Self-Supervised Learning in Vision

From Research Advances to Best Practices

ICML 2023 Tutorial Part 1-A

Xinlei Chen

facebook
Artificial Intelligence Research

Self-Supervised Learning

- Pre-train representations <u>without labels</u> for downstream tasks

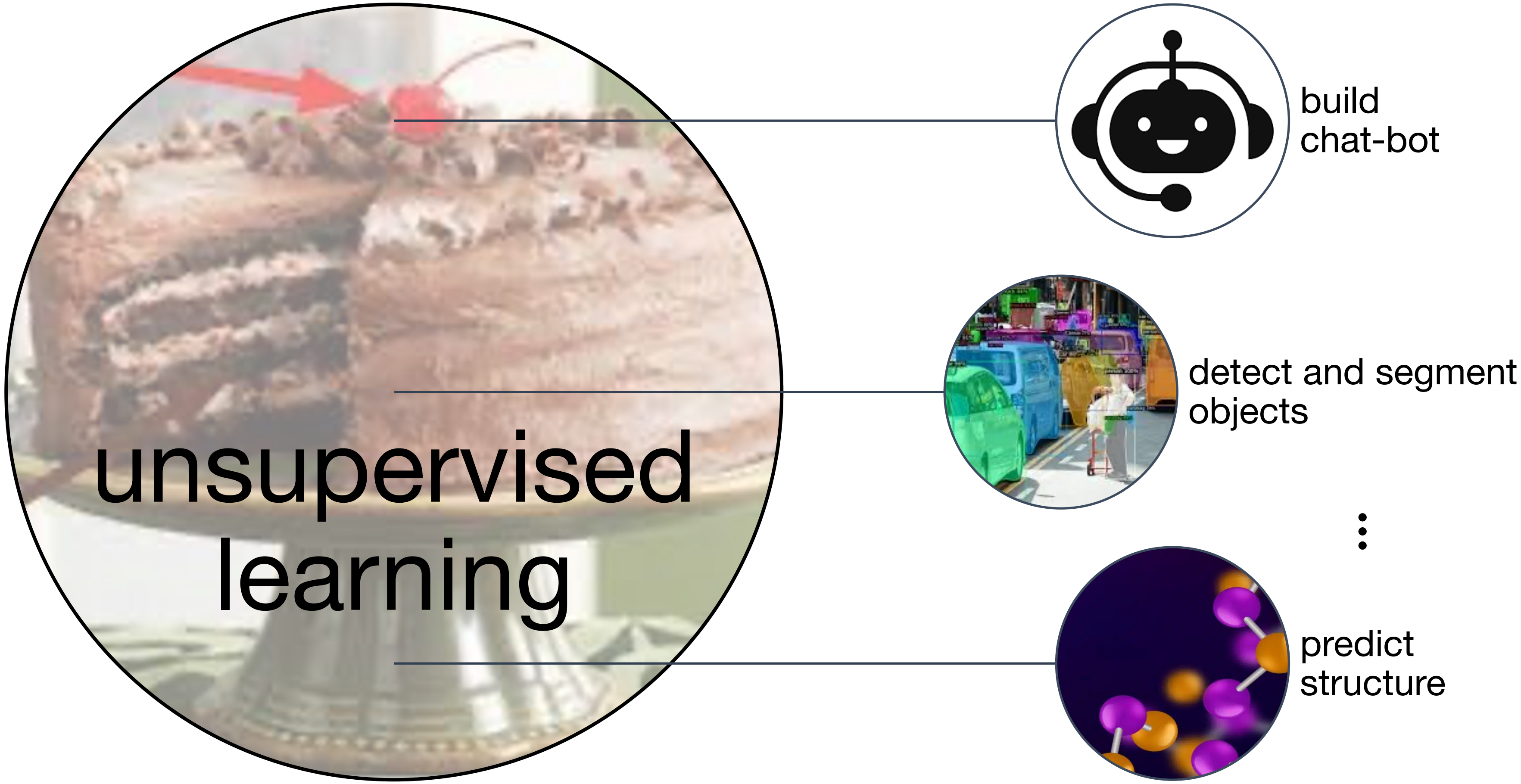
Self-Supervised Learning

- Pre-train representations without labels for downstream tasks

Yann:

