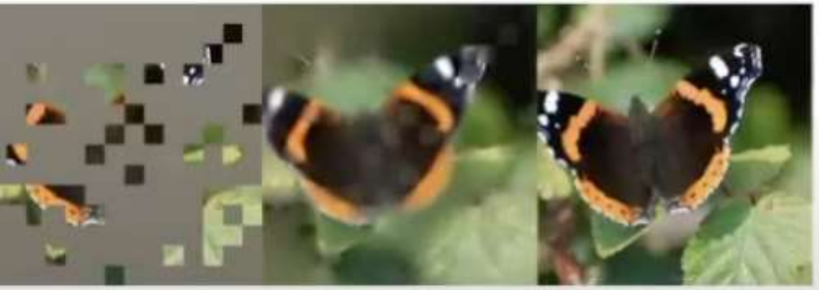
Self-supervised training for images

Severina Ekaterina

Self-supervised pre-training

generative

masked image modelling



generative pre-text tasks

- colorization
- inpainting

discriminative

discriminative pre-text tasks

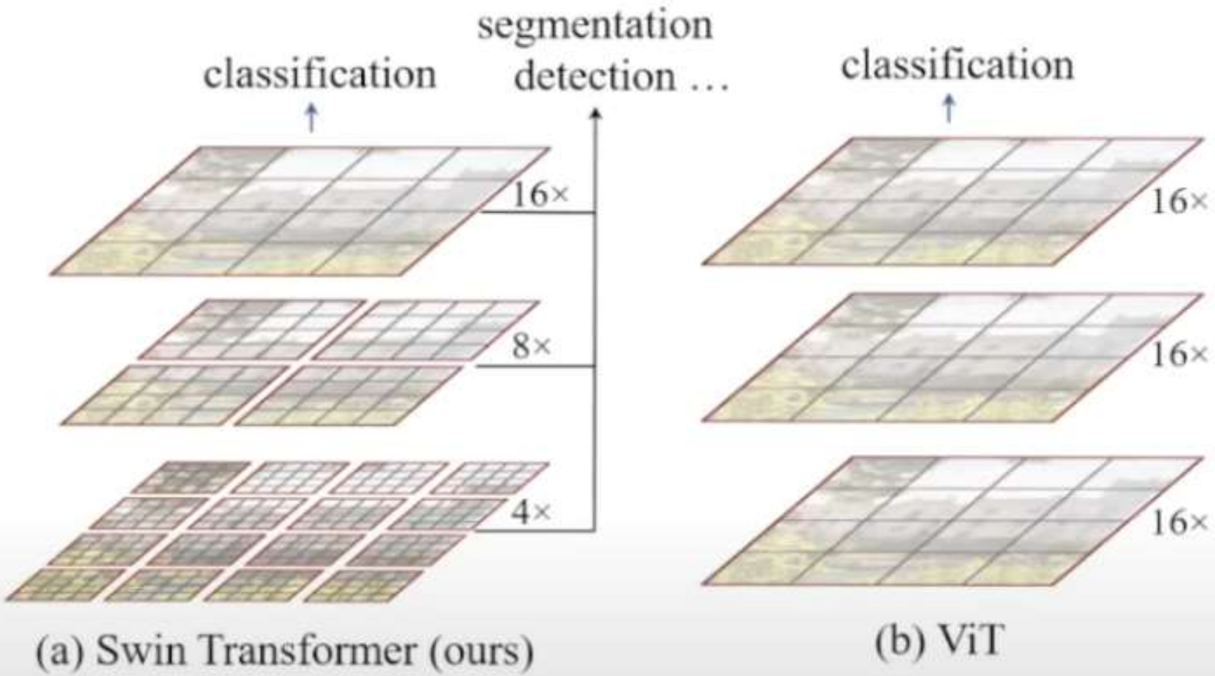
- rotation angle prediction
- jigsaw puzzle

contrastive tasks

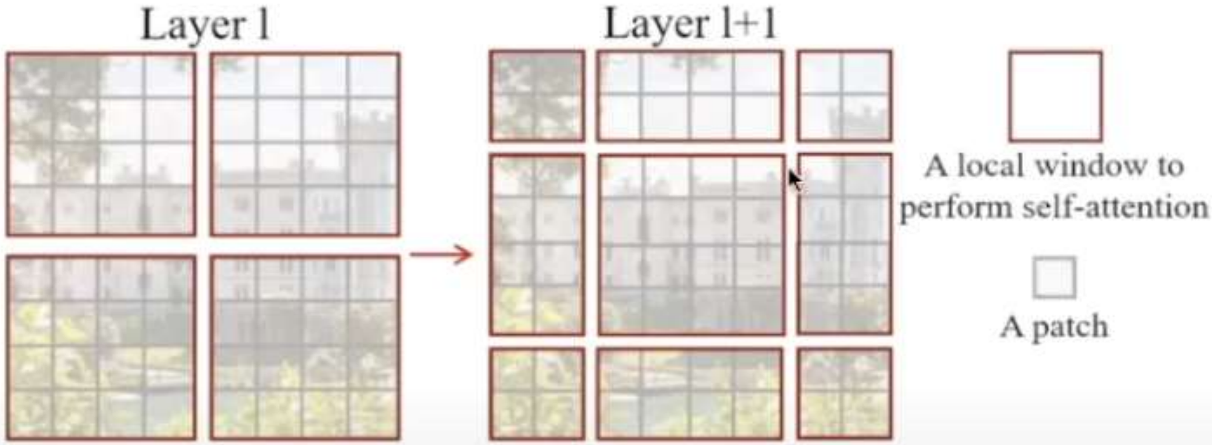
CONTENT

Вспомним!

