Paradigms of Self-Supervised Representation Learning in Vision

Xinlei Chen

CMU 11-667 Guest Lecture, 10/2023

facebook
Artificial Intelligence Research

Self-Supervised Learning

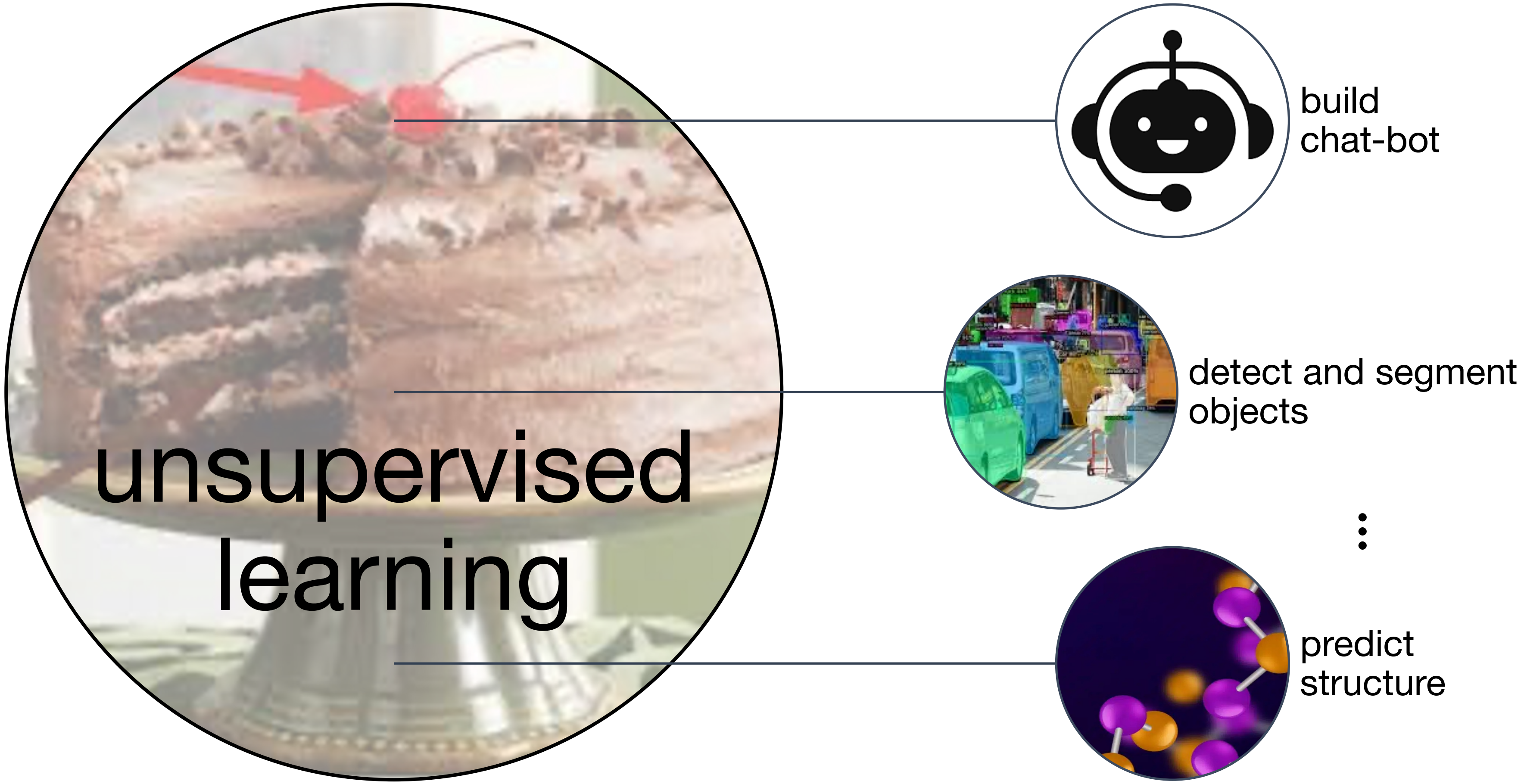
- Pre-train representations without labels for downstream tasks

Self-Supervised Learning

- Pre-train representations without labels for downstream tasks

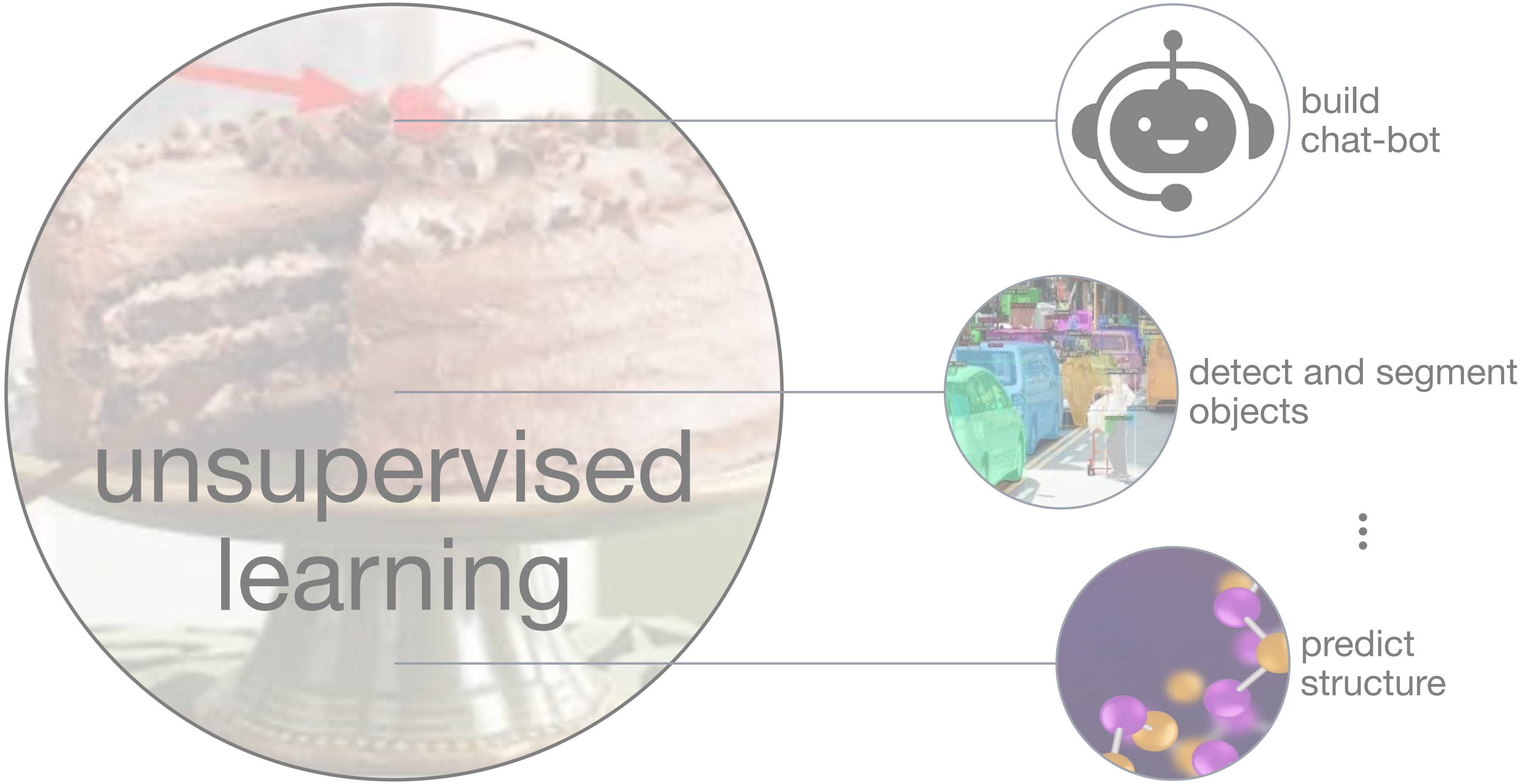
Self-Supervised Learning

- Pre-train representations without labels for downstream tasks



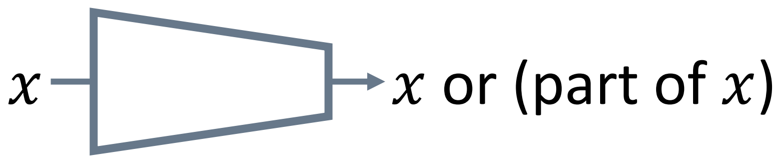
Self-Supervised Representation Learning

- Pre-train representations without labels for downstream tasks

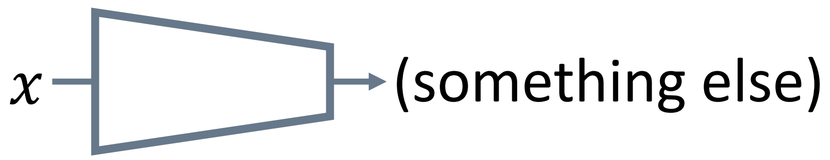


Paradigms for Self-Supervised Learning

- Reconstructive / Autoencoding

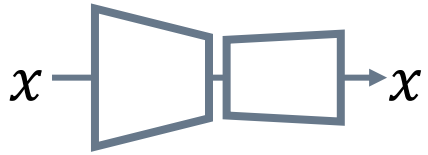


- Non-Reconstructive



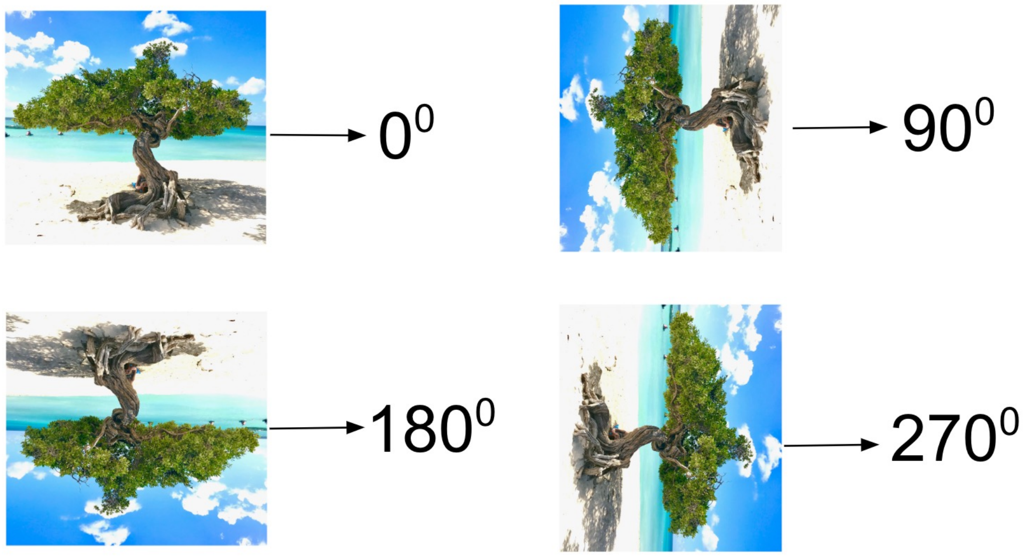
Reconstructive Self-Supervised Learning

- Simplest form --- autoencoding
  - full data x as input, full x as output
  - often in the form of an (encoder, decoder) pair
  - pre-deep learning examples:
    - principal component analysis (PCA)
    - k-means clustering
      - optimize the (1) cluster centers and (2) cluster assignments
      - such that the reconstruction loss is minimized
      - when all data points are replaced with their cluster centers
    - other variants of matrix factorization / decomposition



