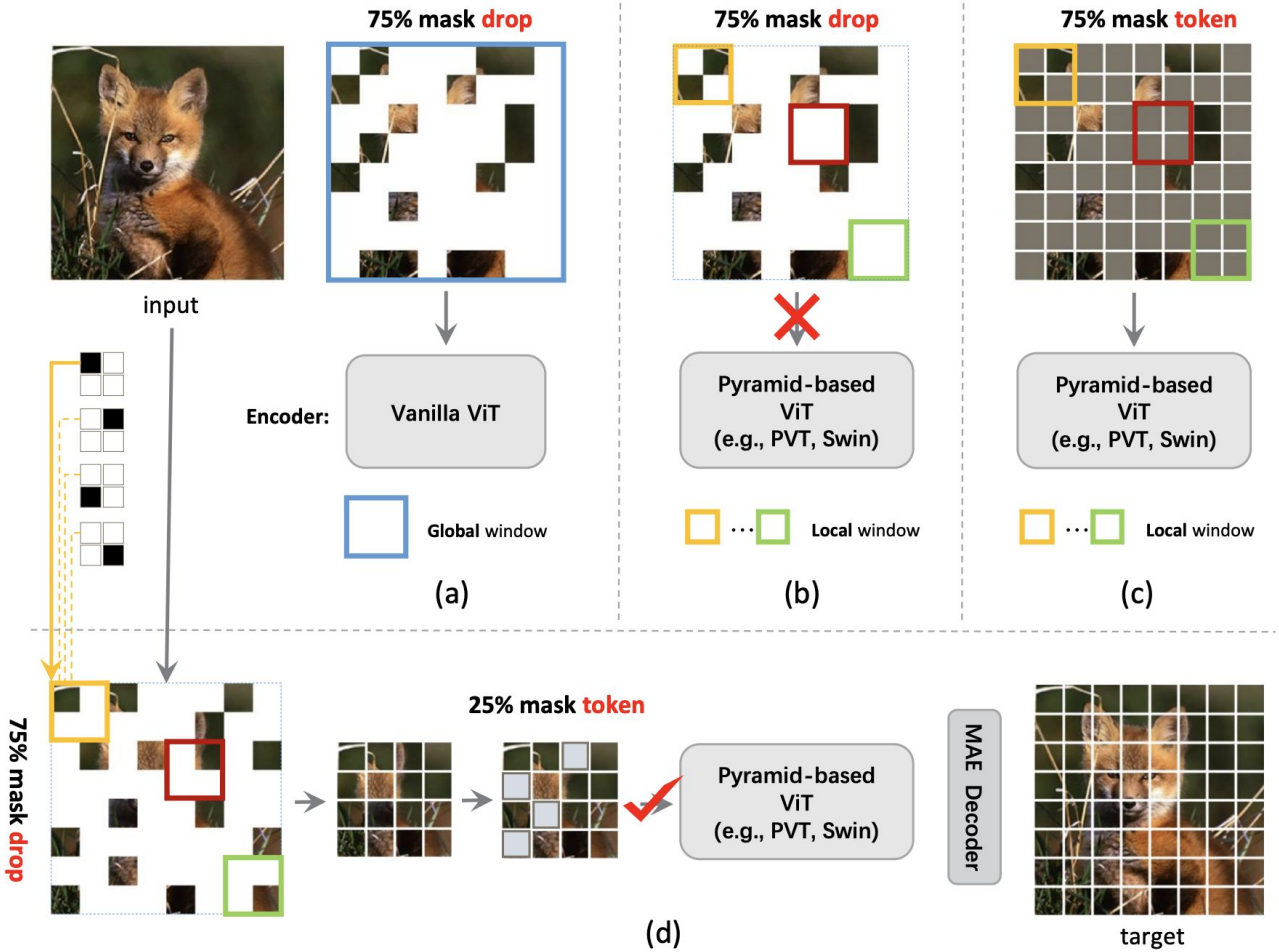
UM-MAE

Uniform-masking MAE

- Follow-up for MAE adapting it to hierarchical transformers
- Two staged masking strategy
- Preserving main features of MAE: processing only non-masked patches with encoder