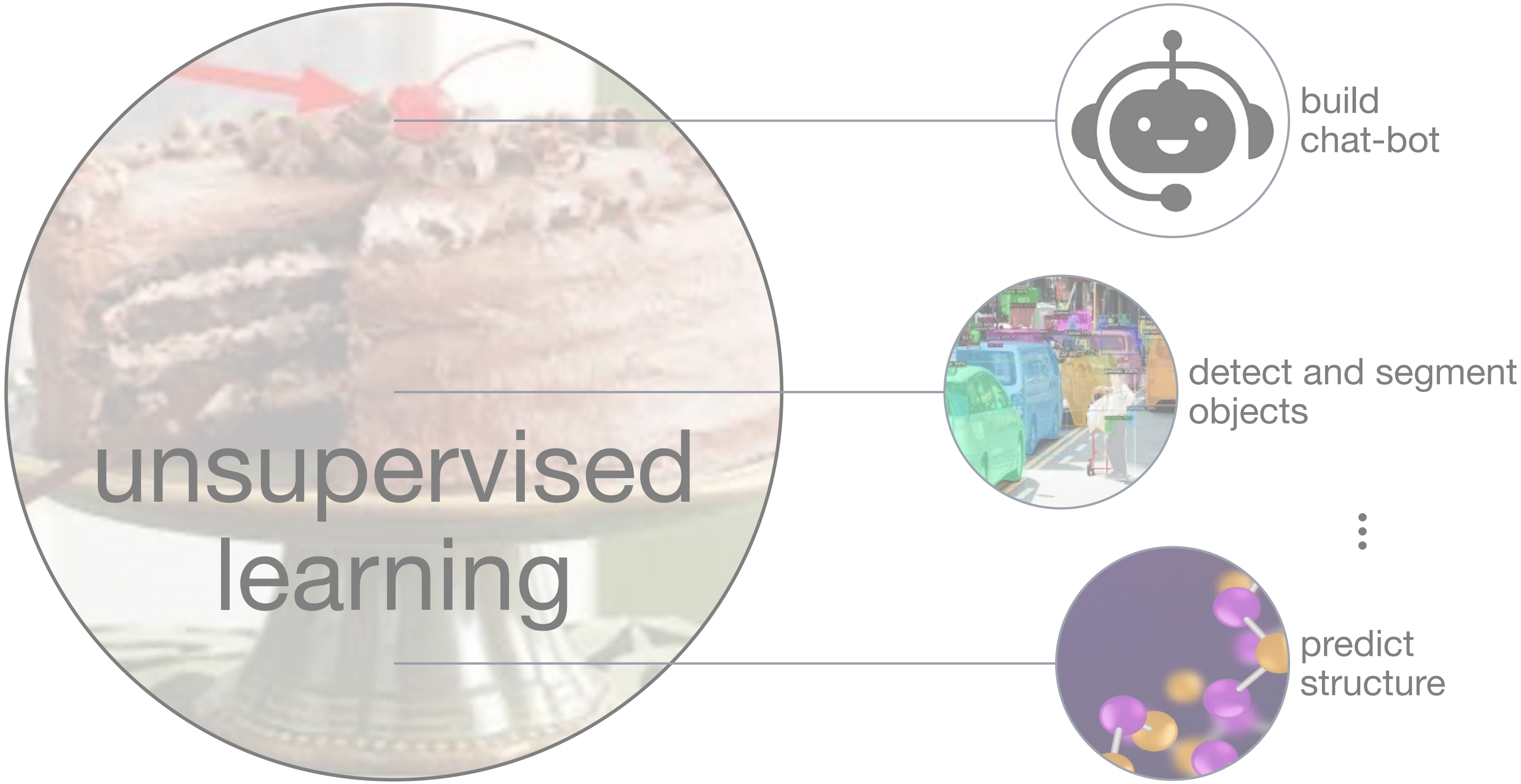
Self-Supervised Learning

- Pre-train representations without labels for downstream tasks



Self-Supervised Representation Learning

- Pre-train representations without labels for downstream tasks



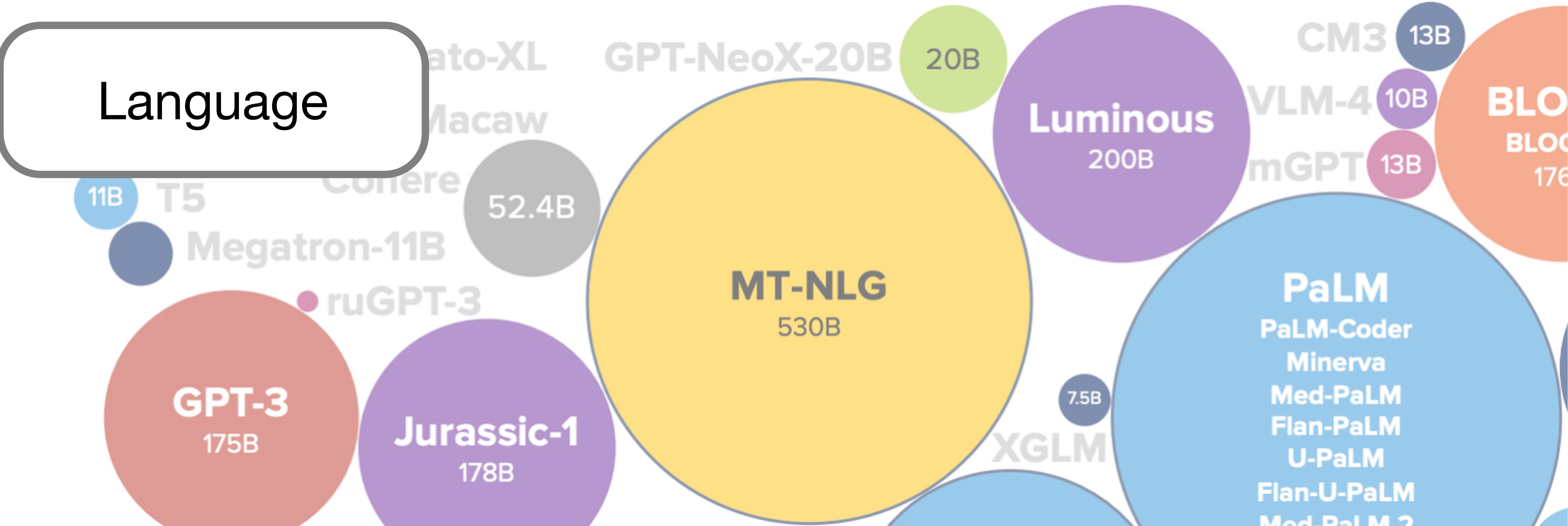
Self-Supervised Representation Learning

- Scalable: train huge models on unlimited data and not worry about overfitting

[Devlin et al, NAACL 2019] [Brown et al, NeurIPS 2020]

Self-Supervised Representation Learning

- **Scalable:** train huge models on unlimited data and not worry about overfitting



[Devlin et al, NAACL 2019] [Brown et al, NeurIPS 2020]

Self-Supervised Representation Learning

- **Scalable:** train huge models on unlimited data and not worry about overfitting

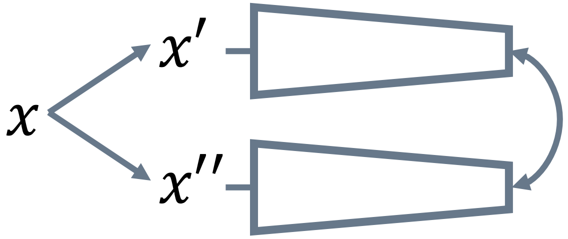
Language

Vision

[Chen et al, ICML 2020] [He et al, CVPR 2020] [Grill et al, NeurIPS 2020] [Caron et al, NeurIPS 2020]

Self-Supervised Paradigms in Vision

- Contrastive / Siamese

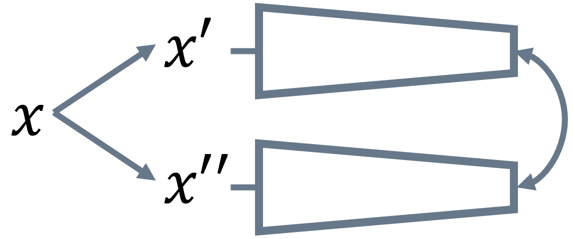


  - Compare data points in the latent *representation* space
  - Computer vision: SimCLR, MoCo, BYOL, DINO, …, with *augmentations*
  - Covered in Part II of this tutorial

[He et al, CVPR 2022]