Reconstructive Self-Supervised Learning

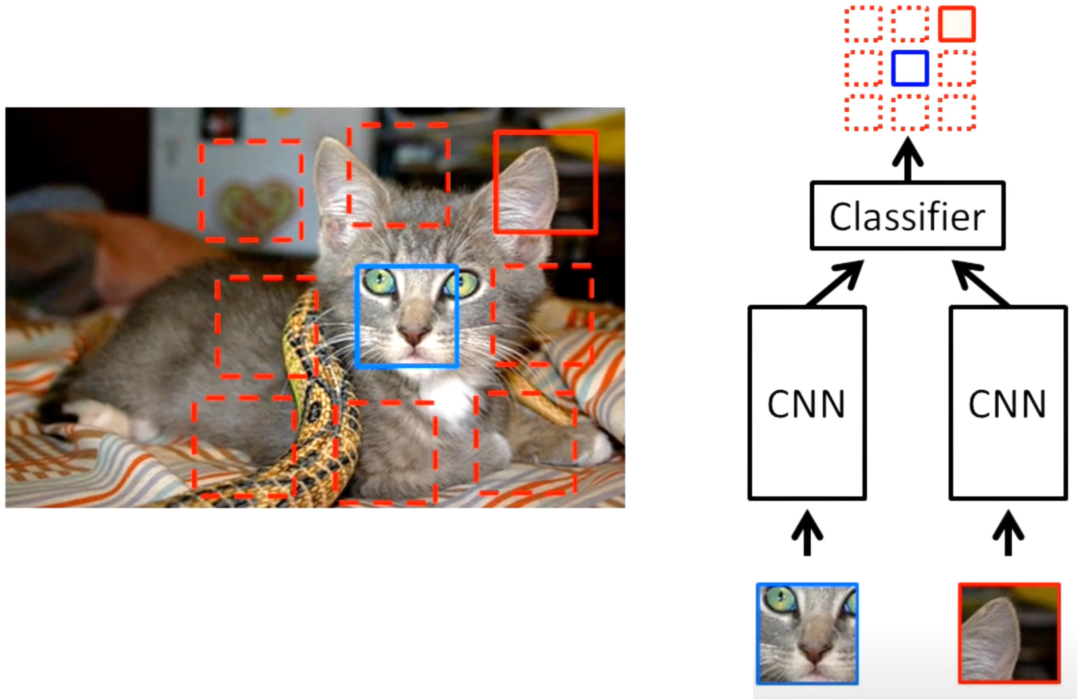
- Augmented form --- with transformation
  - each data x has a transformation t sampled from pre-defined set $\mathcal{T}$
  - easy to design $\mathcal{T}$, so popular in vision
  - now the new data is $\dot{x} = (x, t)$
  - can predict either x or t as part of $\dot{x}$
  - examples:
    - (t) rotation prediction



[Gidaris et al, ICLR 2018]

Reconstructive Self-Supervised Learning

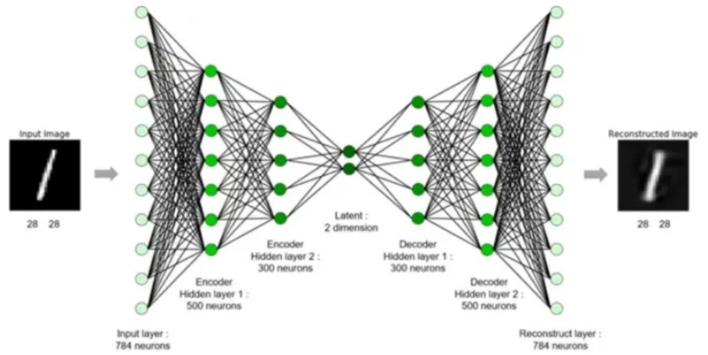
- Augmented form --- with transformation
  - each data x has a transformation t sampled from pre-defined set $\mathcal{T}$
  - easy to design $\mathcal{T}$, so popular in vision
  - now the new data is $\dot{x} = (x, t)$
  - can predict either x or t as part of $\dot{x}$
  - examples:
    - (t) rotation prediction
    - (t) relative position prediction

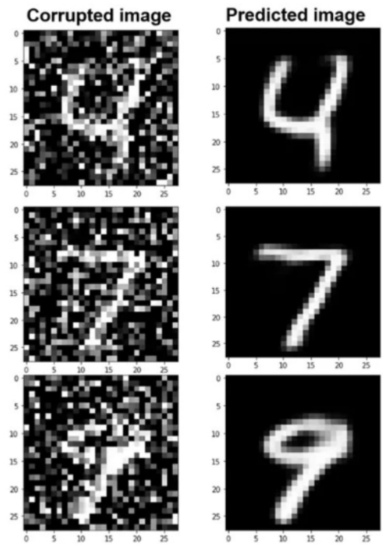


[Doersch et al, ICCV 2015]

Reconstructive Self-Supervised Learning

- <u>Augmented</u> form --- with transformation
  - each data x has a transformation t sampled from pre-defined set $\mathcal{T}$
  - easy to design $\mathcal{T}$, so popular in vision
  - now the new data is $\dot{x} = (x, t)$
  - can predict either x or t as part of $\dot{x}$
  - examples:
    - (t) rotation prediction
    - (t) relative position prediction
    - (x) denoising autoencoder





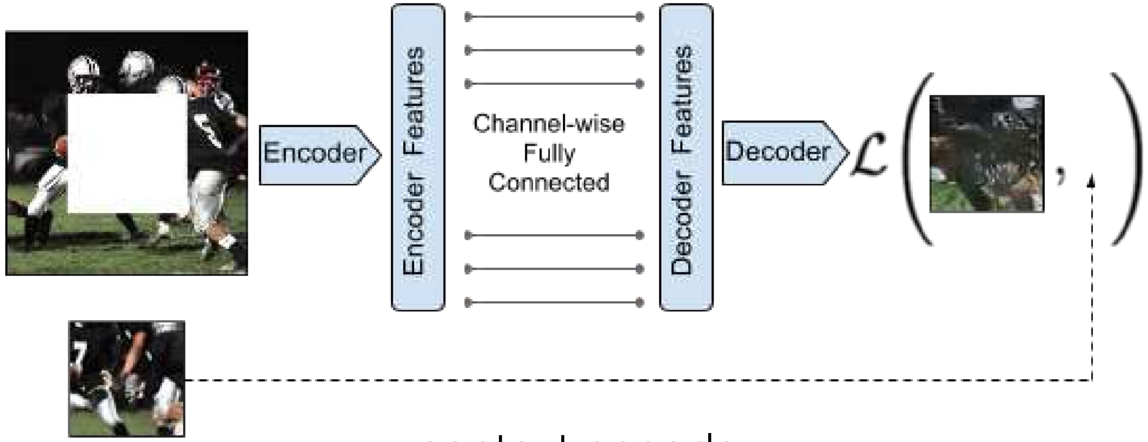
[Vincent et al, ICML 2008]

Reconstructive Self-Supervised Learning

- <u>Augmented</u> form --- with transformation
  - each data x has a transformation t sampled from pre-defined set $\mathcal{T}$
  - easy to design $\mathcal{T}$, so popular in vision
  - now the new data is $\dot{x} = (x, t)$
  - can predict either x or t as part of $\dot{x}$
  - examples:
    - (t) rotation prediction
    - (t) relative position prediction
    - (x) denoising autoencoder
    - (x) masked autoencoder

Reconstructive Self-Supervised Learning

- <u>Augmented</u> (special) form --- with *masking / dropping*
  - *channels*: colorization
  - *center patch*: context encoder
  - *random patch*: MAE (to talk about)



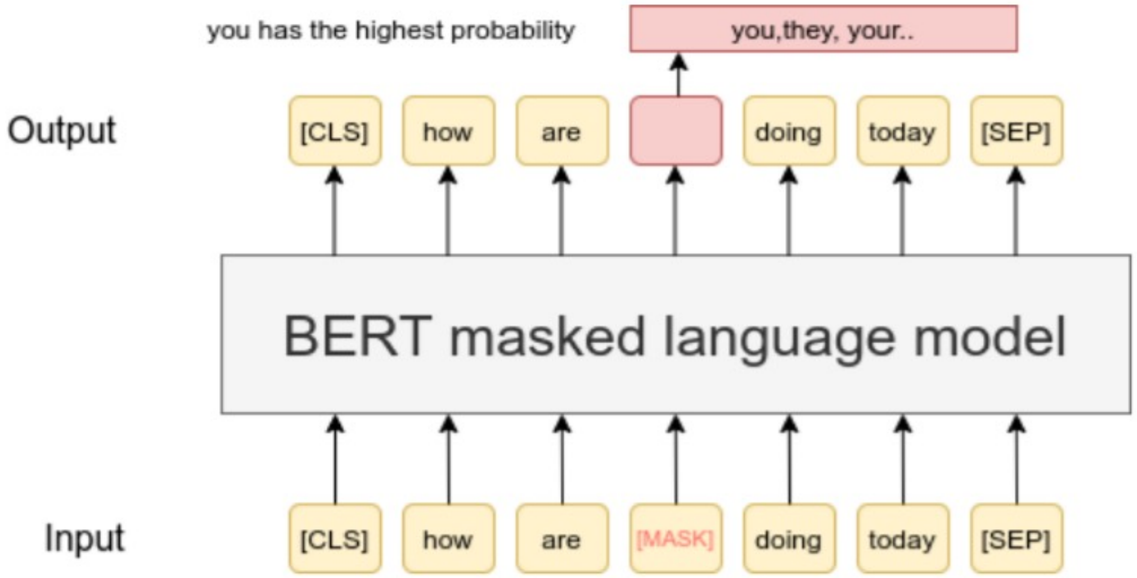
context encoder

[Zhang et al, ECCV 2016] [Zhang et al, CVPR 2017] [Pathak et al, CVPR 2016] [He et al, CVPR 2022]

Reconstructive Self-Supervised Learning

- Augmented (special) form --- with *masking / dropping*
  - *channels*: colorization
  - *center* patch: context encoder
  - *random* patch: MAE (to talk about)