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



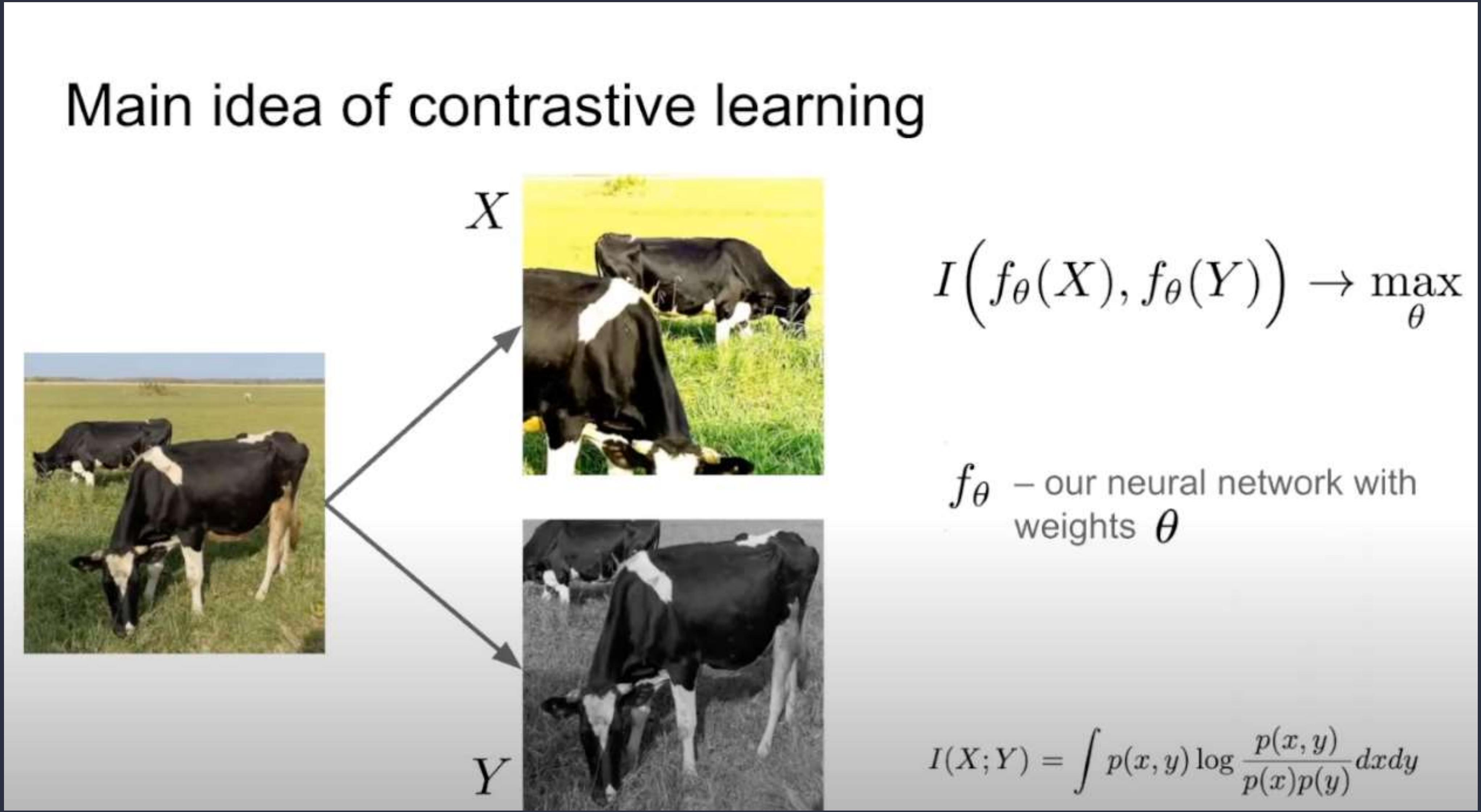
hierarchical attention mapping



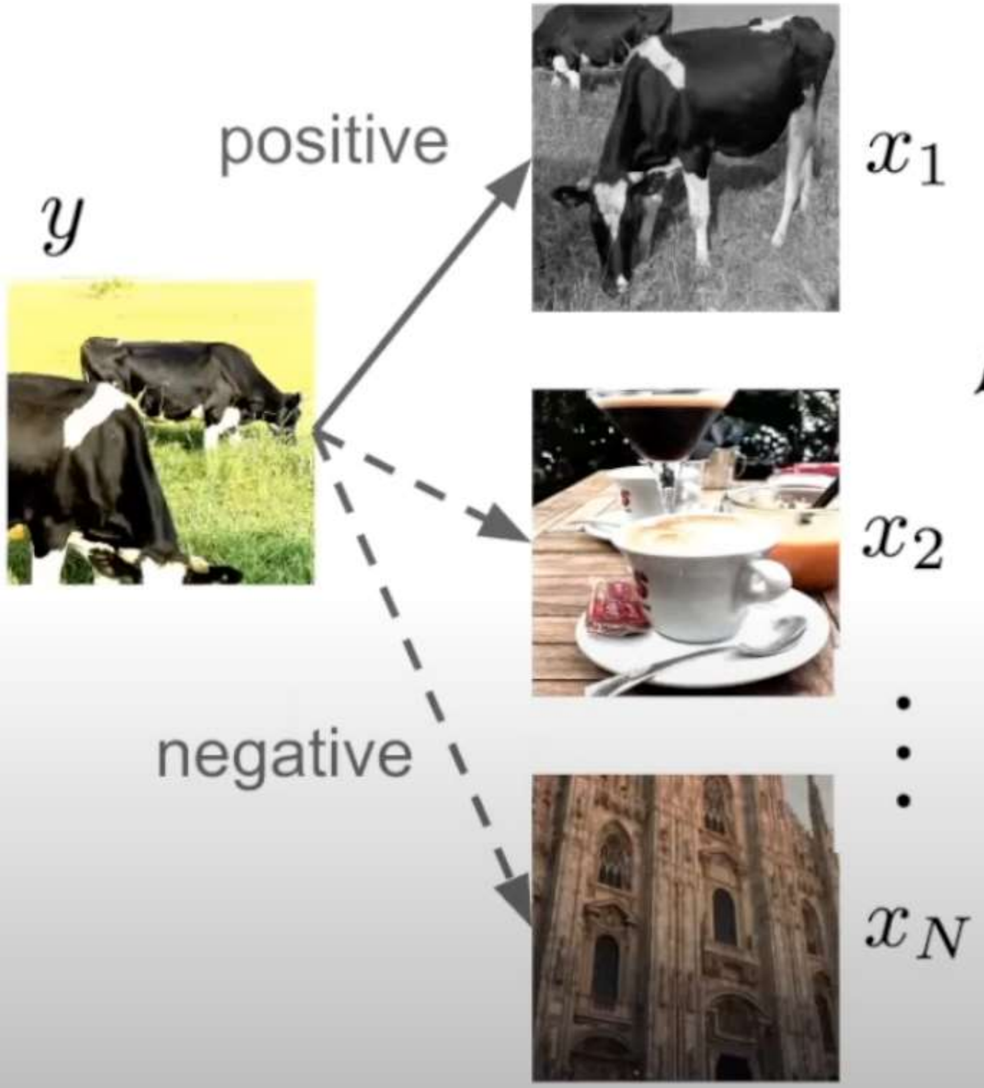
shifted windows

01 SimCLR

SimCLR - a simple framework for Contrastive learning of representation



InfoNCE loss and negative examples



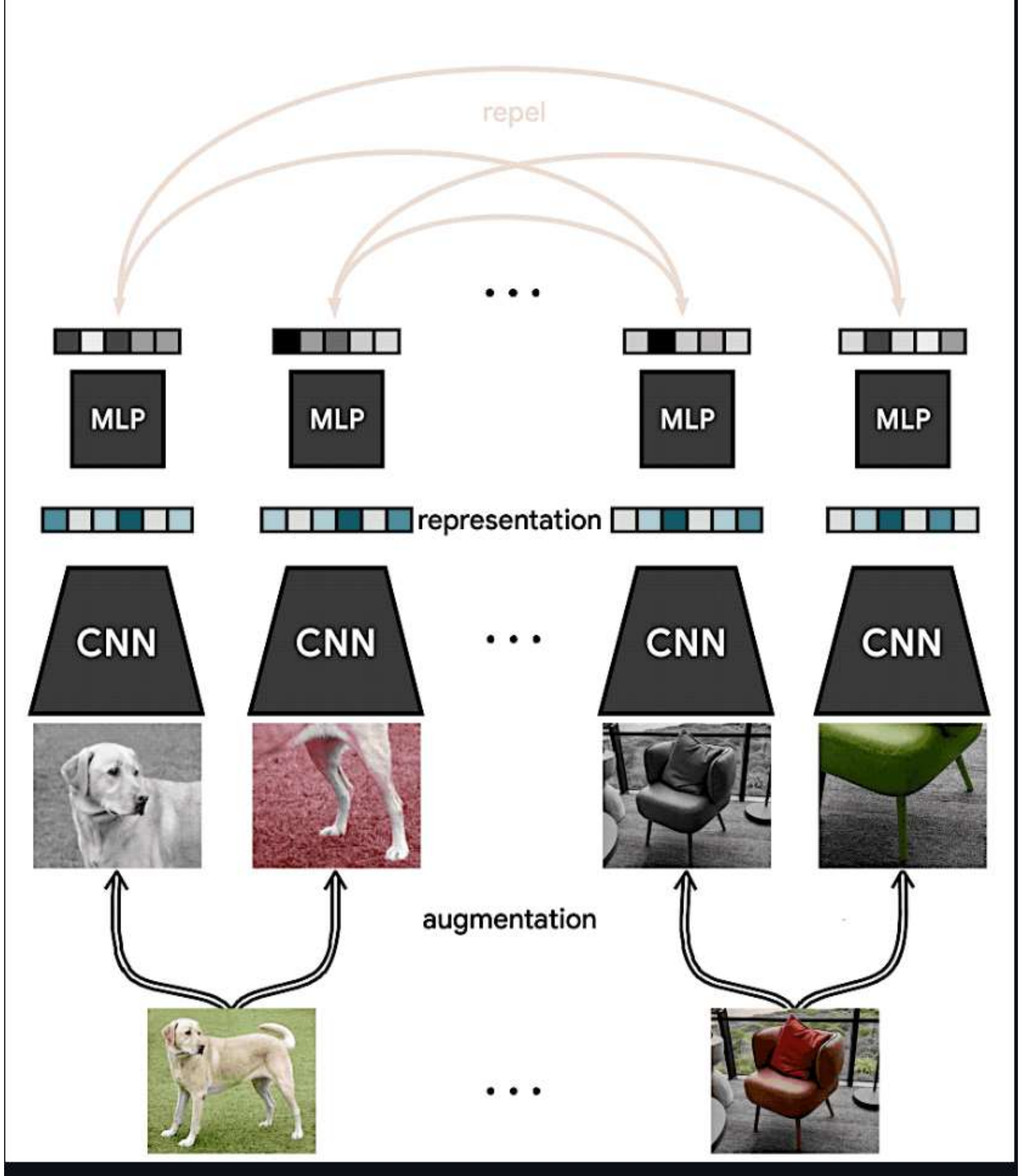
$$\mathcal{L}_{NCE}(\theta) = \mathbb{E}_{p(x_{1:N},y)}\left[-\log \frac{e^{f_\theta(x_1,y)}}{\sum_{n=1}^{N} e^{f_\theta(x_n,y)}}\right] \to \min_\theta$$