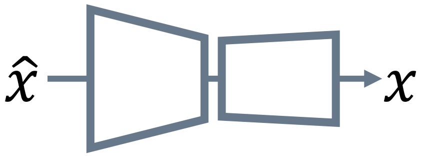
Self-Supervised Paradigms in Vision

- Contrastive / Siamese



→ Covered in Part II

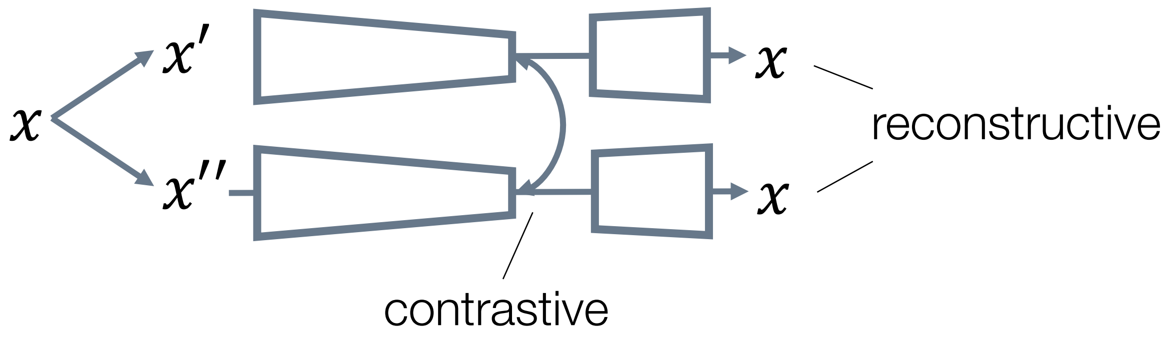
- Reconstructive / Auto-Encoding



  - Reconstruct *corrupted* data points
  - *Grounded* in the input space
  - Paradigm of BERT & GPT in NLP
  - Computer Vision: MAE

Self-Supervised Paradigms in Vision

- "Contrastive + Reconstructive" is also possible

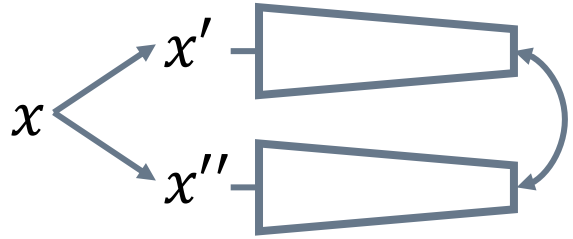


  - Multi-tasking makes representations more *versatile*: iBOT, MAGE
  - But the pipeline is *less clean* to understand scientifically

[He et al, CVPR 2022]

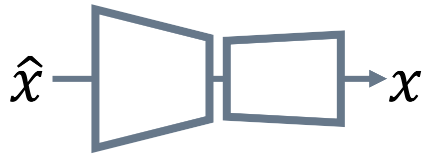
Self-Supervised Paradigms in Vision

- Contrastive / Siamese



→ Covered in Part II

- Reconstructive / Auto-Encoding



→ Covering *now* in Part I

  - Xinlei: MAE – reconstructive on images
  - Christoph: SSL on Videos

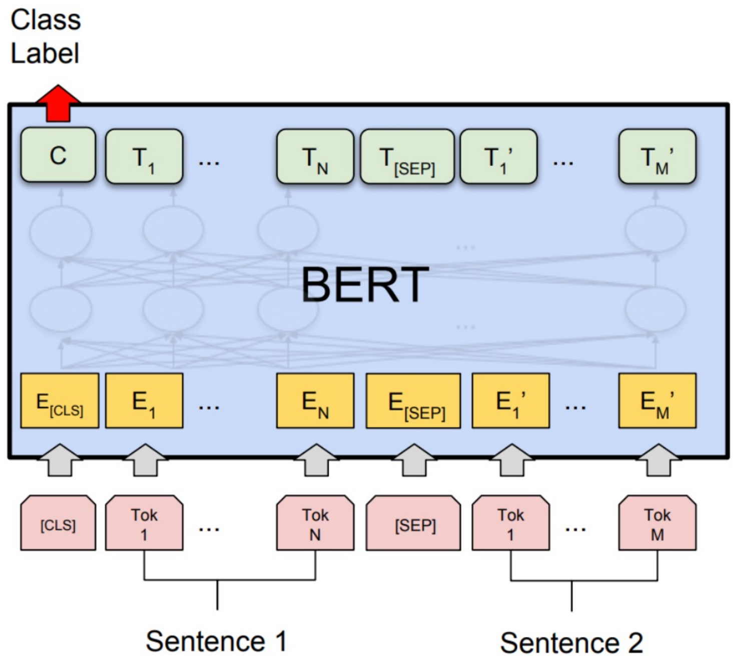
[He et al, CVPR 2022]

What is Masked Auto-Encoding (MAE)?

- Very simple method, but highly effective

What is Masked Auto-Encoding (MAE)?

- Very simple method, but highly effective
- BERT-like masked modeling objective, but with crucial design changes for computer vision

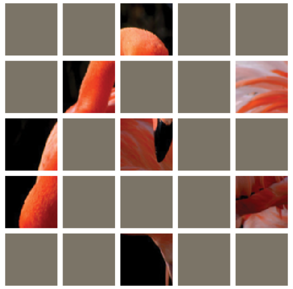


[Devlin et al, NAACL 2019] [He et al, CVPR 2022]

What is Masked Auto-Encoding (MAE)?

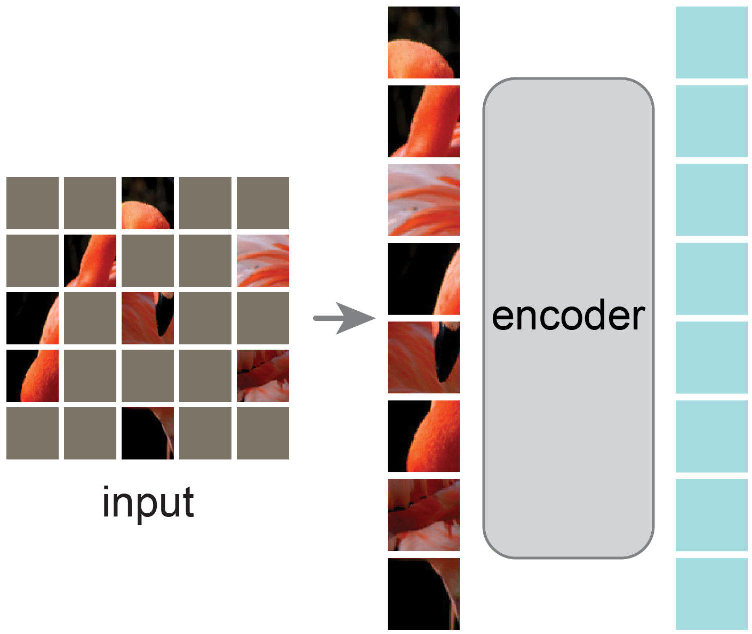
- Very simple method, but highly effective
- BERT-like masked modeling objective, but with crucial design changes for computer vision
- Intriguing properties – better scalability and more

How MAE Works?