- Even more effective for *text*
  - *random* masking: BERT

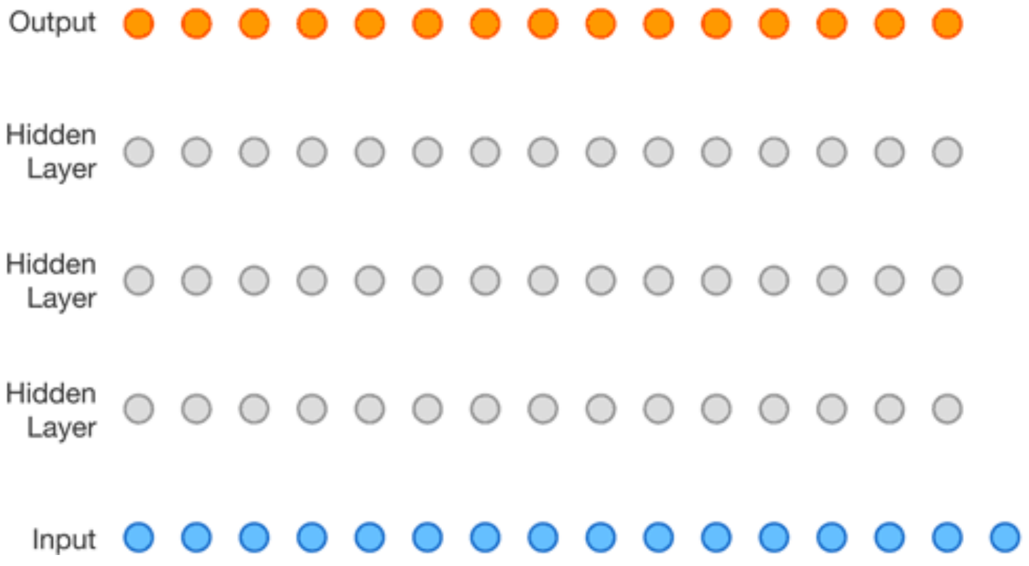


[Devlin et al, NAACL 2019]

Reconstructive Self-Supervised Learning

- Augmented (special) form --- with *masking / dropping*
  - *channels*: colorization
  - *center* patch: context encoder
  - *random* patch: MAE (to talk about)

- Even more effective for *text*
  - *random* masking: BERT
  - *sequential* masking: GPT



[Radford et al, 2018] [Radford et al, 2019] [Brown et al, 2020] [OpenAI, 2023]

ArXiv: https://arxiv.org/abs/2111.06377, CVPR 2022
Code: https://github.com/facebookresearch/mae

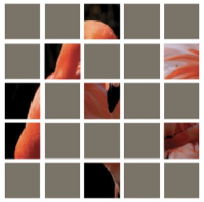
Masked Auto-Encoders Are Scalable Vision Learners

Kaiming He*†, Xinlei Chen*, Saining Xie, Yanghao Li, Piotr Dollar, Ross Girshick

facebook
Artificial Intelligence Research

What is MAE?

- Very simple self-supervised learning method
- BERT-like algorithm and behavior
- But with crucial changes for images

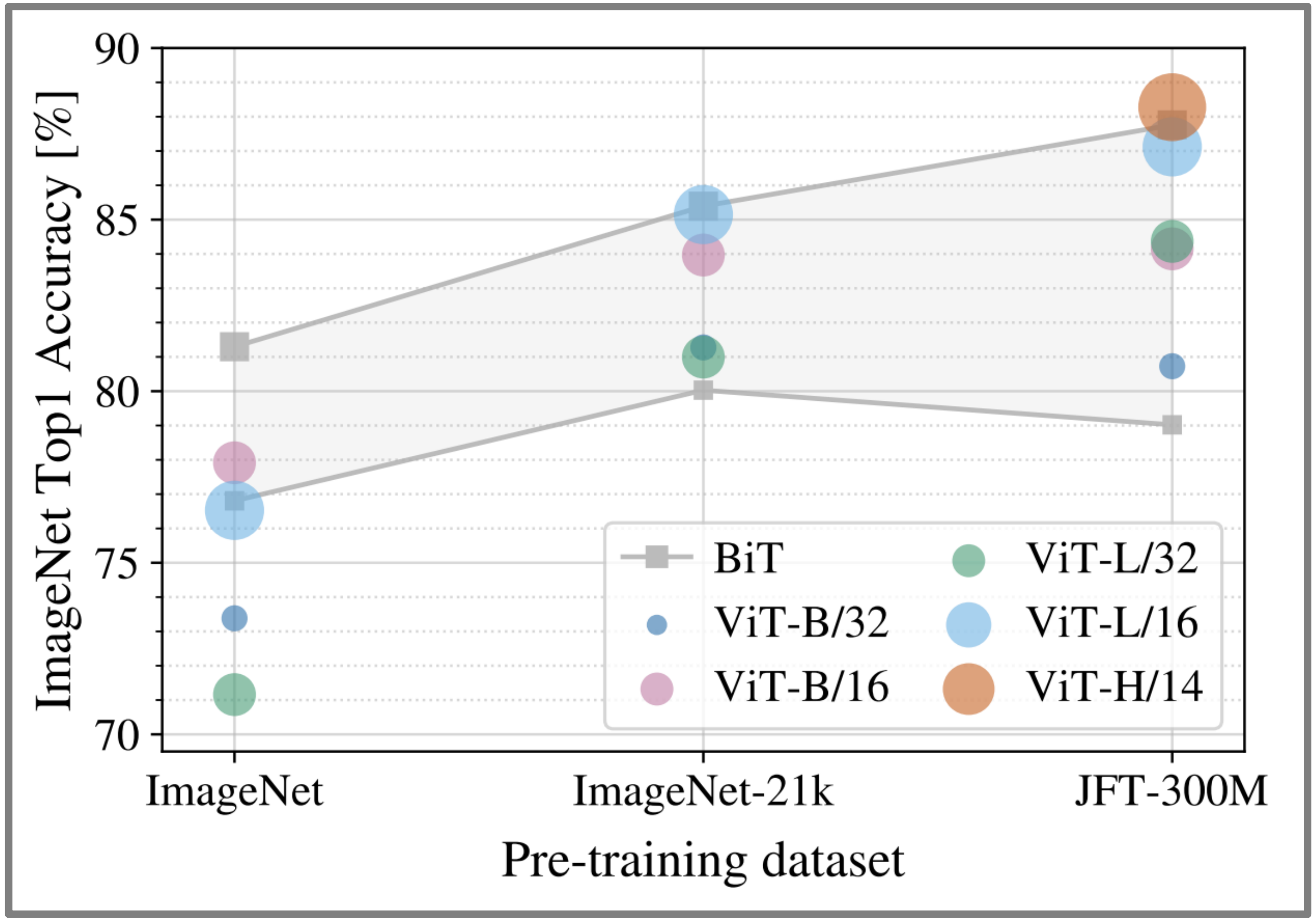


input

Directly predict pixels!

BERT-like: Transformers

- Vision Transformer (ViT)
  - less inductive bias
  - non-overlapping tokenization
    - easier for MAE
- Scalable
  - with larger models
  - on larger datasets



Changes from BERT: *Mask Ratio*

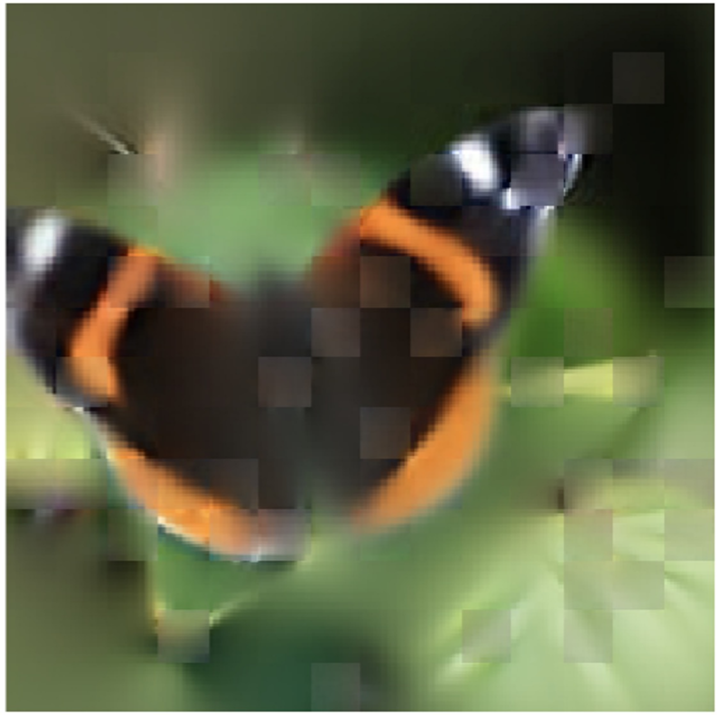
?

Masked input: 80%

You guess?