Li et al., 2022

UM-MAE



Li et al., 2022

Uniform Sampling	Secondary Masking	MAE-style Pre-training

UM-MAE & Swin

Li et al., 2022

UM-MAE: masking strategies

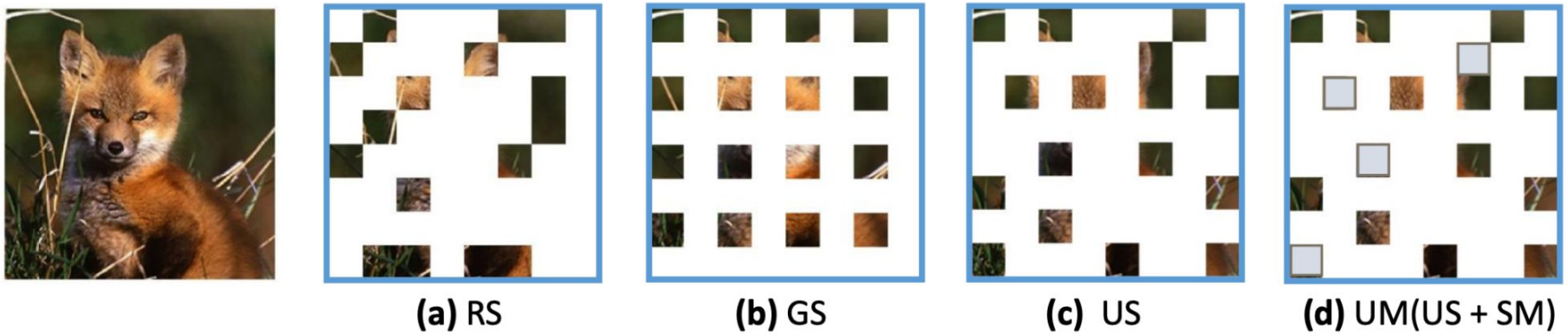


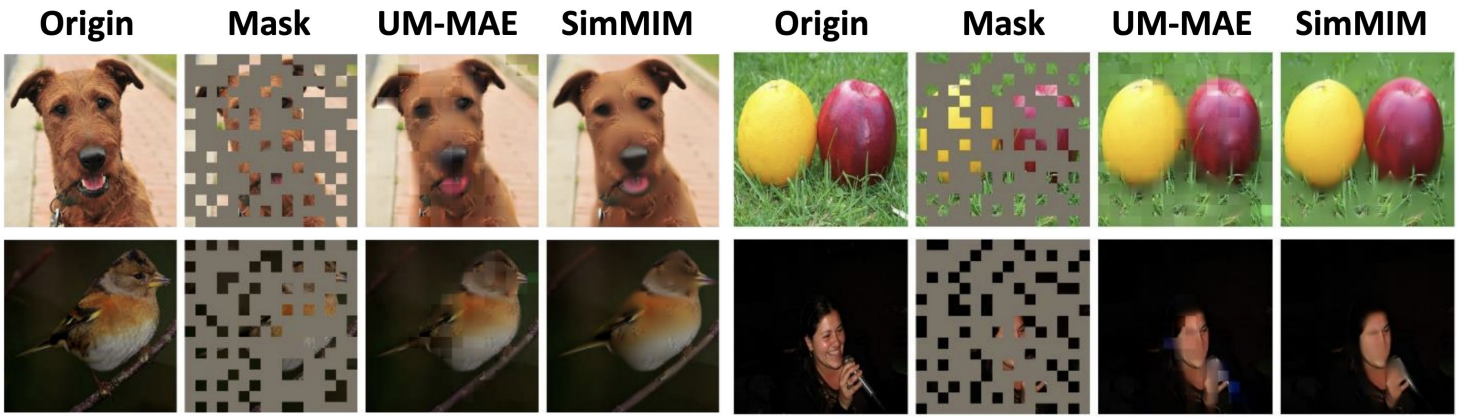
Figure 4: **Different sampling strategy with a sample ratio of** 25%. (a) Random Sampling (RS) in MAE [19]; (b) Grid-wise Sampling (GS) in MAE; and the proposed (c) Uniform Sampling (US); (d) Uniform Masking (UM) that includes US and Secondary Masking (SM).