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

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

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

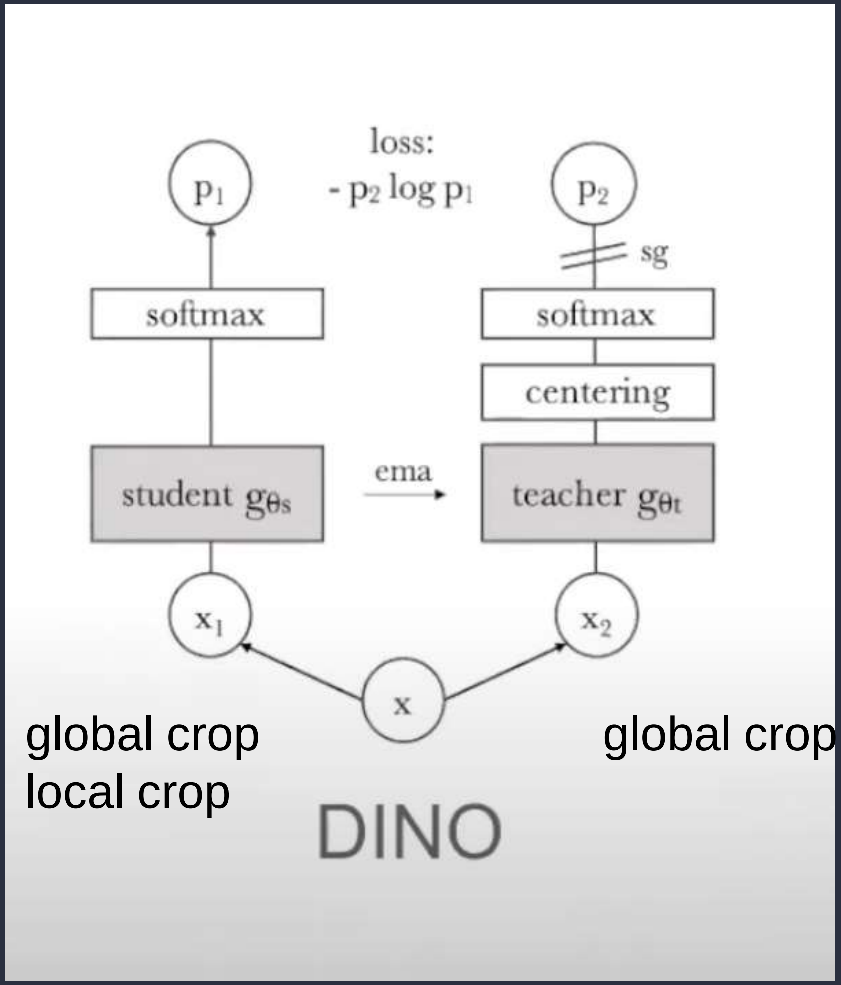
SimCLR



repel
MLP
MLP
MLP
MLP
representation
CNN
CNN
CNN
CNN
augmentation

02 DINO

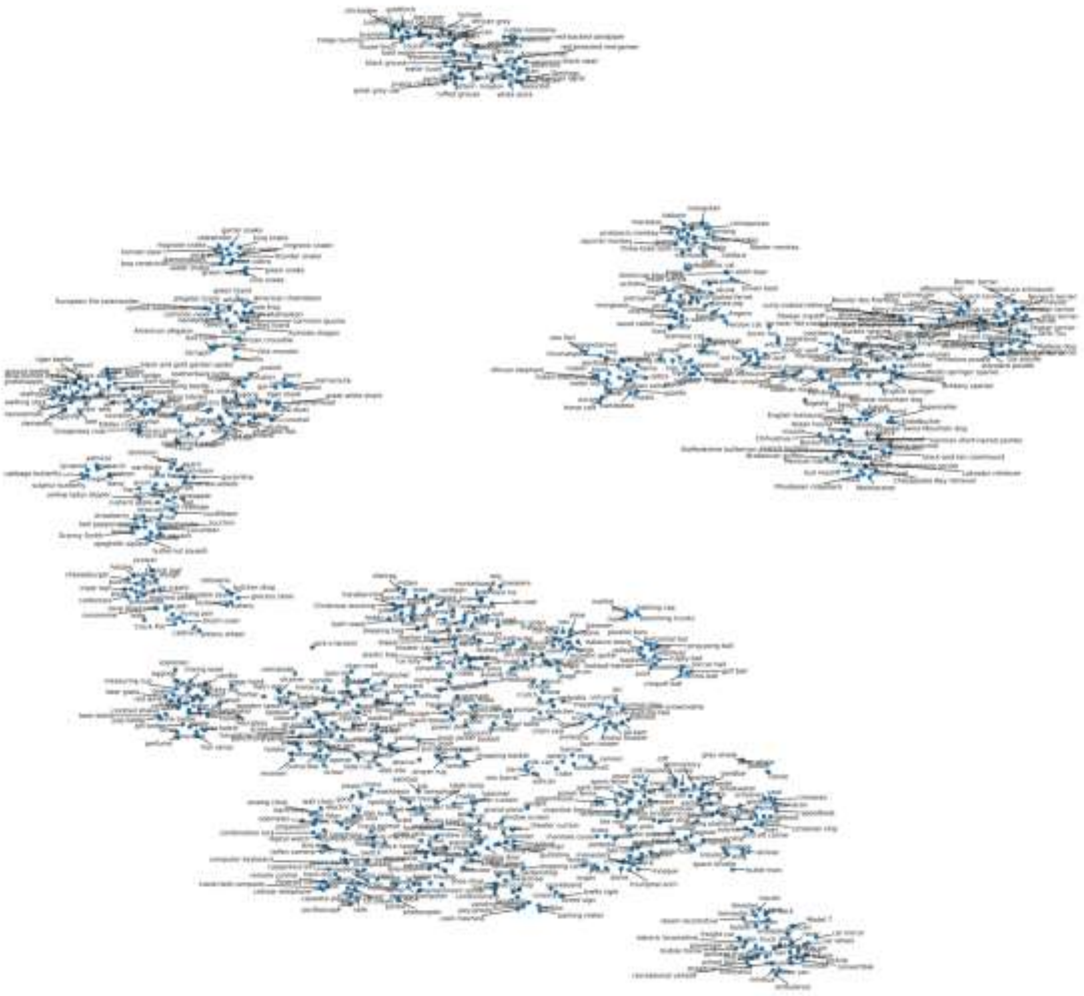
DINO - self-DIstillation with NO labels

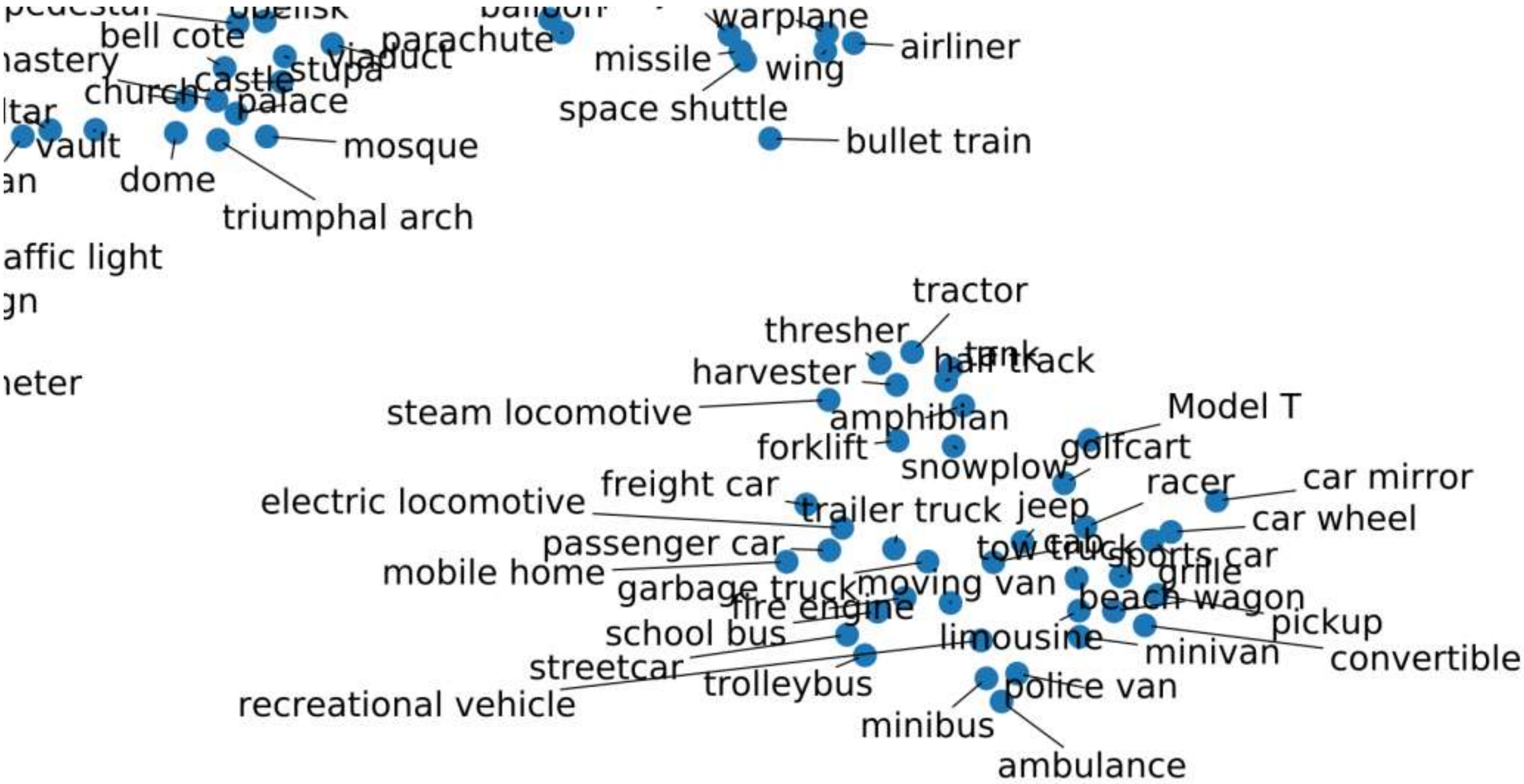


Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# 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()
```

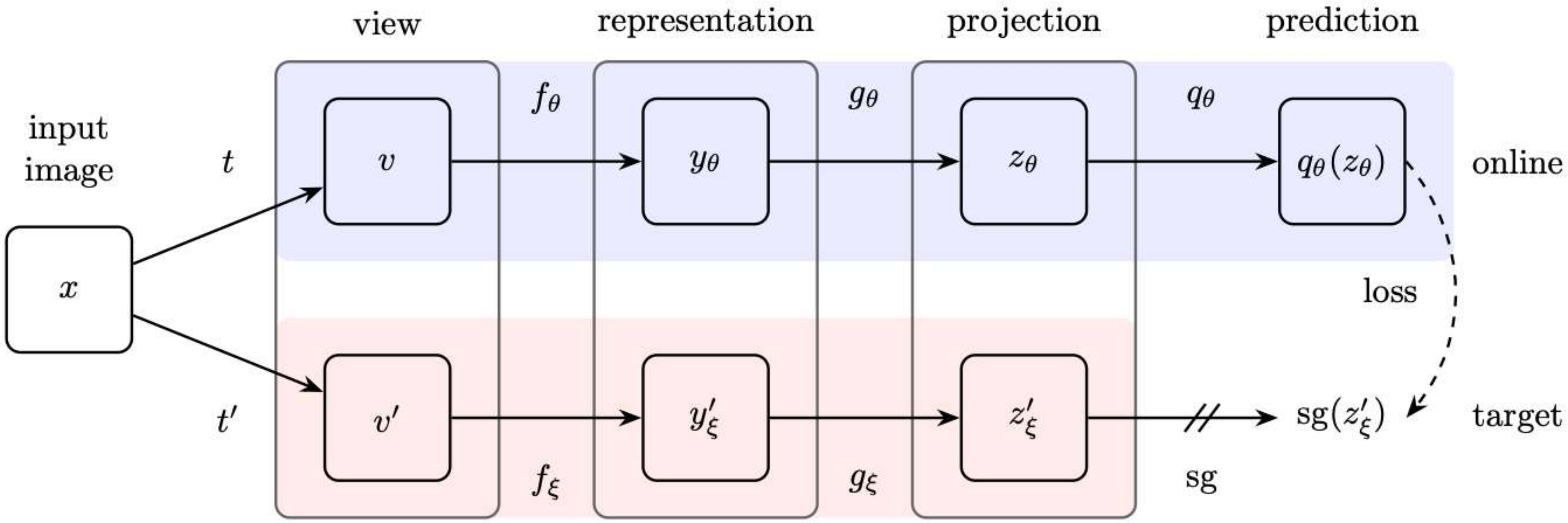
Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	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**



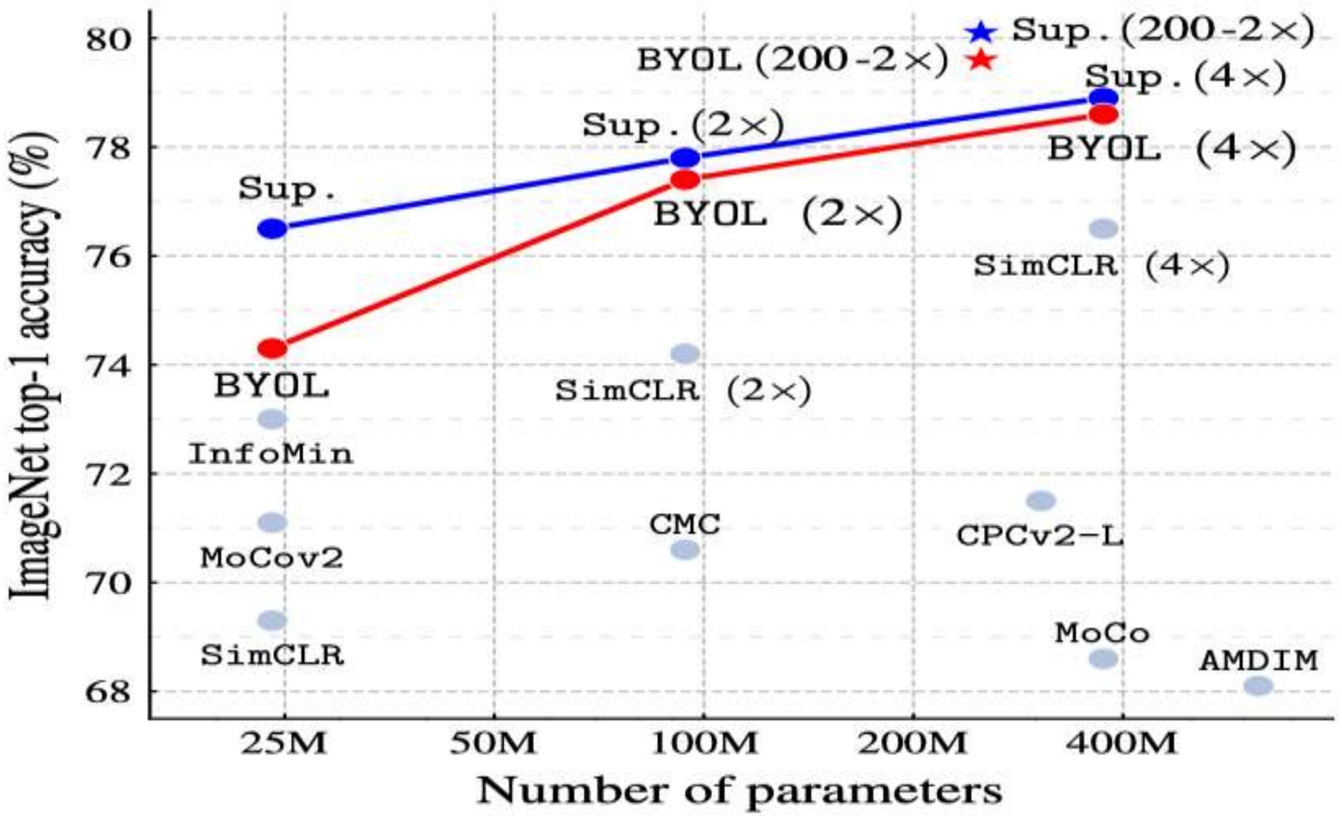


bell cote
parachute
viaduct
stupa
castle
church
palace
vault
mosque
dome
triumphal arch
warplane
airliner
missile
wing
space shuttle
bullet train
tractor
thresher
tank
half track
harvester
steam locomotive
amphibian
forklift
snowplow
Model T
golfcart
racer
car mirror
freight car
electric locomotive
trailer truck
jeep
car wheel
cab
passenger car
tow truck
sports car
mobile home
grille
garbage truck