Changes from BERT: *Mask Ratio*

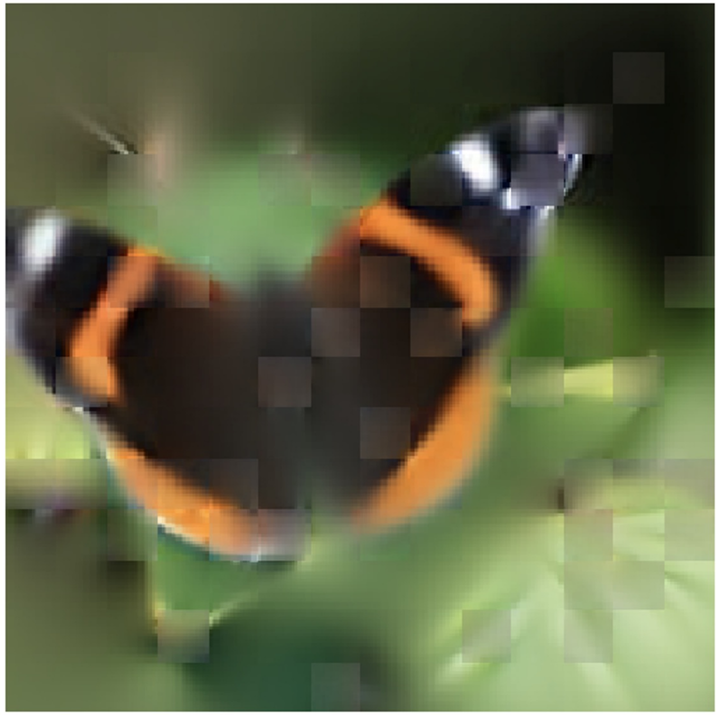
Masked input: 80%



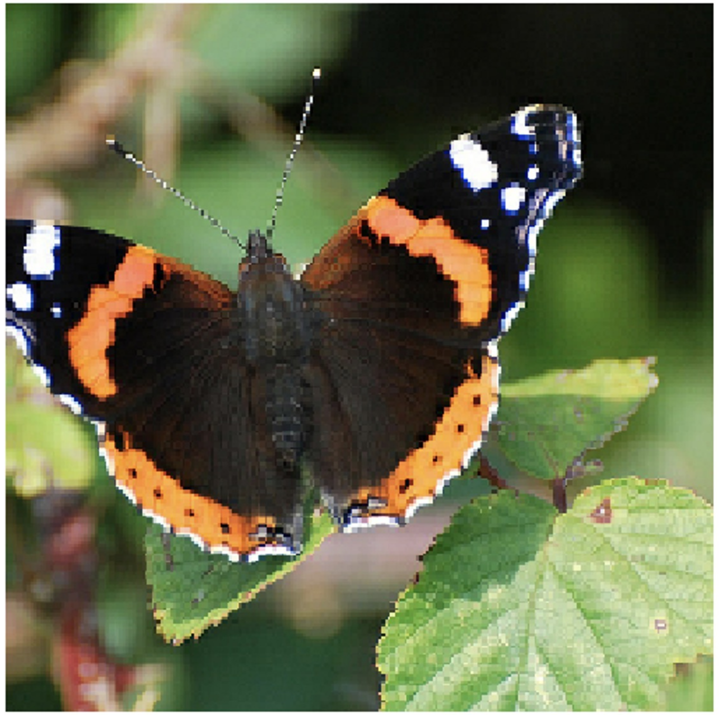
MAE's guess

Changes from BERT: *Mask Ratio*

Masked input: 80%



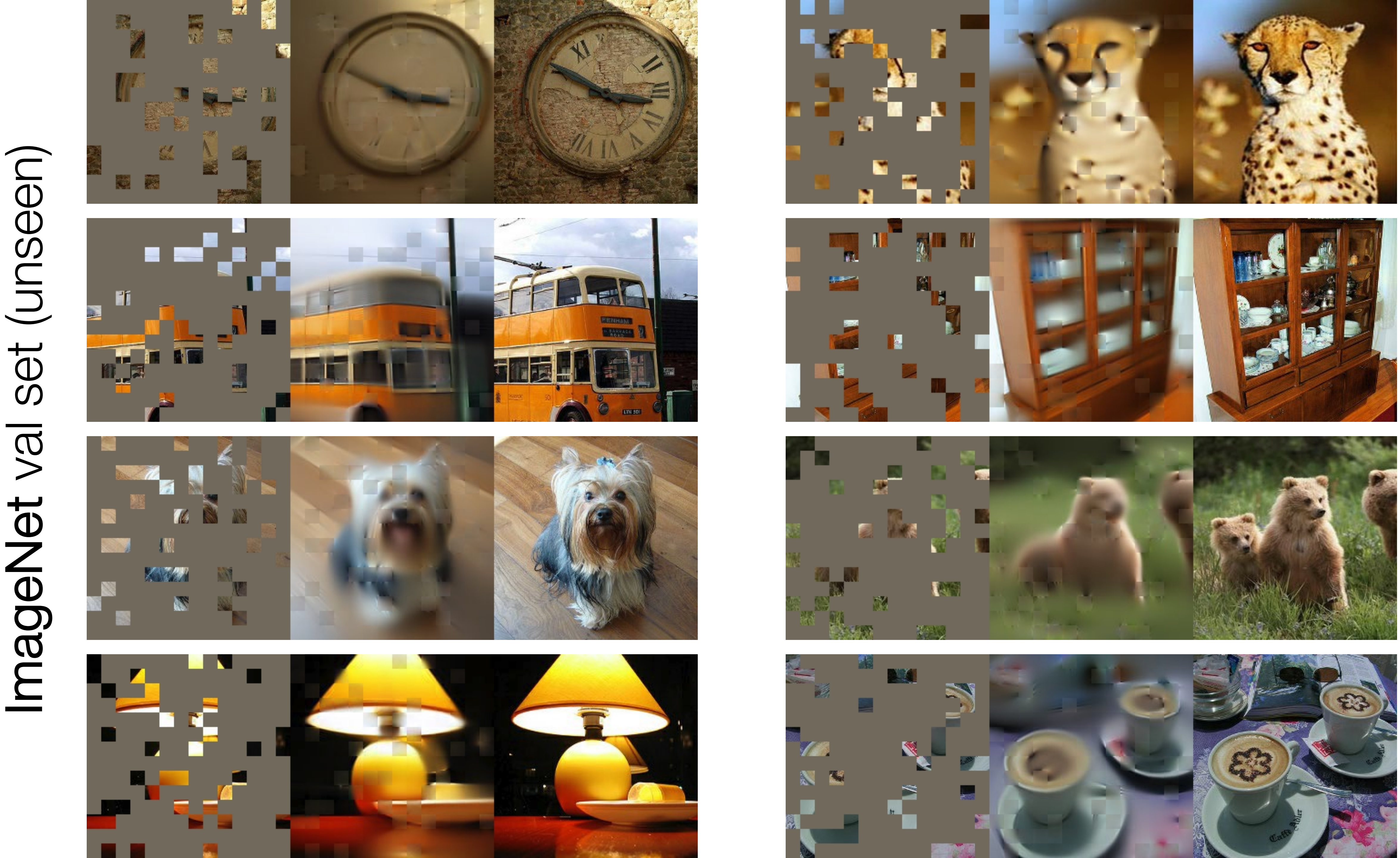
MAE's guess



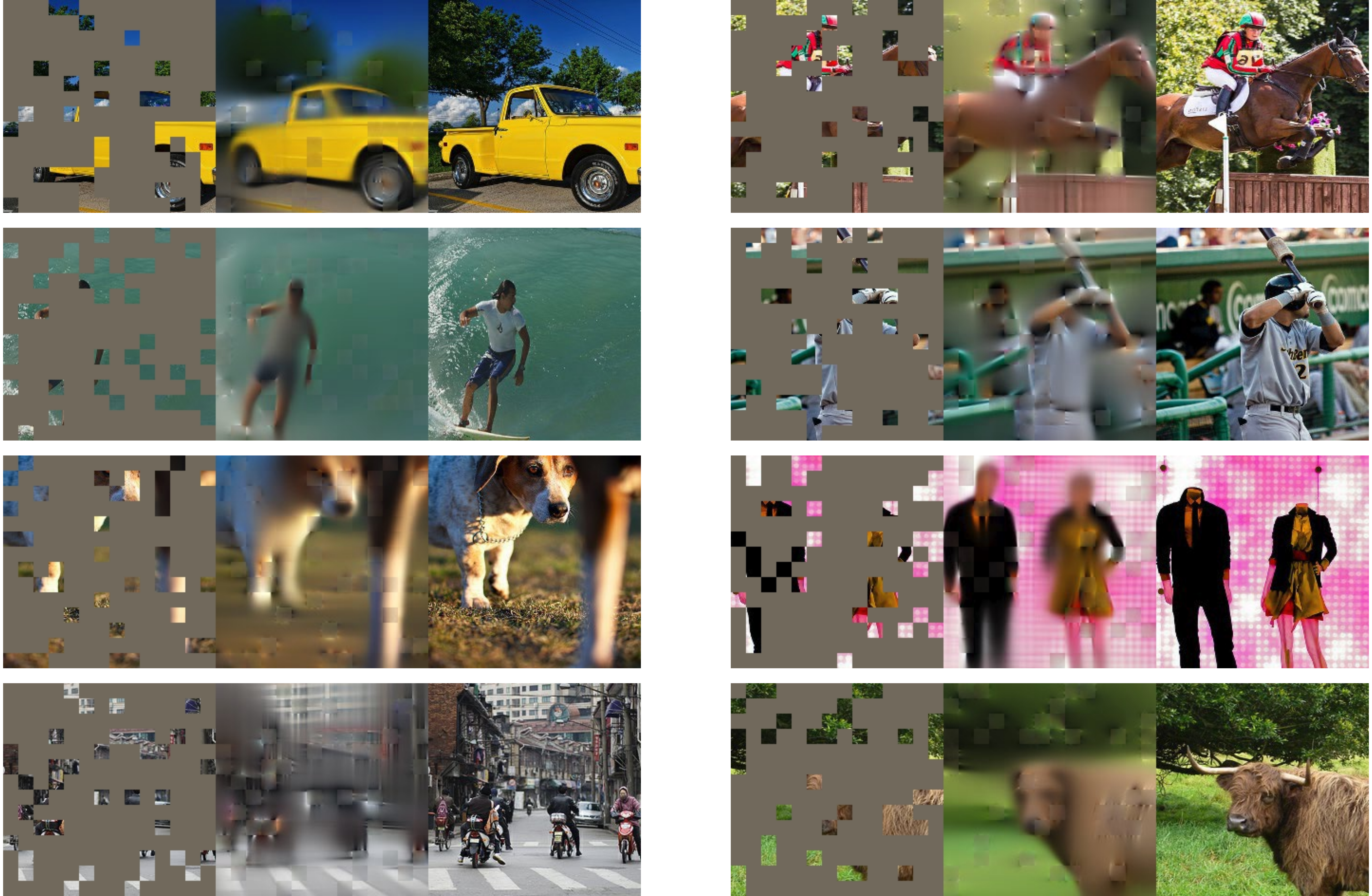
Ground truth

Changes from BERT: *Mask Ratio*

- BERT: 15% is enough to create a challenging task
- MAE: 75% - 80% is about optimal

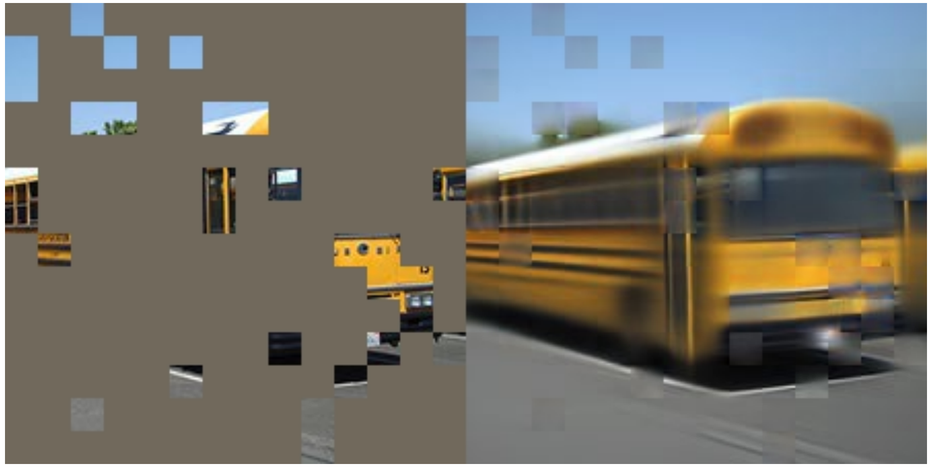

ImageNet val set (unseen)

COCO val set (unseen)