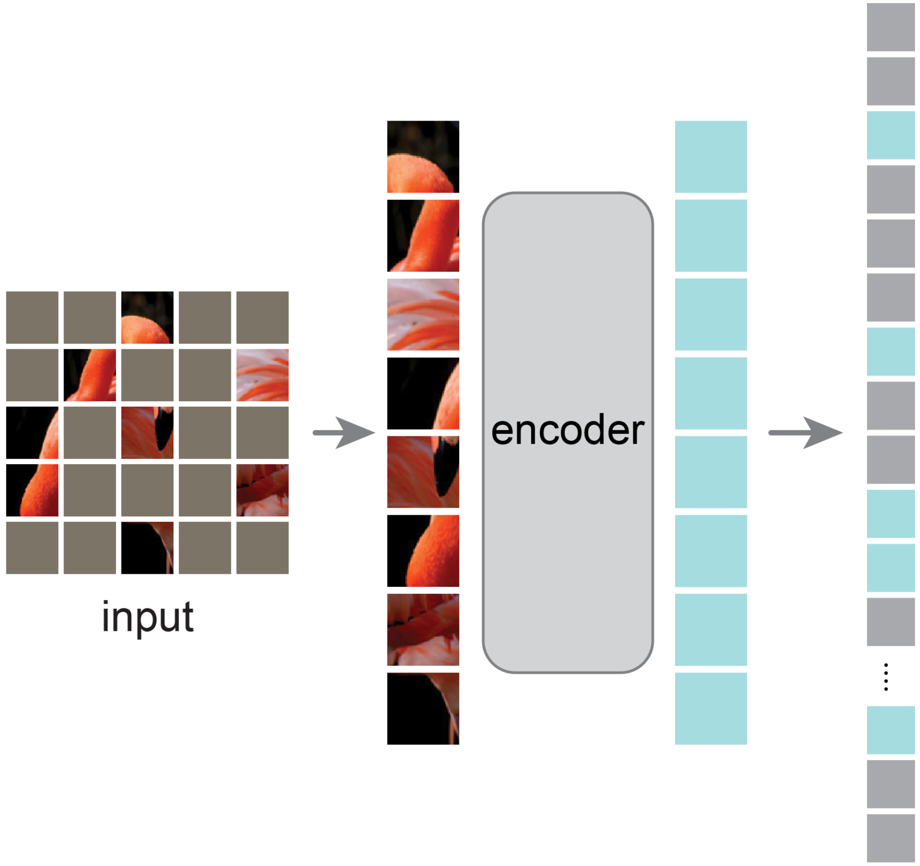
Random masking

How MAE Works?



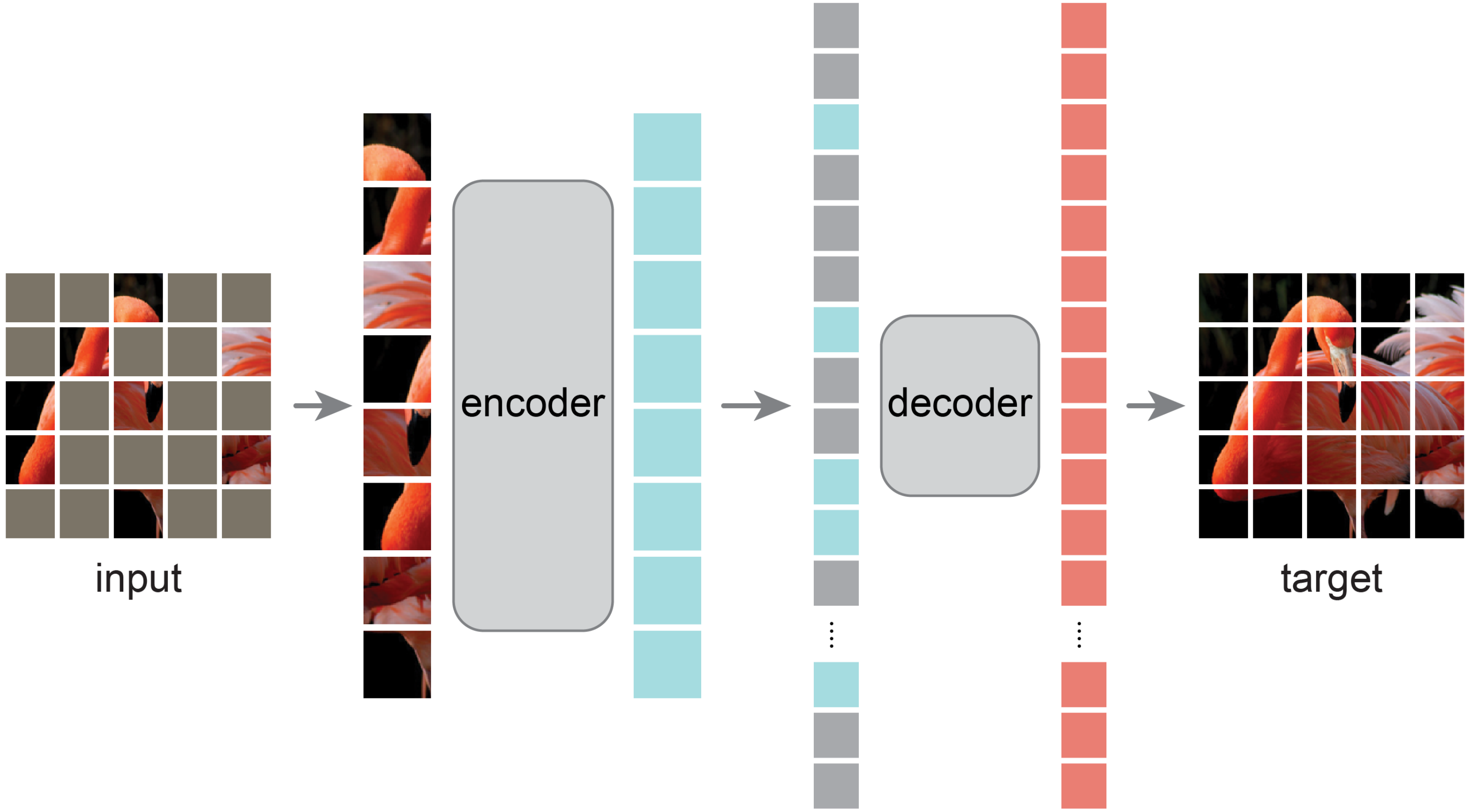
Encode visible patches

How MAE Works?