moving van
beach wagon
fire engine
pickup
school bus
limousine
minivan
convertible
streetcar
trolleybus
police van
recreational vehicle
minibus
ambulance

03 BYOL



$$\mathcal{L}_{\theta,\xi} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2$$

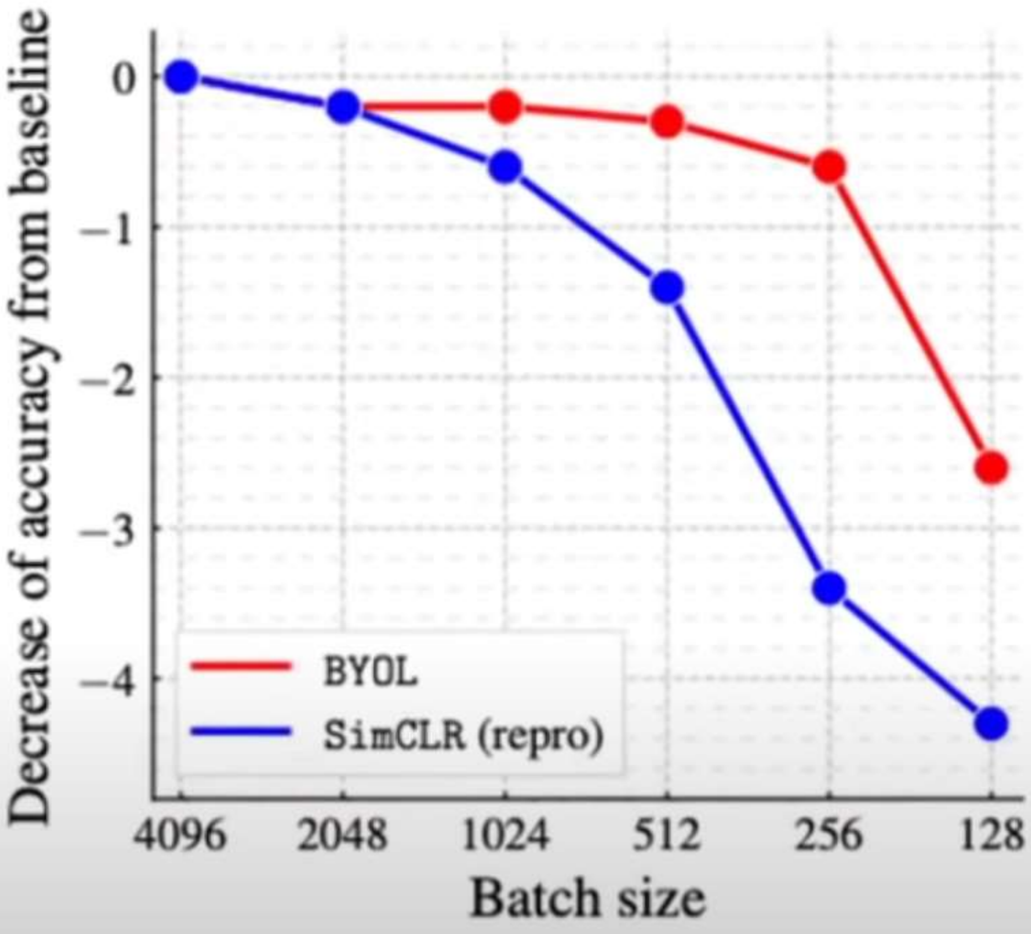


Method	Top-1		Top-5	
	1%	10%	1%	10%
Supervised [77]	25.4	56.4	48.4	80.4
InstDisc	-	-	39.2	77.4
PIRL [35]	-	-	57.2	83.8
SimCLR [8]	48.3	65.6	75.5	87.8
BYOL (ours)	**53.2**	**68.8**	**78.4**	**89.0**

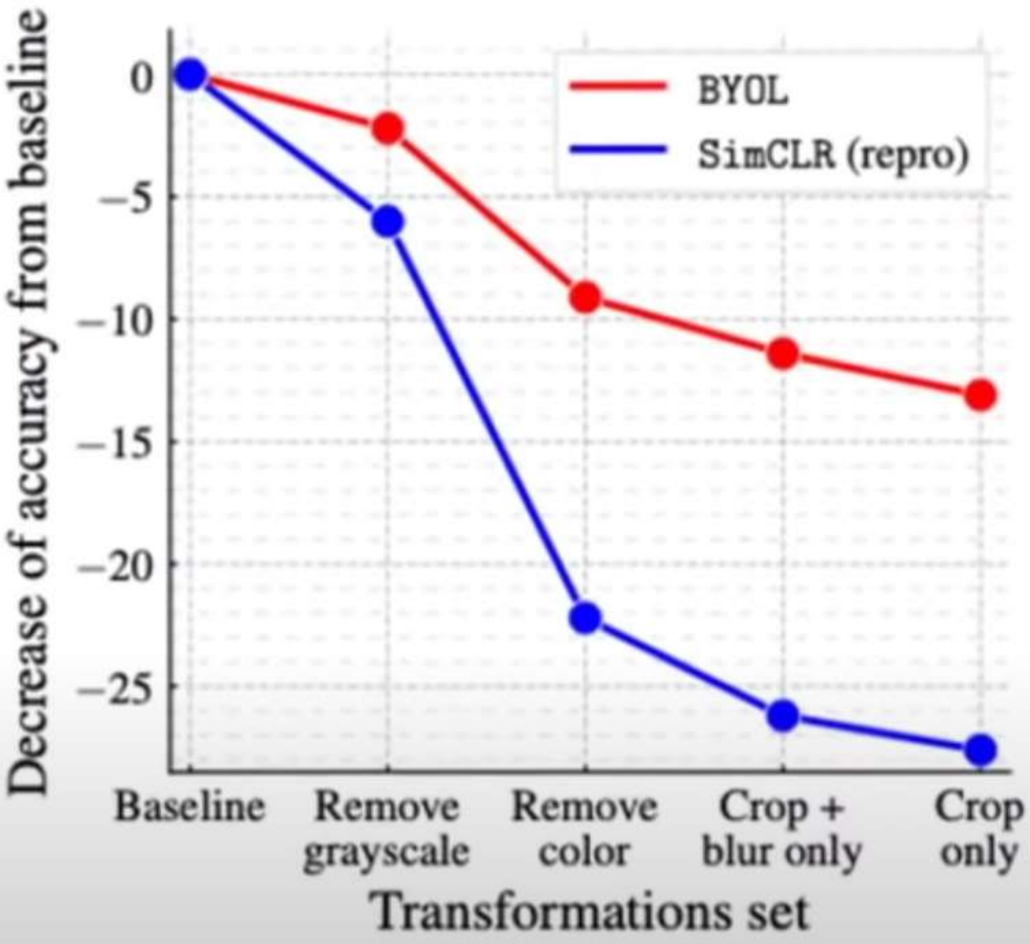
(a) ResNet-50 encoder.