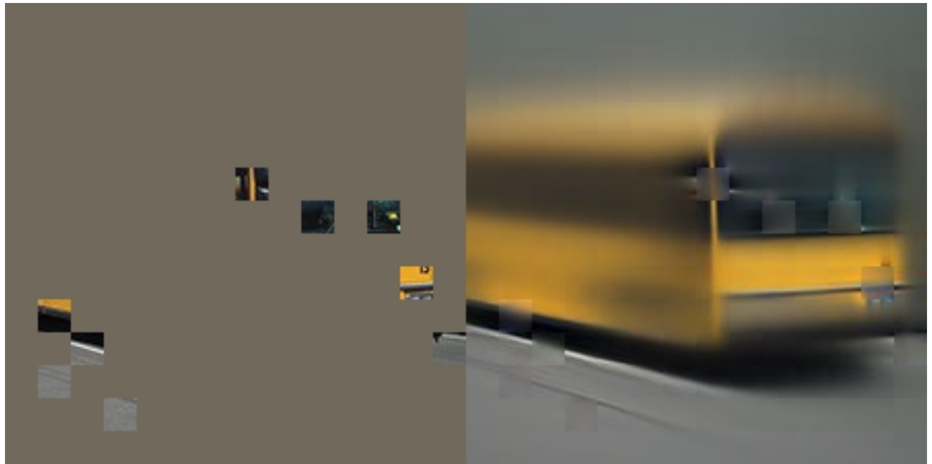
75% mask

original



85% mask

95% mask



MAE Can Generalize

75% mask

original

85% mask

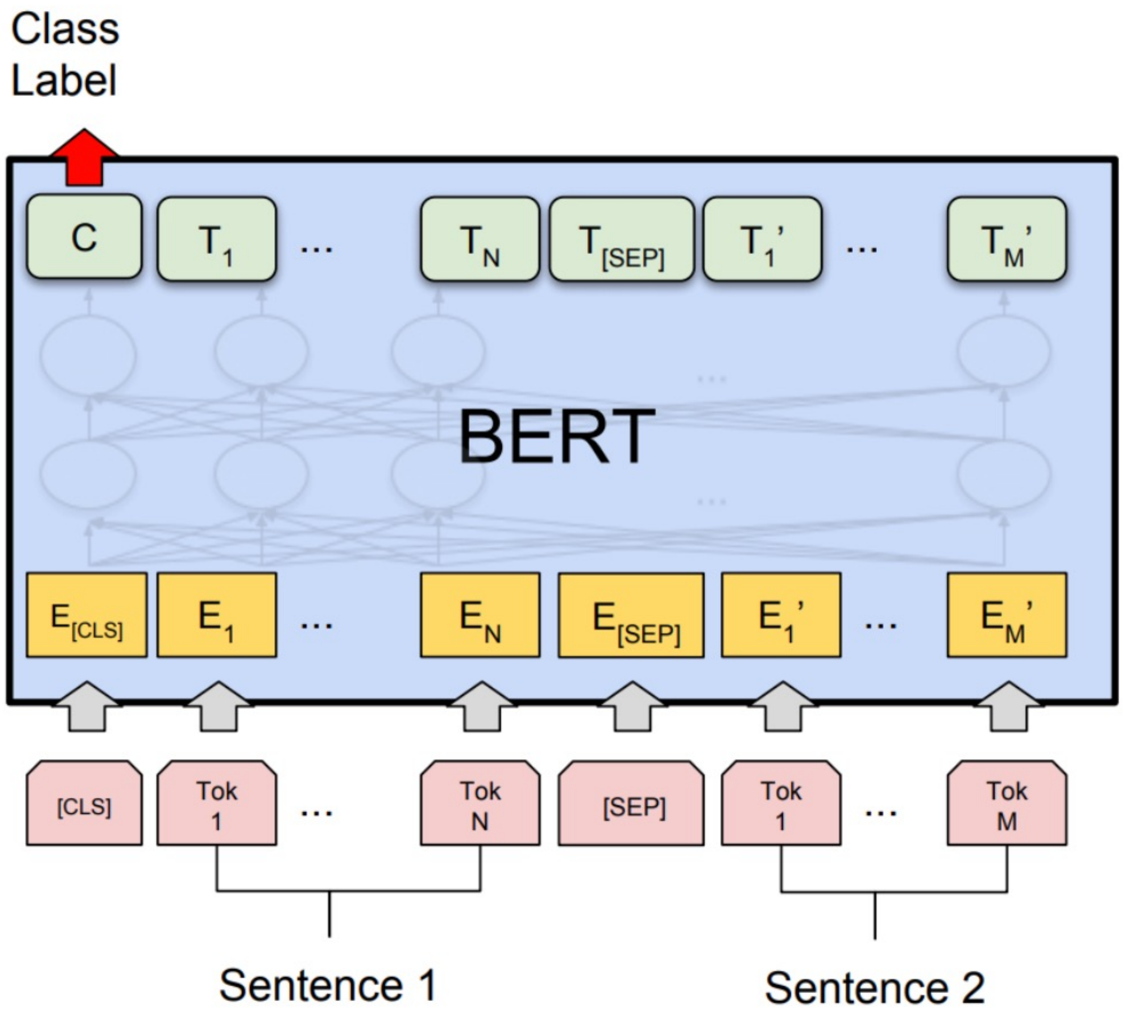
95% mask

MAE Can Generalize

Changes from BERT: *Encoder-Decoder*

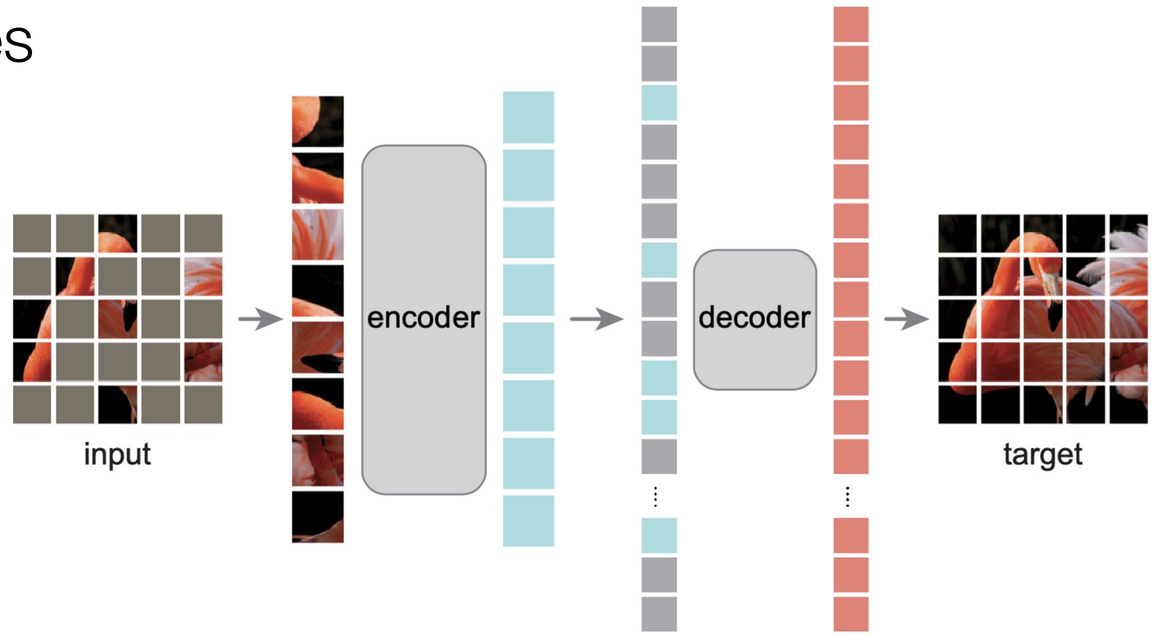
- BERT: encoder only

- MAE:
  - **large** encoder (e.g., ViT-Large)
  - **small** decoder (e.g., ViT-Base)



Changes from BERT: *Encoder-Decoder*

- MAE:
  - *large* encoder on *visible* patches
  - small decoder on *all* patches

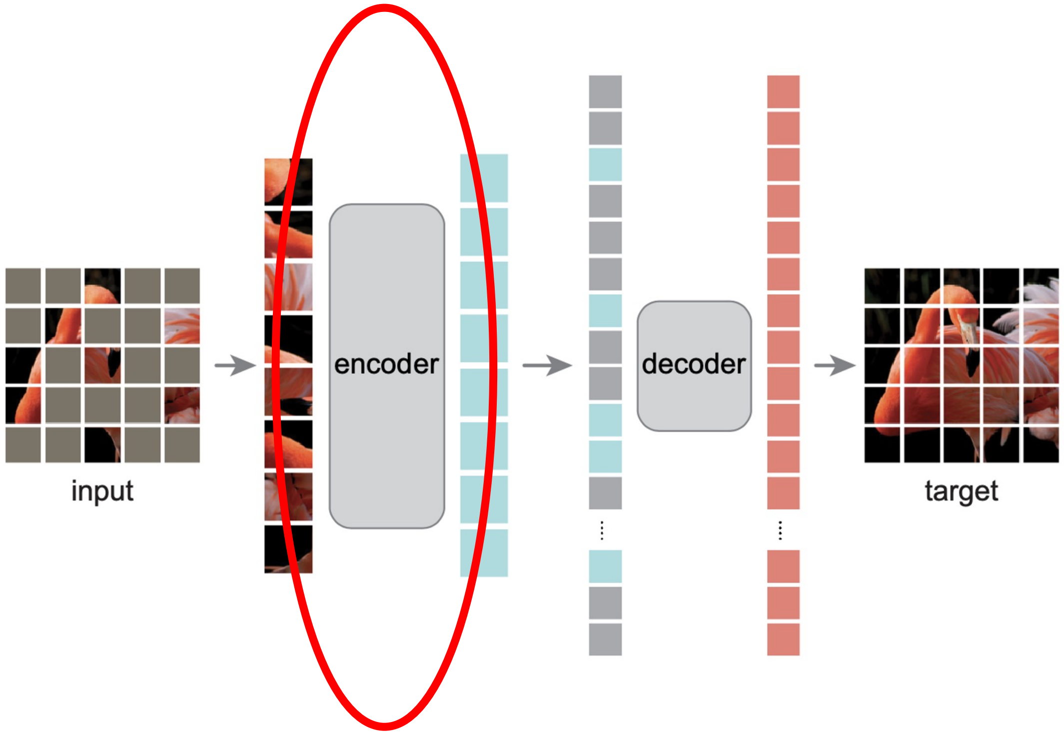


- Very efficient when coupled with <u>high</u> mask ratio (75%)