Sampling Strategy (25%)	Pyramid Support	SM Ratio	Pre-train Loss	ImageNet-1K	ADE20K		COCO		
				Top-1 Acc	mIoU	aAcc	AP	AP_{50}	AP_{75}
(a) RS (MAE [19] Baseline)	×	–	0.4256	**82.88**	**42.54**	**80.85**	**46.0**	**64.7**	**49.8**
(b) GS	✓	–	0.3682	82.48	38.79	79.16	44.4	63.2	48.6
(c) US (Ours)	✓	–	0.3858	82.74	41.55	80.48	45.5	64.2	49.6
(d) UM (Ours)	✓	15%	0.4171	82.75	41.68	80.54	45.8	**64.6**	49.8
	✓	25%	0.4395	**82.88**	**42.59**	**80.80**	**45.9**	64.5	**50.2**
	✓	35%	0.4645	82.68	42.02	80.72	**45.9**	**64.6**	50.1

Li et al., 2022

UM-MAE: results & performance

Architecture	Method	Pre-train (200 epoch)		Fine-tune (/Scratch) Performance		
		Time	Memory	ImageNet-1K	ADE20K	COCO
PVT-S [37]	Supervised from Scratch (Baseline)			77.84	40.38	42.3
	SimMIM [42]	38.0 h	20.6 GB	79.28 (+1.44)	**43.04 (+2.66)**	44.8 (+2.5)
	UM-MAE **(ours)**	**21.3 h**	**11.6 GB**	**79.31 (+1.47)**	43.01 (+2.63)	**45.1 (+2.8)**
Swin-T [28]	Supervised from Scratch (Baseline)			81.82	44.51	47.2
	SimMIM [42]	49.3 h	37.4 GB*	**82.20 (+0.38)**	45.35 (+0.84)	47.6 (+0.4)
	UM-MAE **(ours)**	**25.0 h**	**13.4 GB**	82.04 (+0.22)	**45.96 (+1.45)**	**47.7 (+0.5)**