Method	Architecture	Param.	Top-1		Top-5	
			1%	10%	1%	10%
CPC v2 [32]	ResNet-161	305M	-	-	77.9	91.2
SimCLR [8]	ResNet-50 (2×)	94M	58.5	71.7	83.0	91.2
BYOL (ours)	ResNet-50 (2×)	94M	62.2	73.5	84.1	91.7
SimCLR [8]	ResNet-50 (4×)	375M	63.0	74.4	85.8	92.6
BYOL (ours)	ResNet-50 (4×)	375M	69.1	75.7	87.9	92.5
BYOL (ours)	ResNet-200 (2×)	250M	**71.2**	**77.7**	**89.5**	**93.7**

(b) Other ResNet encoder architectures.



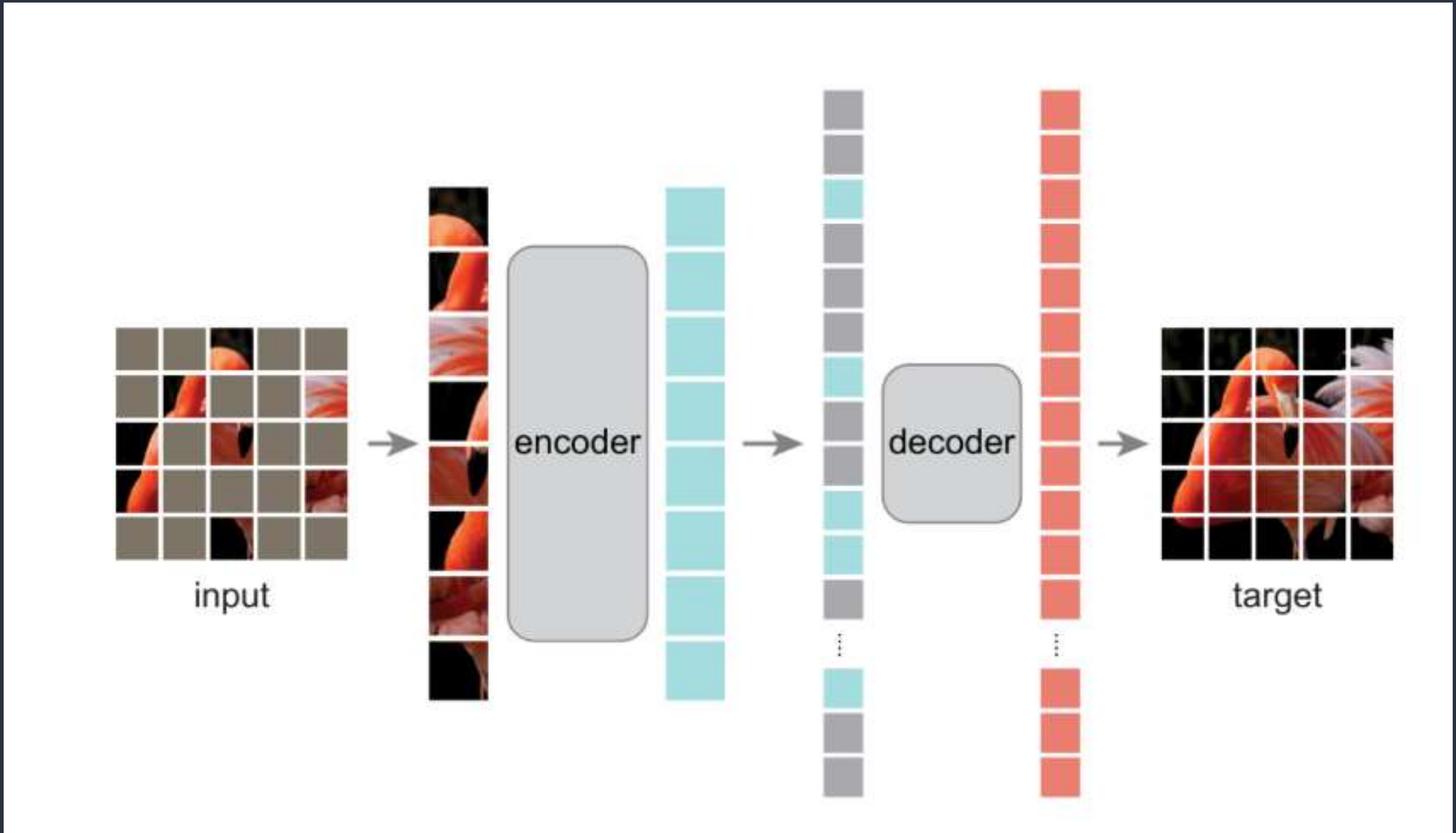
(a) Impact of batch size

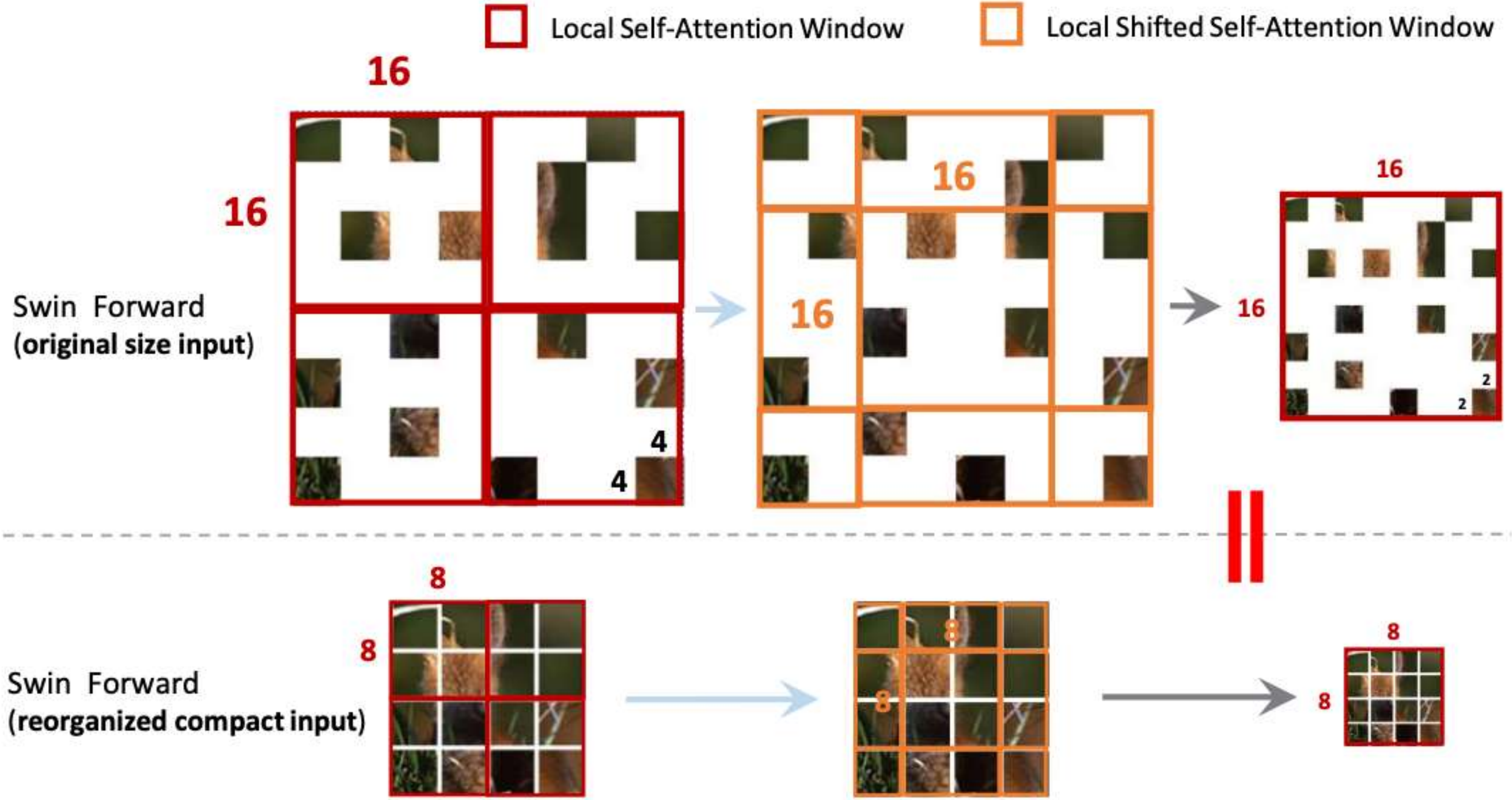


(b) Impact of progressively removing transformations