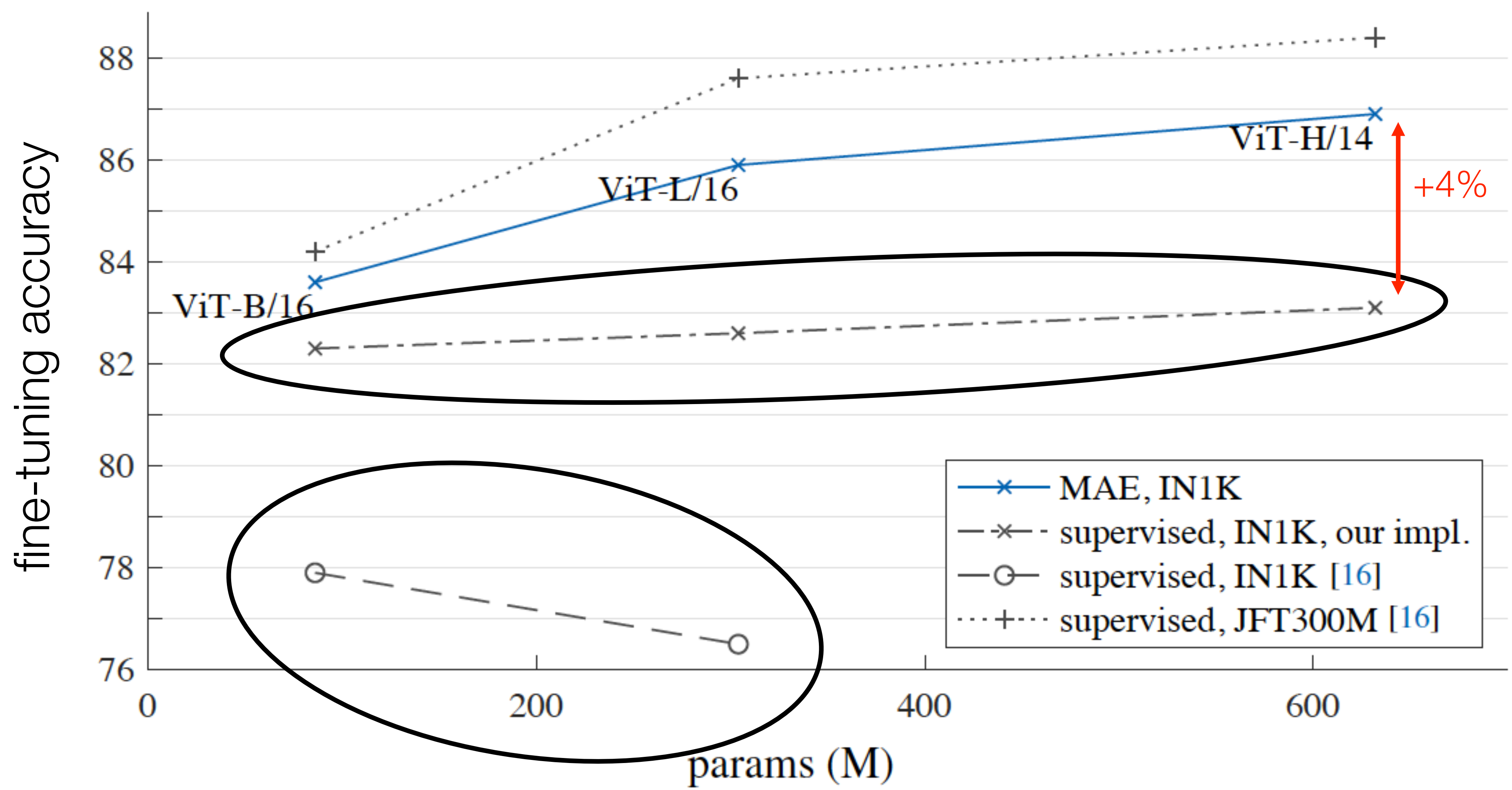
- Single projection layer to map from encoder to decoder

Representation Evaluation: *Encoder Only*

- After MAE pre-training, throw away decoder
- Encoder with *full sequence* is used for benchmark representations



Scalability on ImageNet Classification



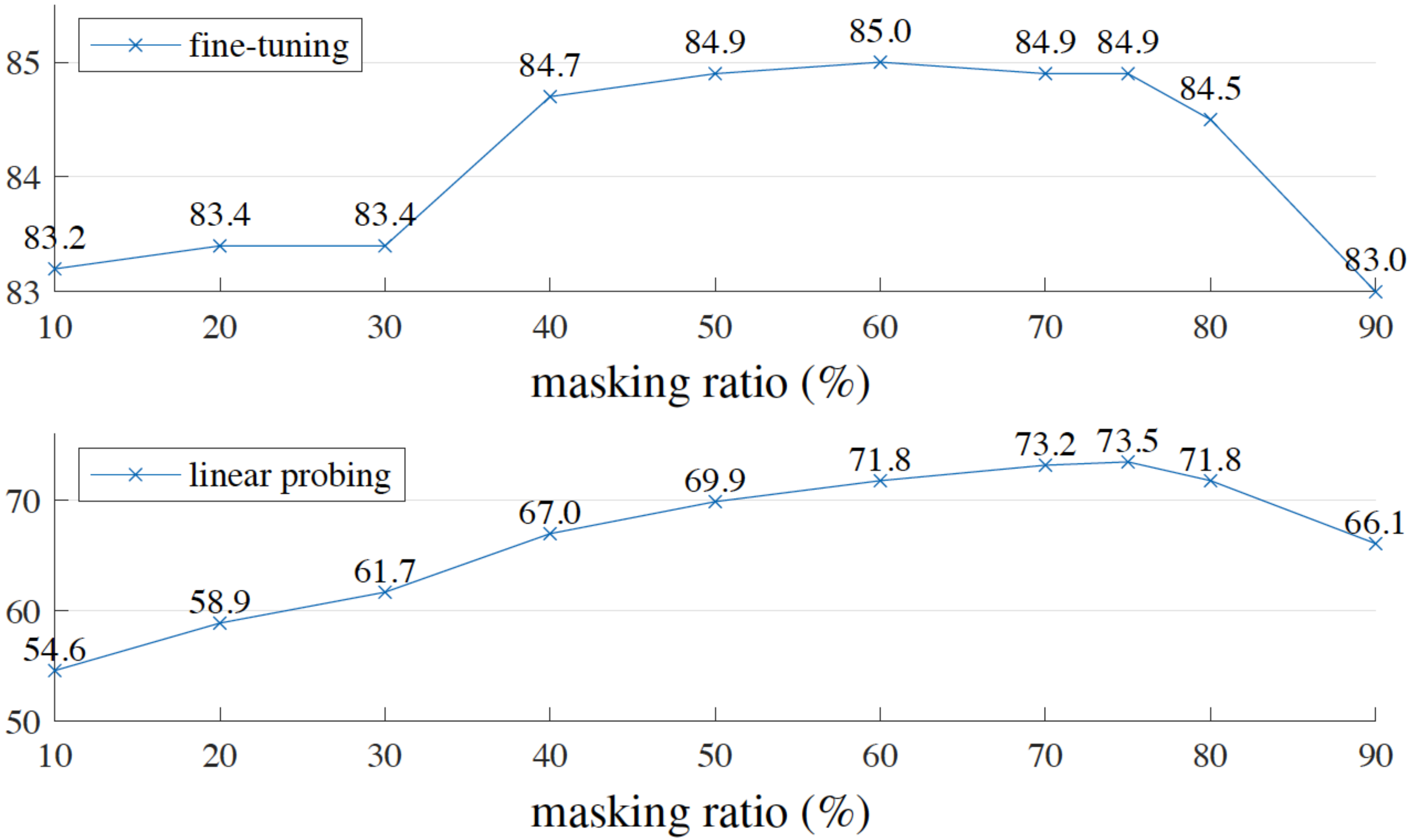
[Li et al, ArXiv 2021]

Object Detection Transfer

initialization	pre-training data	AP^{box} ViT-B	AP^{box} ViT-L	AP^{mask} ViT-B	AP^{mask} ViT-L
supervised	IN1k w/ labels	47.9	49.3	42.9	43.9
random	*none*	48.9	50.7	43.6	44.9
MoCo v3	IN1k	47.9	49.3	42.7	44.0
BEiT	IN1k+DALL·E	49.8	**53.3**	44.4	47.1
MAE	IN1k	**50.3**	**53.3**	**44.9**	**47.2**

- On COCO, improved previous pre-training by 3 to 4%

Analysis: Mask Ratio



Analysis: Mask Token in Encoder

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

- Encoder with [M] is default in BERT
  - big domain gap for linear probing
  - pre-train sees 25% of the images only, while evaluation sees 100%
- Encoder w/o [M] is default in MAE

Analysis: Augmentations

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

- MAE can work with minimal data augmentation
- In contrast, augmentation recipes can be crucial for others
- Well, one can view “masking” as a type of augmentation

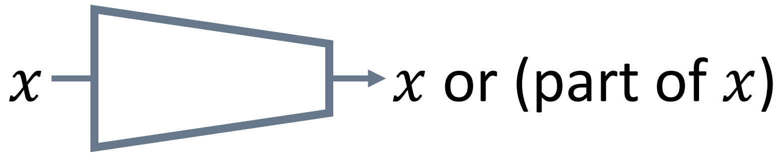
Analysis: Reconstruction Target

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.

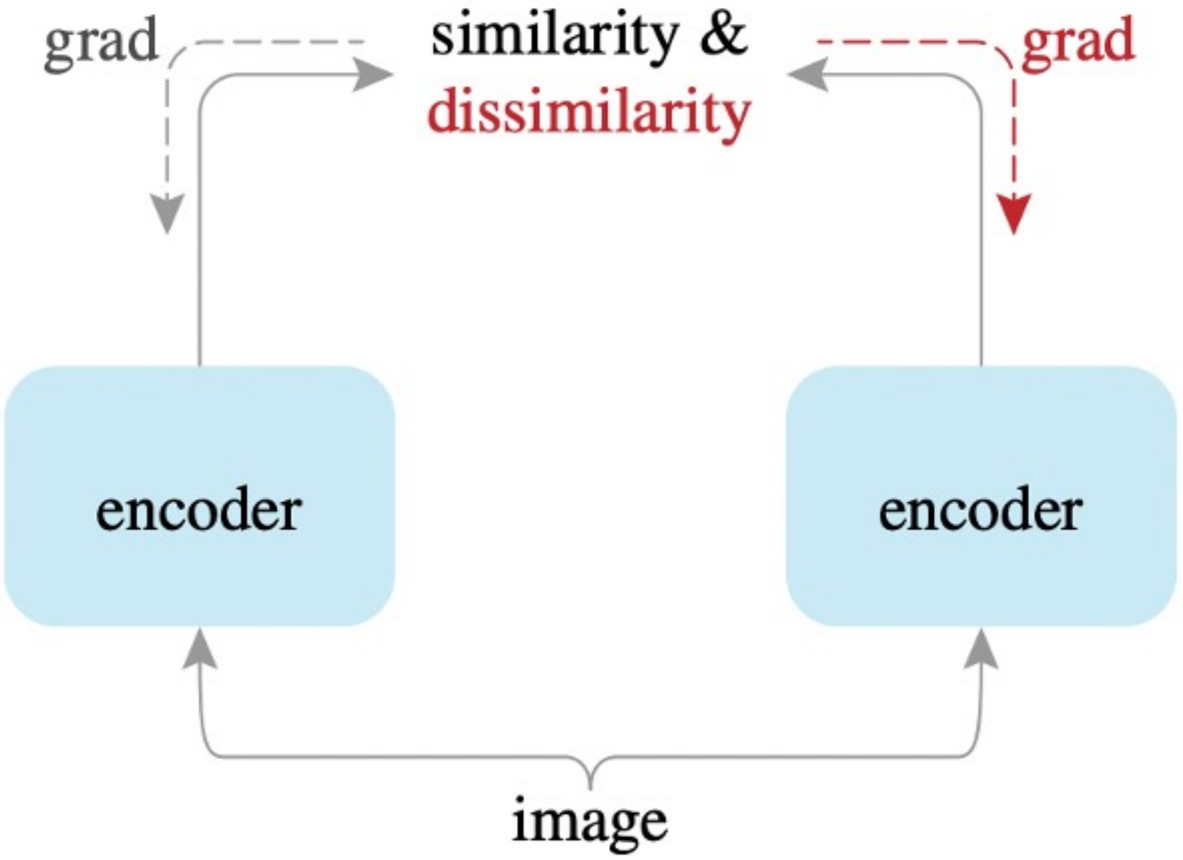
- Pixels with normalization: per-patch
- PCA: only keeps low-frequency component
- dVAE token: from DALLE

Reconstructive Self-Supervised Learning



- *Simplest* form
  - autoencoding
- *Augmented* form
  - with transformation
- *Augmented* (special) form
  - with masking / dropping

How about Contrastive Learning?