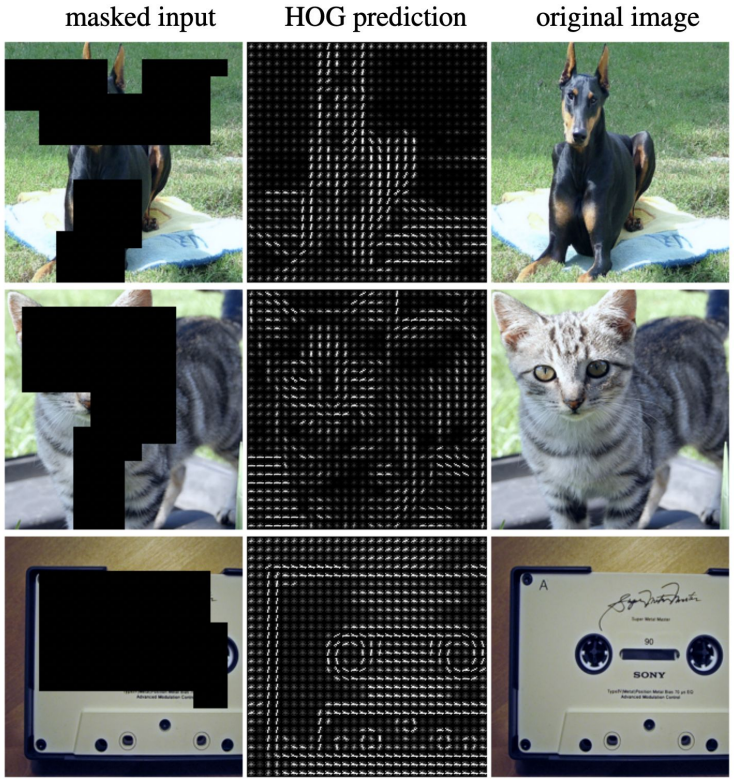


Li et al., 2022

MaskFeat

Masked **Feat**ure Prediction, Meta AI

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



Wei et al., 2022

MaskFeat: scheme & features

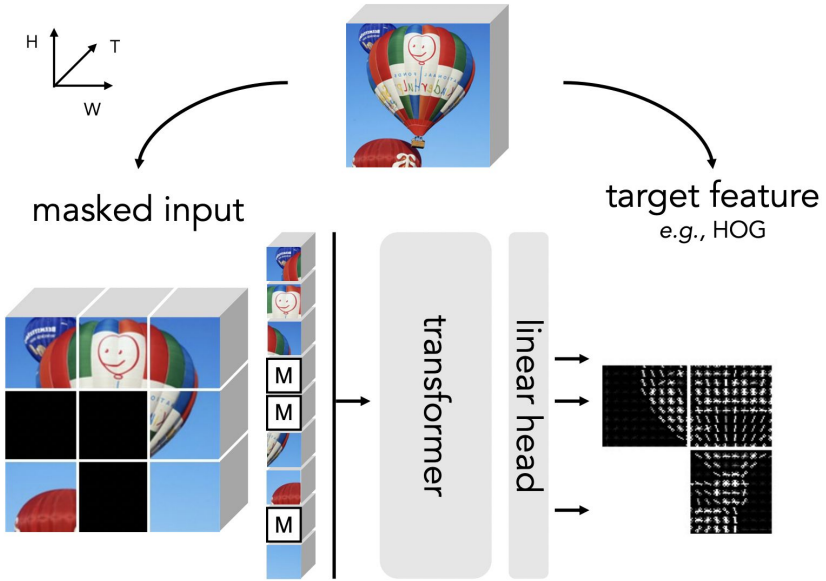


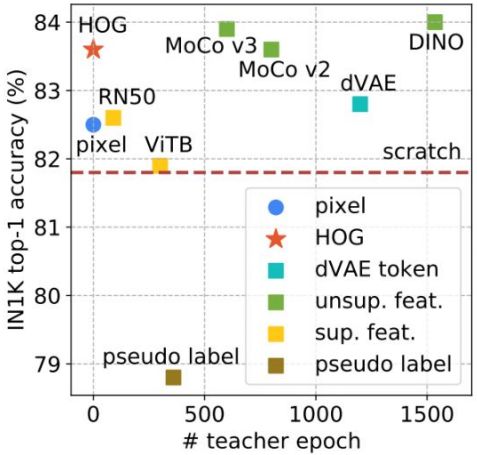
Figure 2. **MaskFeat pre-training.** We randomly replace the input space-time cubes of a video with a `[MASK]` token and directly regress features (*e.g.* HOG) of the masked regions. After pre-training, the Transformer is fine-tuned on end tasks.

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`.

Wei et al., 2022

MaskFeat: results



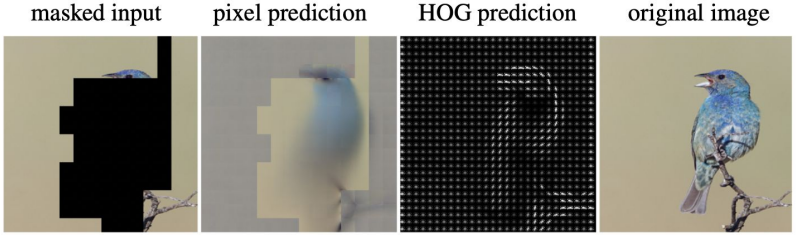
feature type	one-stage	variant	arch.	param.	epoch†	top-1
scratch	-	DeiT [63]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5
image descriptor	✓	HOG [18]	-	-	-	**83.6**
dVAE token	✗	DALL-E [58]	dVAE	54	1199	82.8
unsupervised feature	✗	MoCo v2 [15]	ResNet50	23	800	83.6
unsupervised feature	✗	MoCo v3 [17]	ViT-B	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.

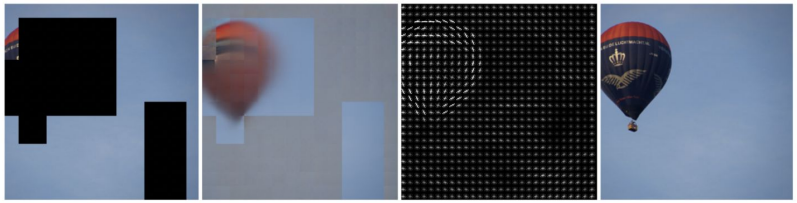
† Different teachers use different training strategies. dVAE is pre-trained on an external 250M dataset, while self-supervised methods require multi-view training. To measure the cost in a unified way, we normalize the number of epochs by the cost of one epoch on IN-1K training set with *one* 224^2 *view*.

Wei et al., 2022

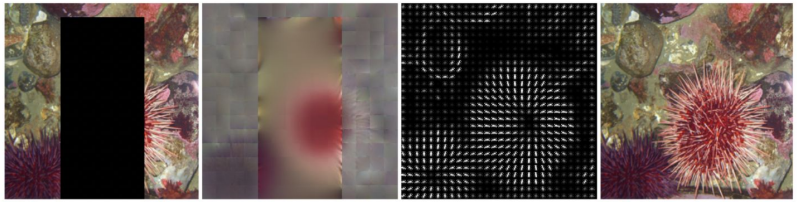
MaskFeat: results



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 supervised$_{384}$ 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 224^2 except supervised$_{384}$ at 384^2.

Wei et al., 2022