04 MAE

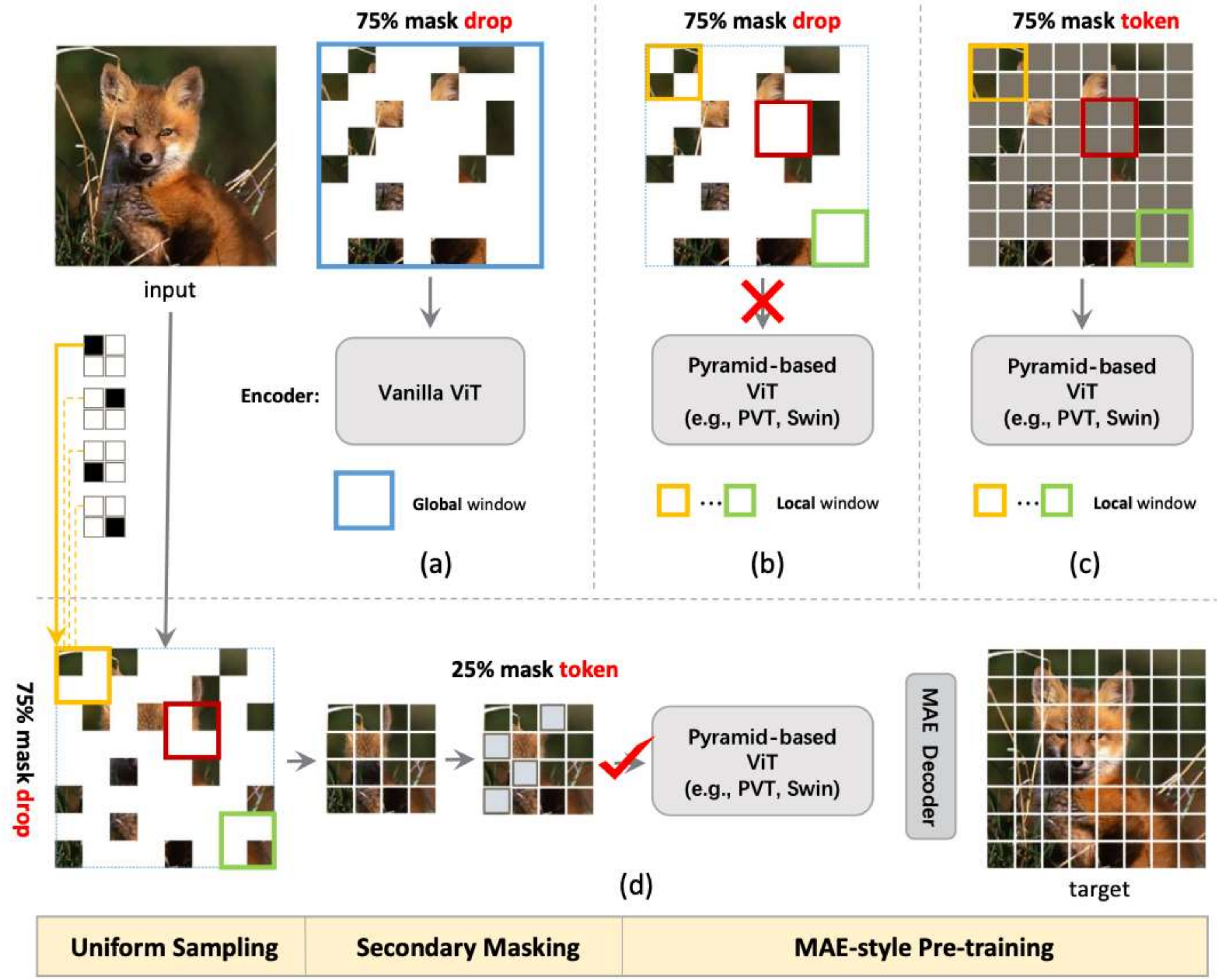
MAE - Masked AutoEncoder



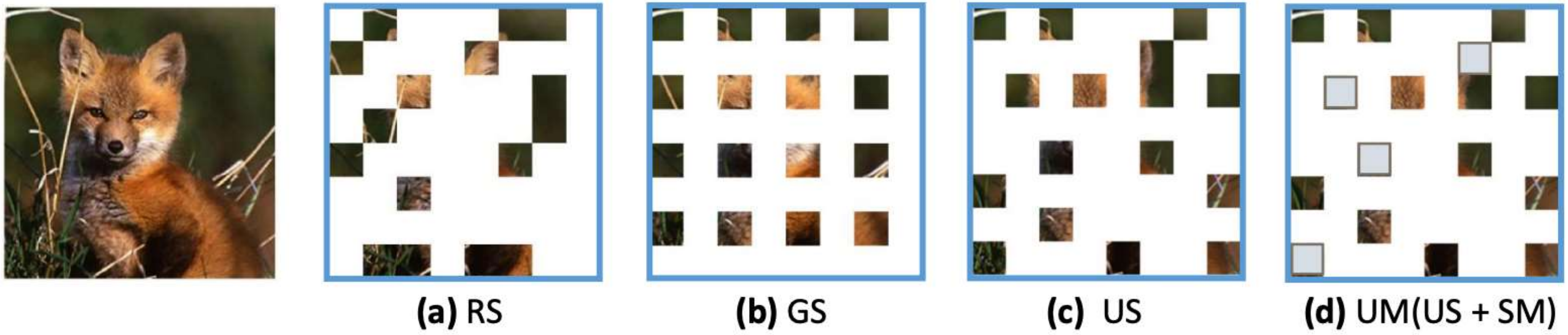


Local Self-Attention Window
Local Shifted Self-Attention Window
Swin Forward
(original size input)
Swin Forward
(reorganized compact input)
16
16
4
4
16
16
16
16
2
2
8
8
8
8
8
8

UM-MAE



75% mask drop
75% mask drop
75% mask token
input
Encoder:
Vanilla ViT
Pyramid-based ViT (e.g., PVT, Swin)
Pyramid-based ViT (e.g., PVT, Swin)
Global window
Local window
Local window
(a)
(b)
(c)
75% mask drop
25% mask token
Pyramid-based ViT (e.g., PVT, Swin)