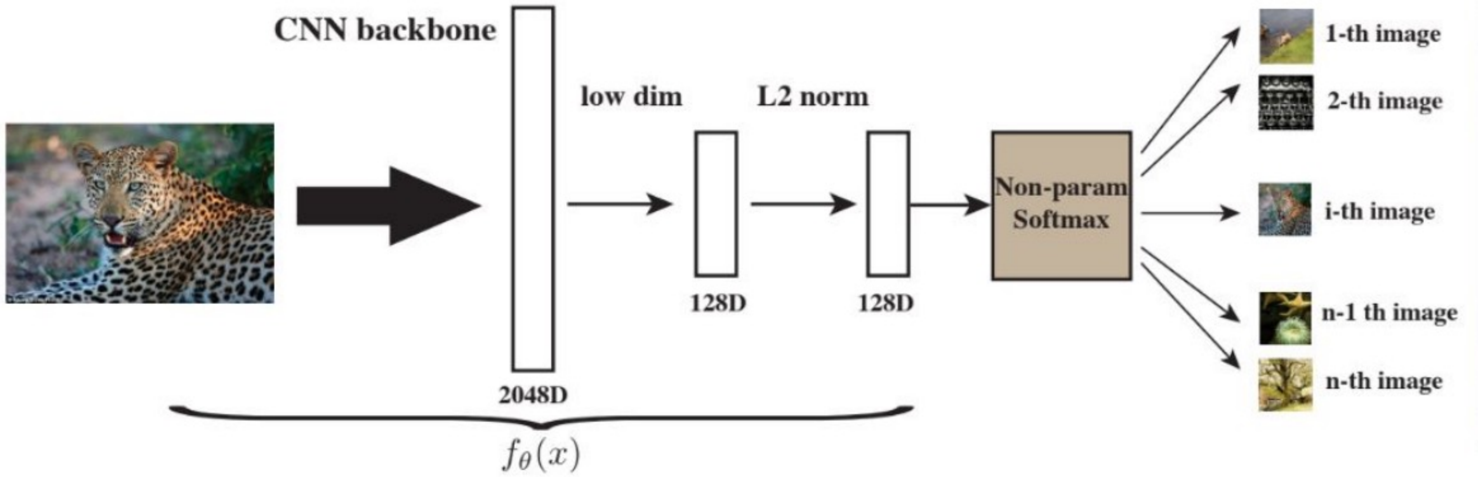


Claim (!): it is also an *implicit* form of reconstructive learning

[Chen et al, ICML 2020]

Connection Point: Instance Discrimination

- Implicit form --- with *instance discrimination* on a dataset
  - each data has its own class, so one instance per class
  - for a data set with N data points, we have N classes
  - now the new data is $\ddot{x} = (x, i)$, where i is an instance indicator
  - the task is to predict i as part of $\ddot{x}$



[Dosovitskiy et al, TPAMI 2015] [Wu et al, CVPR 2018]

Contrastive is Reconstructive

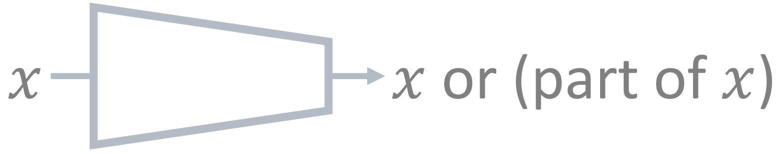
- Implicit form --- with *contrastive learning*
  - instance discrimination: predict i from a fixed set as part of $\ddot{x}$
  - contrastive (Siamese net): predict i from a dynamic set as part of $\ddot{x}$
  - can be easily augmented with transformations $\mathcal{T}$
  - now we have $\ddot{x} = (x, t, i)$, the task is to predict i as part of $\ddot{x}$

- So contrastive learning is reconstructive learning
  - And a rather *weak* one --- that relies heavily on $\mathcal{T}$ to make it meaningful

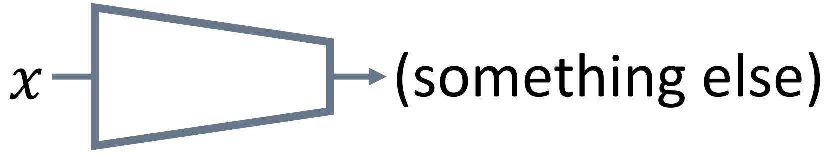
[Chen et al, ICML 2020] [He et al, CVPR 2020] [Chen et al, arXiv 2020] [Chen et al, ICCV 2021]

Paradigms for Self-Supervised Learning

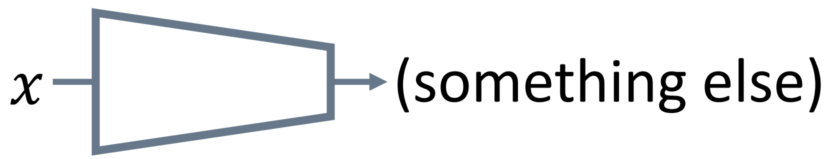
- Reconstructive / Autoencoding



- Non-Reconstructive



Does Non-Reconstructive Even Work?



- If it predicts something else, won't it simply *ignore* the data?

- Yes, it is! So circumventing this issue is a crucial topic in non-reconstructive SSL

- Will take our work, SimSiam as an example
  - but the underlying mechanism is still unclear

ArXiv: https://arxiv.org/abs/2011.10566, CVPR 2021
Code: https://github.com/facebookresearch/simsiam

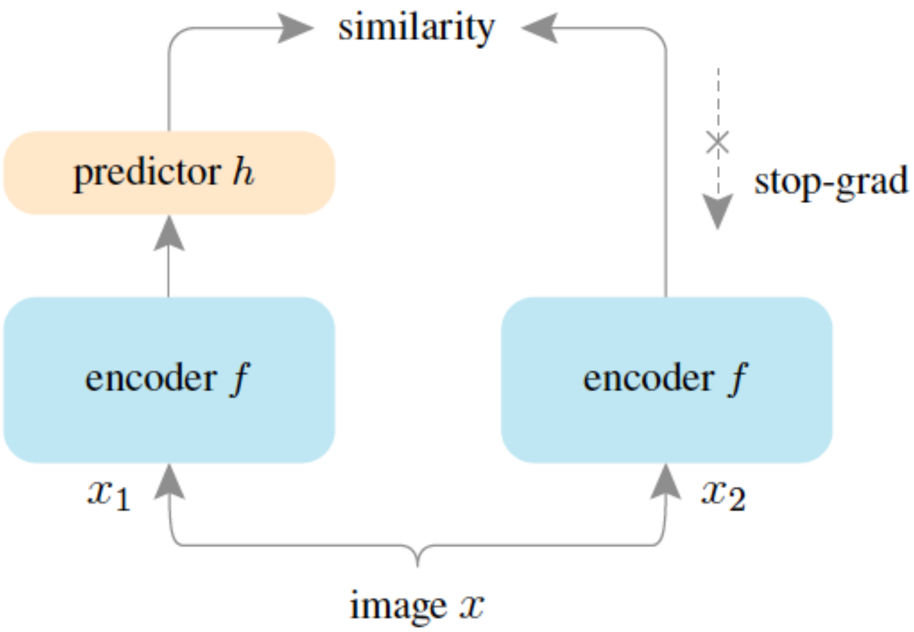
Exploring Simple Siamese Representation Learning

Xinlei Chen

Kaiming He

facebook
Artificial Intelligence Research

SimSiam Architecture

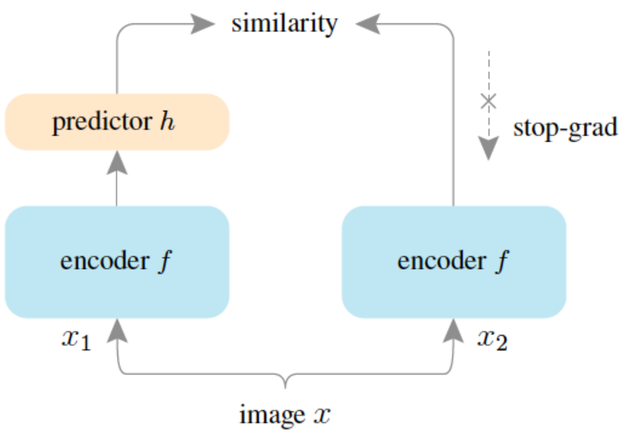


- Contrastive learning: reconstruct i via (similarity + dissimilarity)
- SimSiam: only predict similarity, so <u>no reconstruction of input</u> i

SimSiam Algorithm

Algorithm 1 SimSiam Pseudocode, PyTorch-like

```
# f: backbone + projection mlp
# h: prediction mlp

for x in loader: # load a minibatch x with n samples
   x1, x2 = aug(x), aug(x) # random augmentation
   z1, z2 = f(x1), f(x2) # projections, n-by-d
   p1, p2 = h(z1), h(z2) # predictions, n-by-d

   L = D(p1, z2)/2 + D(p2, z1)/2 # loss

   L.backward() # back-propagate
   update(f, h) # SGD update

def D(p, z): # negative cosine similarity
   z = z.detach() # stop gradient

   p = normalize(p, dim=1) # l2-normalize
   z = normalize(z, dim=1) # l2-normalize
   return -(p*z).sum(dim=1).mean()
```

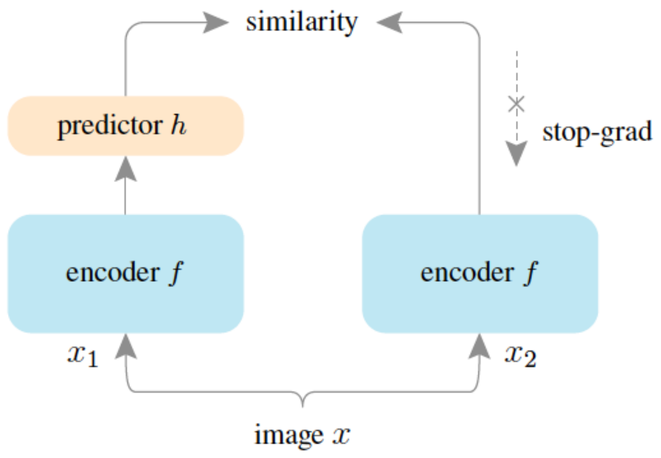


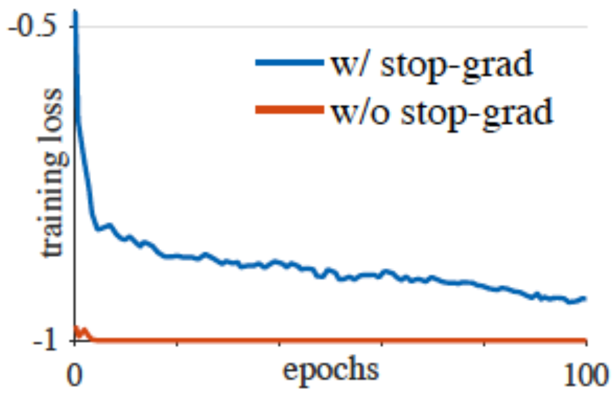
- *Symmetrized* loss
- Simple cosine similarity
- Gradient only via *predictor*
  - stop-grad on other