Add mask tokens

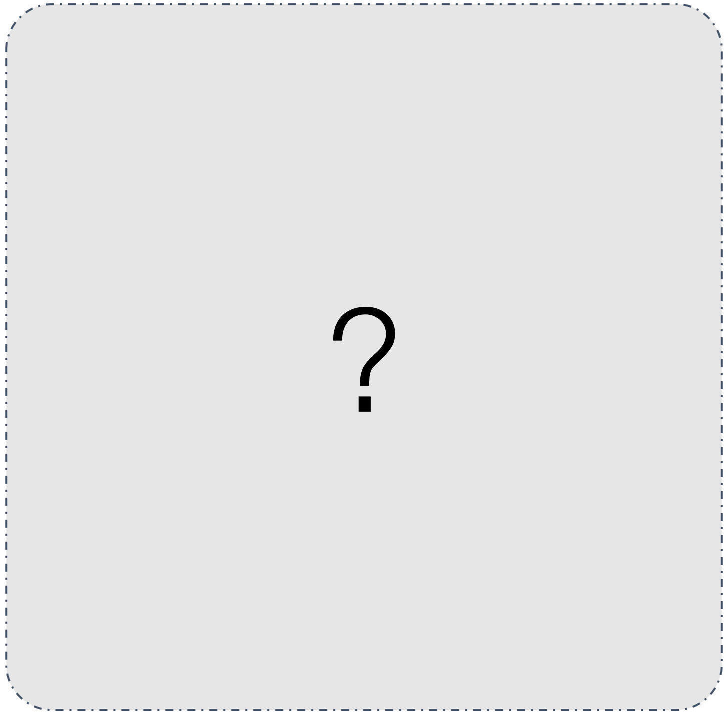
How MAE Works?



Reconstruct

MAE Reconstruction Example

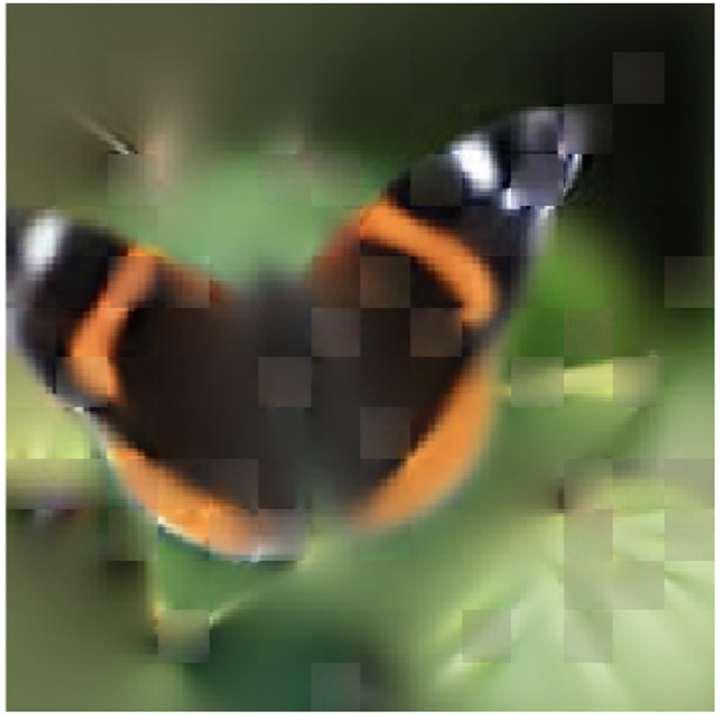
Masked input: 80%



You guess?