MAE Decoder
target
(d)
Uniform Sampling
Secondary Masking
MAE-style Pre-training



75% masked

Sampling Strategy (25%)	Pyramid Support	SM Ratio	Pre-train Loss	ImageNet-1K Top-1 Acc	ADE20K mIoU	ADE20K aAcc	COCO AP	COCO AP_{50}	COCO 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

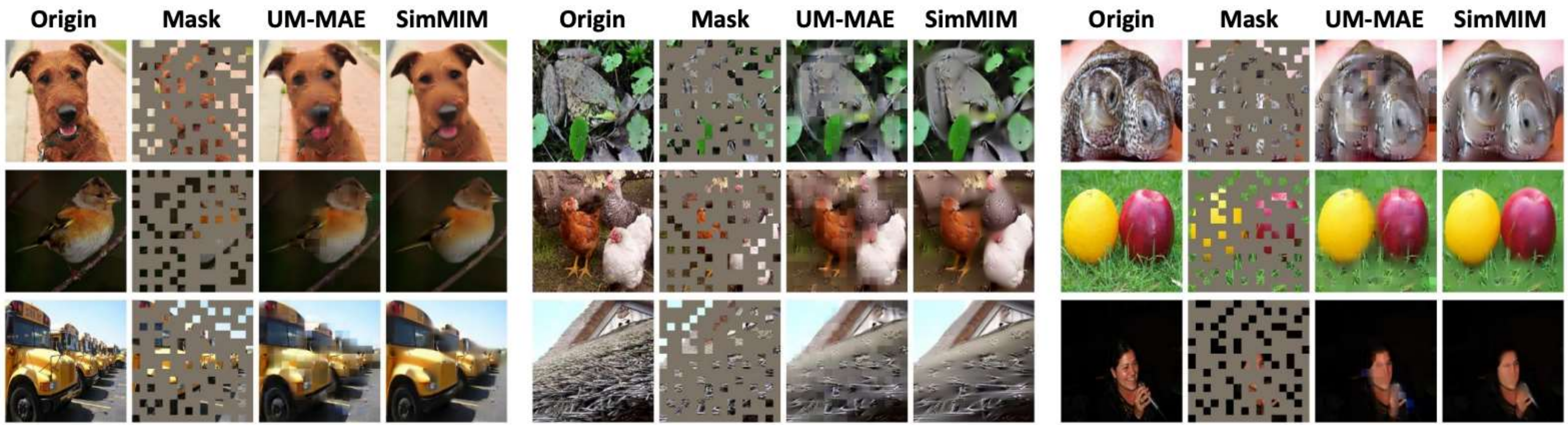


Figure 8: **Uncurated reconstruction visualizations under the same 75% mask pattern.** The models are both pre-trained for 800 epochs.

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)**

05 BEIT

block-wise mask 40%