Stop-Grad is Crucial for SimSiam

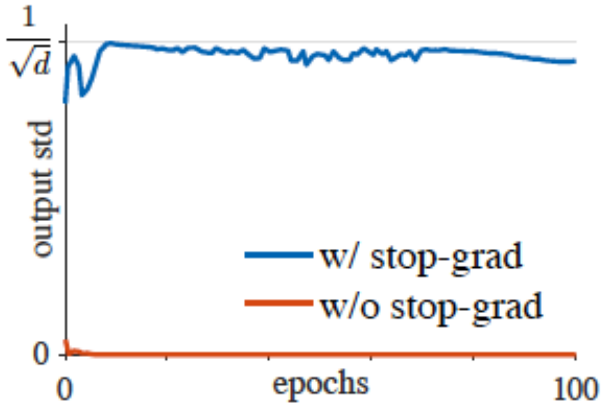
- Without it, representation collapses
  - *Implicit* for momentum encoder

setting	top-1
w/ stop-grad	67.7±0.1
w/o stop-grad	0.1

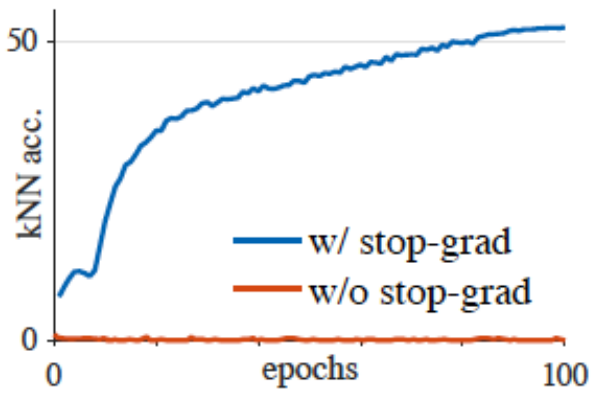




loss curve



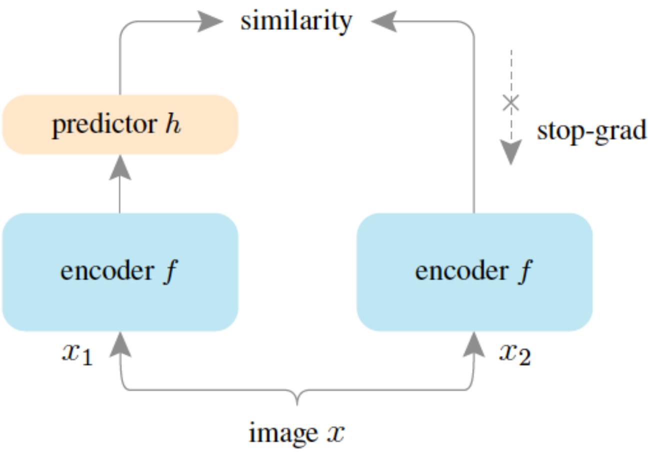
monitor 1: std of p



monitor 2: KNN classifier

Predictor is Important

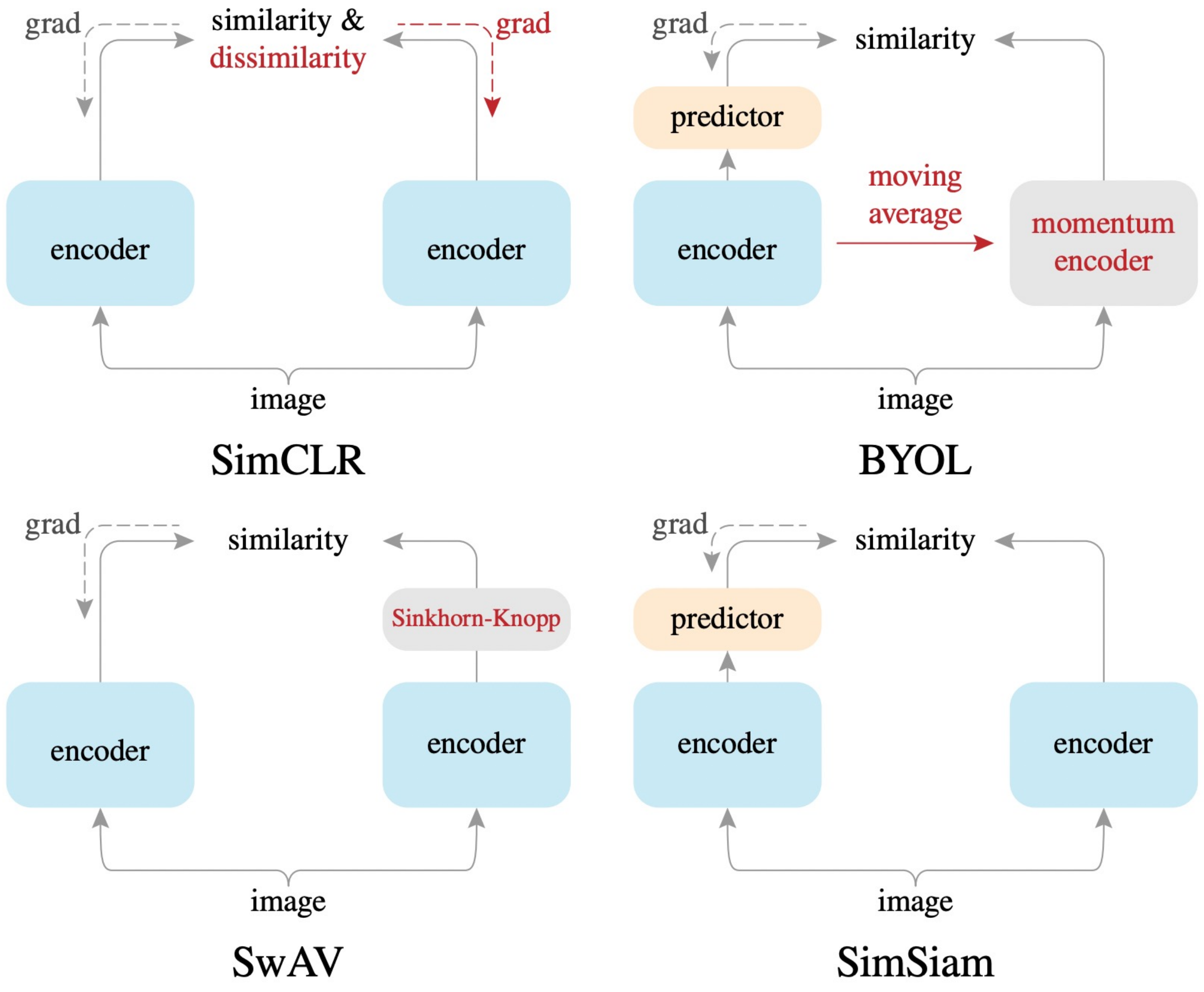
- Tried different settings:



setting	top-1	
previous default	67.7	
w/o predictor	0.1	effectively w/o stop-grad: symmetrized loss
random predictor	1.5	does not converge
not decay predictor ***lr***	68.1	**default** for comparisons

- Not crucial: predictor **can** be removed without collapsing

Comparison to Other Siamese Learning

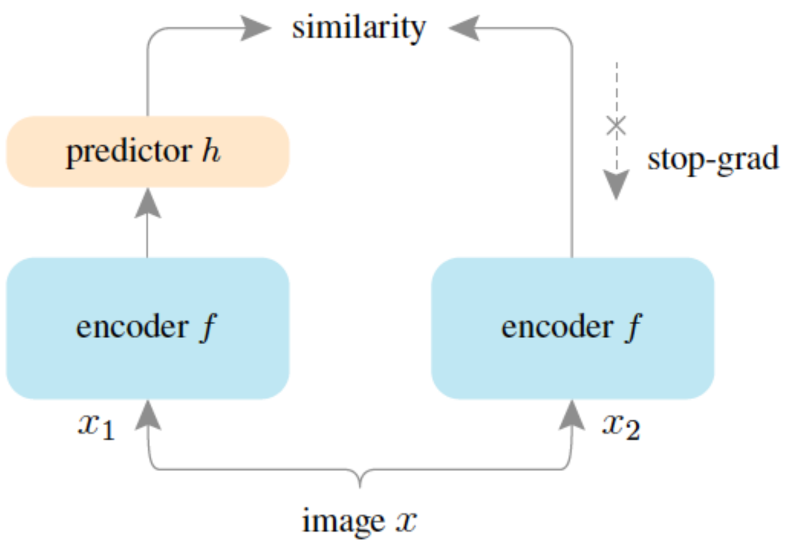


- *Momentum encoder*
  - Exponential Moving Average on encoder weights
- *Sinkhorn-Knopp*
  - online clustering algorithm that balances cluster assignments

[Grill et al, NeurIPS 2020] [Caron et al, NeurIPS 2020]

SimSiam Simplifies Siamese Learning

- SimCLR *w/o* negatives
- SwAV *w/o* online clustering
- BYOL *w/o* momentum encoder
- MoCo *w/o* negatives or momentum encoder



Comparisons to Others, ImageNet

method	batch size	negative pairs	momentum encoder	100-ep	200-ep	400-ep	800-ep
SimCLR	4096	✅		66.5	68.3	69.8	70.4
MoCo	256	✅	✅	67.4	69.9	71.0	72.2
BYOL	4096		✅	66.5	70.6	73.2	74.3
SwAV	4096			66.5	69.1	70.7	71.8
SimSiam	256			68.1	70.0	70.8	71.3

- SimSiam is batch size friendly, momentum encoder free, and competitive

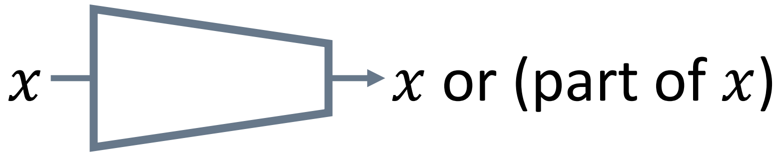
Comparisons to Others, VOC Detection

Pre-train	AP50	AP75	AP
Supervised	74.4	42.4	42.7
SimCLR	75.9	46.8	50.1
MoCo	77.1	48.5	52.5
BYOL	77.1	47.0	49.9
SwAV	75.5	46.5	49.6
SimSiam (Optimal)	77.3	48.5	52.5

- All methods generally perform well, and *outperform* ImageNet supervised pre-training

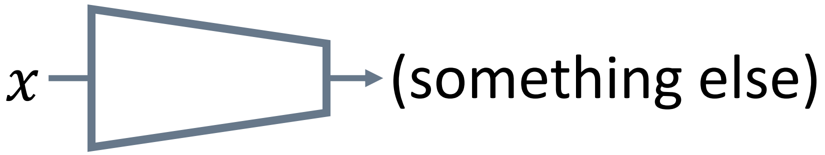
Paradigms for Self-Supervised Learning

- Reconstructive / Autoencoding



1. **M**asked **A**uto-**E**ncoders

- Non-Reconstructive



2. **Sim**ple **Siam**ese

Question: Is Contrastive learning reconstructive? Why?

Xinlei Chen
xinleic@meta.com