MAE Reconstruction Example

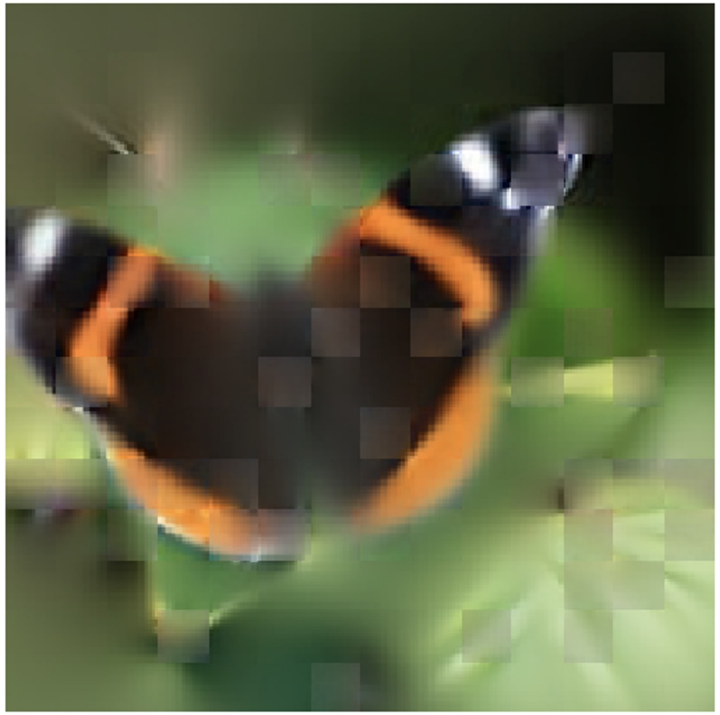
Masked input: 80%



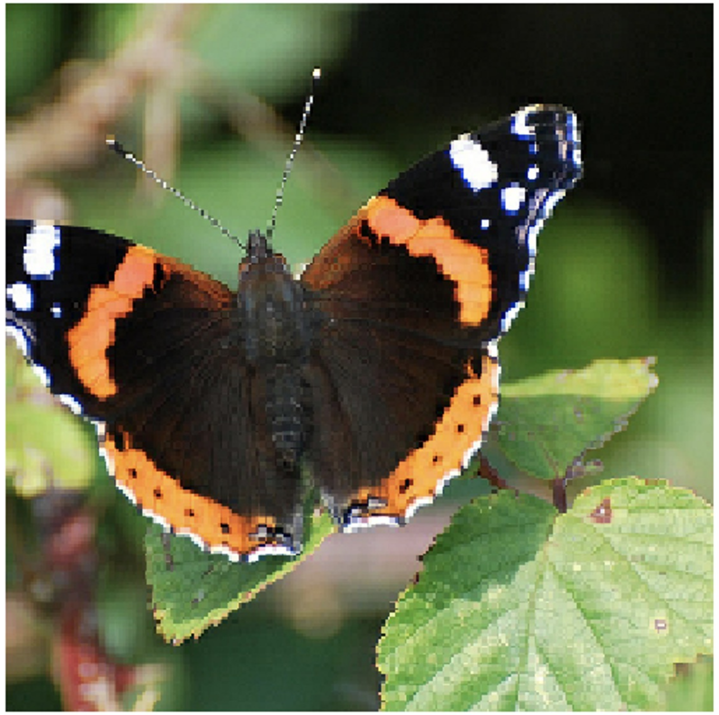
MAE's guess

MAE Reconstruction Example

Masked input: 80%

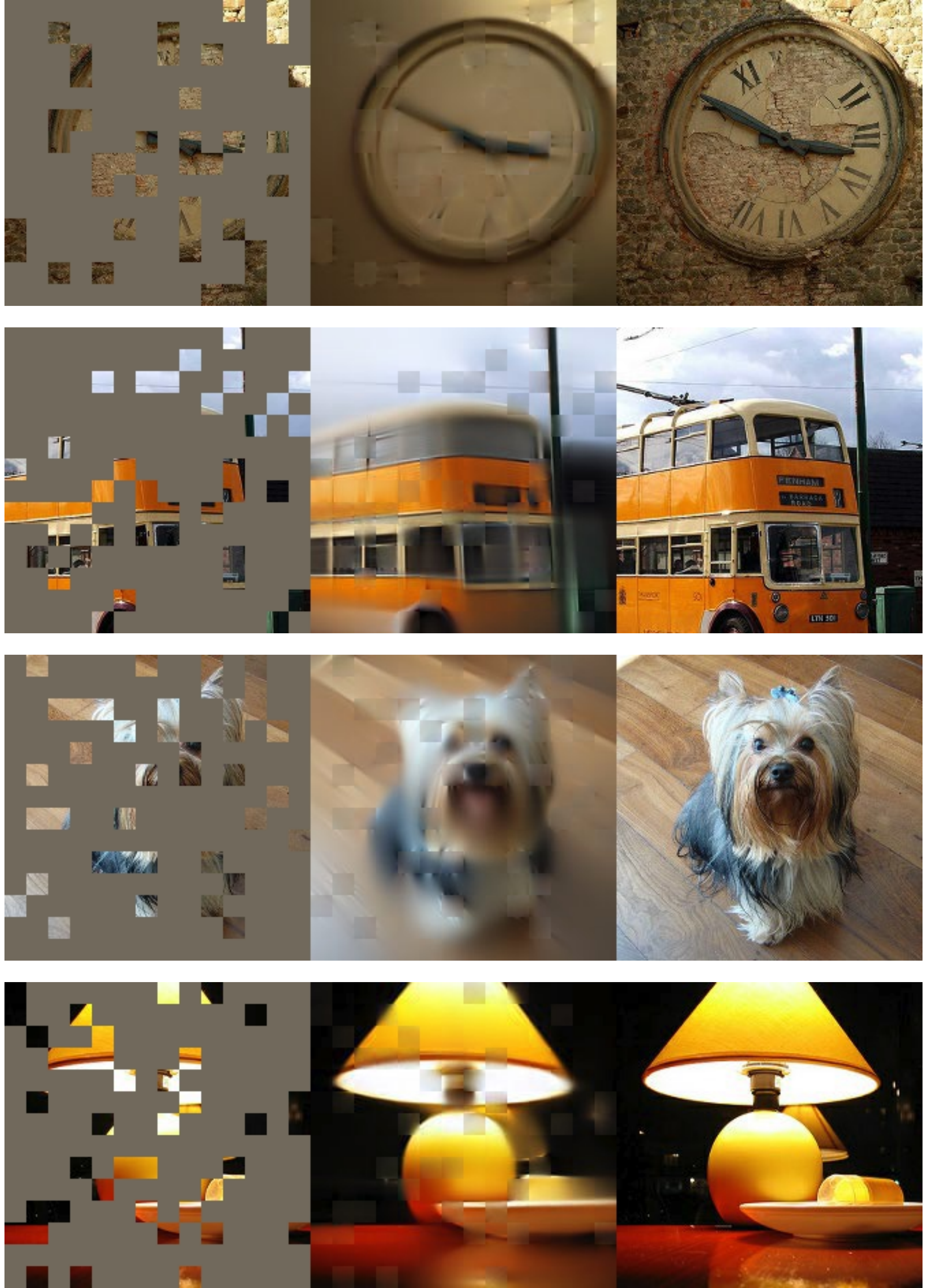


MAE's guess

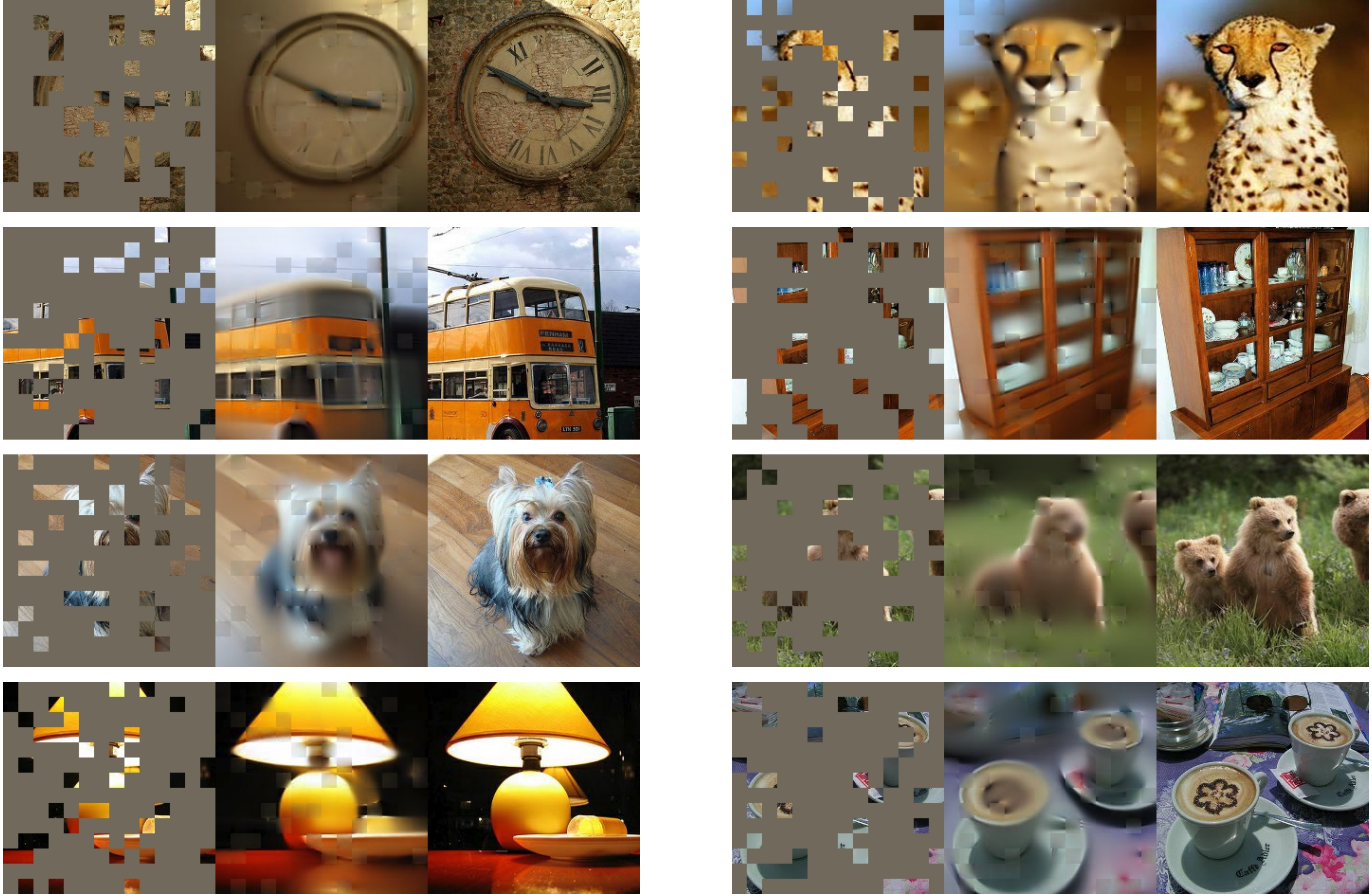


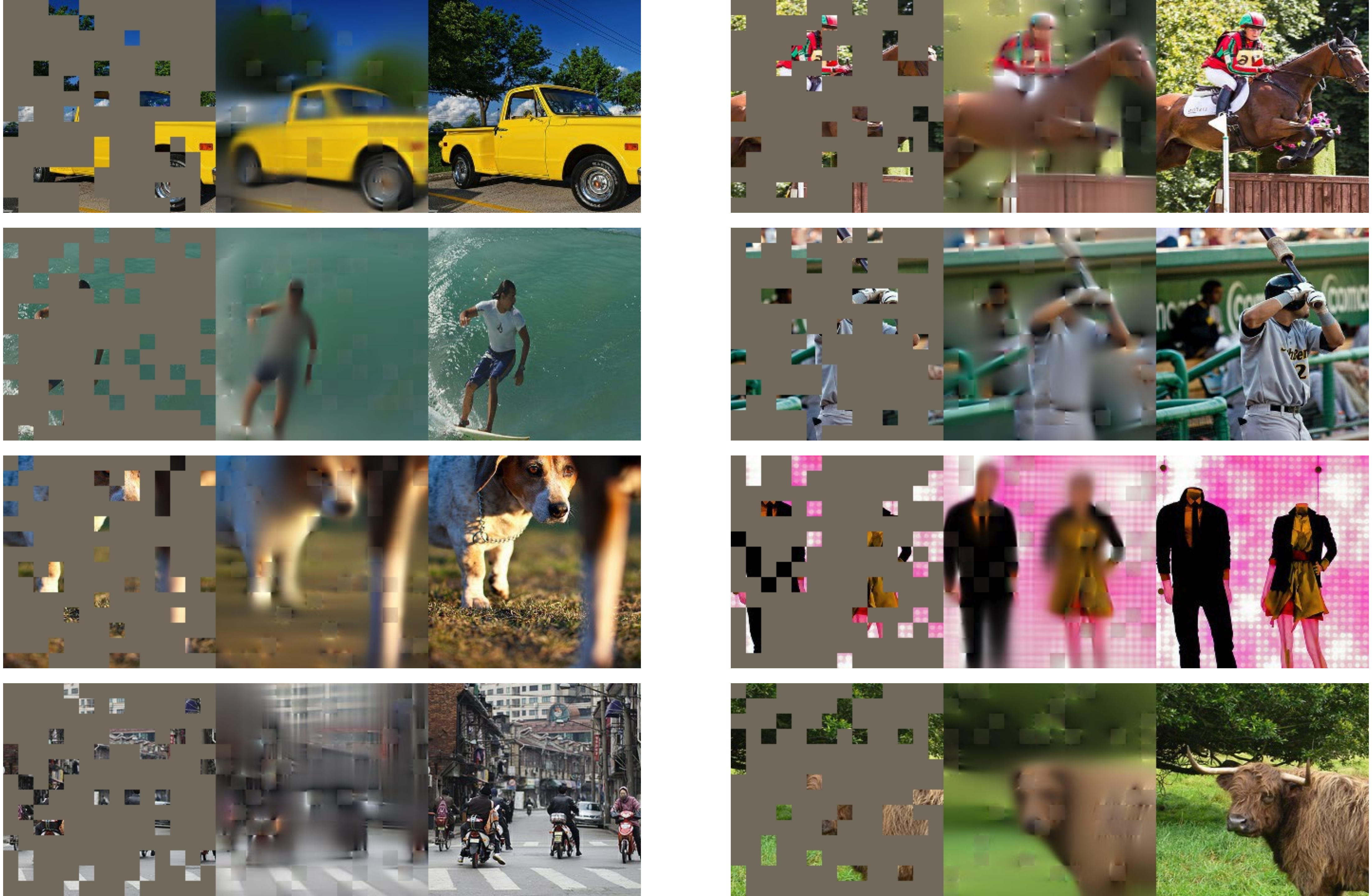
Ground truth

ImageNet val set (unseen)



ImageNet val set (unseen)





COCO val set (unseen)

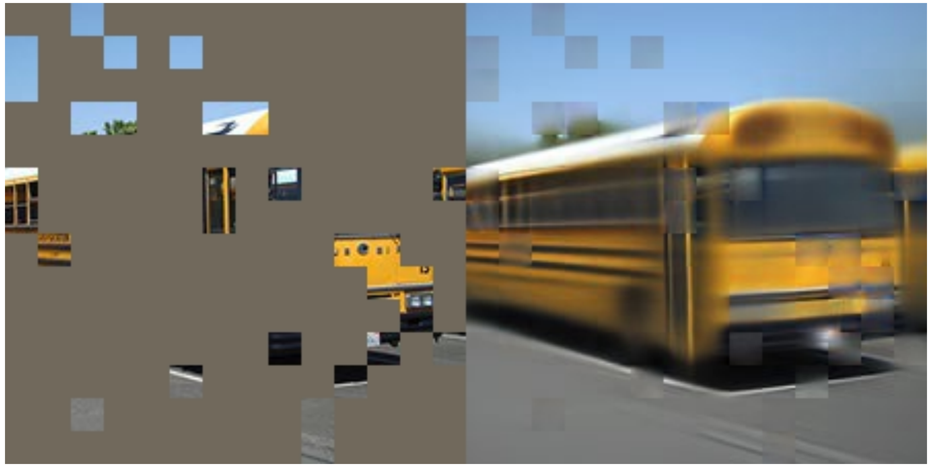
MAE Can Generalize

75% mask

original

75% mask

original



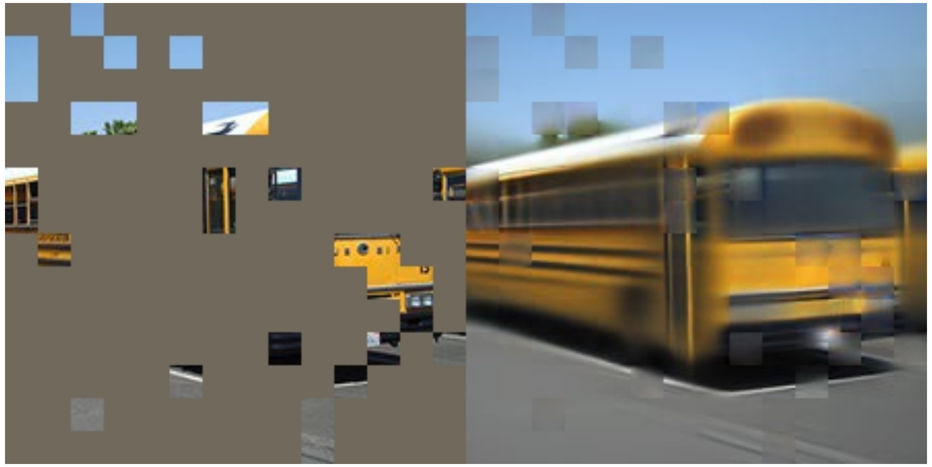
85% mask

MAE Can Generalize

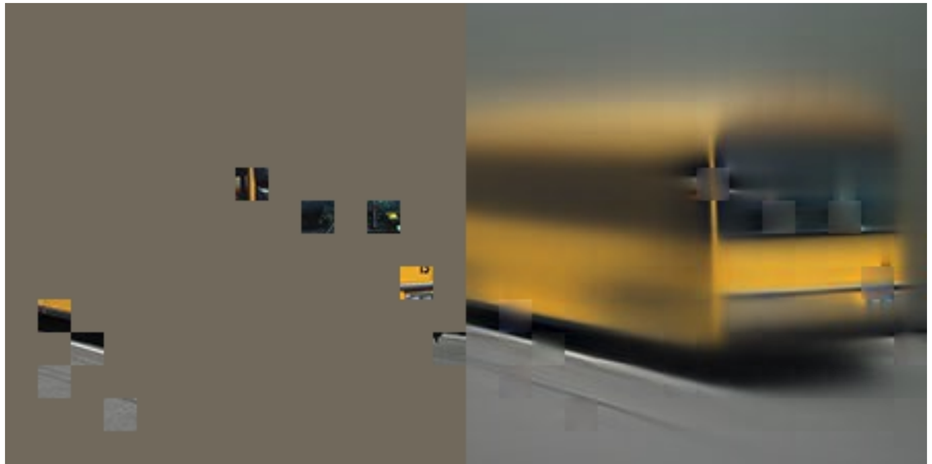
MAE Can Generalize

original

75% mask



85% mask



95% mask

MAE Can Generalize

original

75% mask

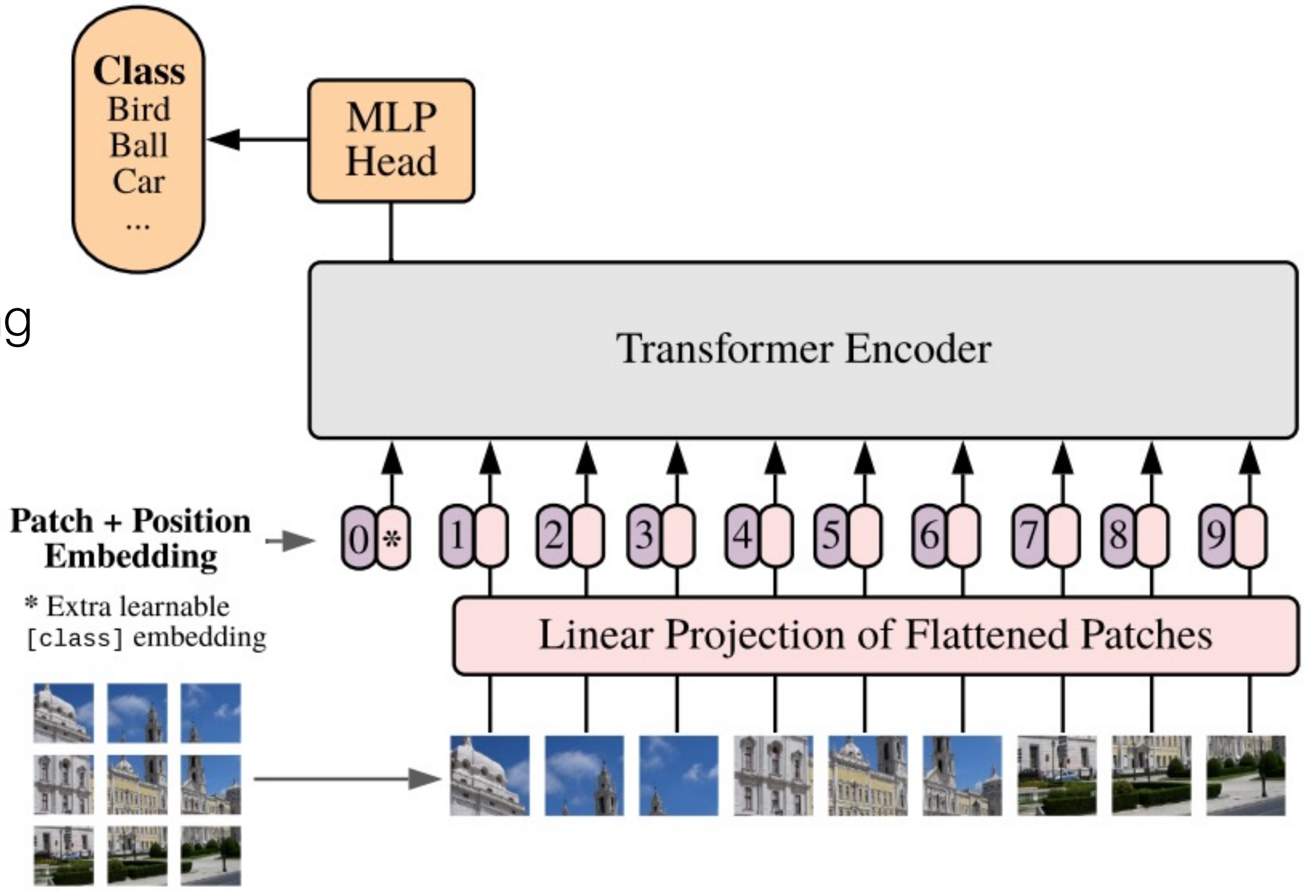
85% mask

95% mask

[Dosovitskiy et al, ICLR 2021]

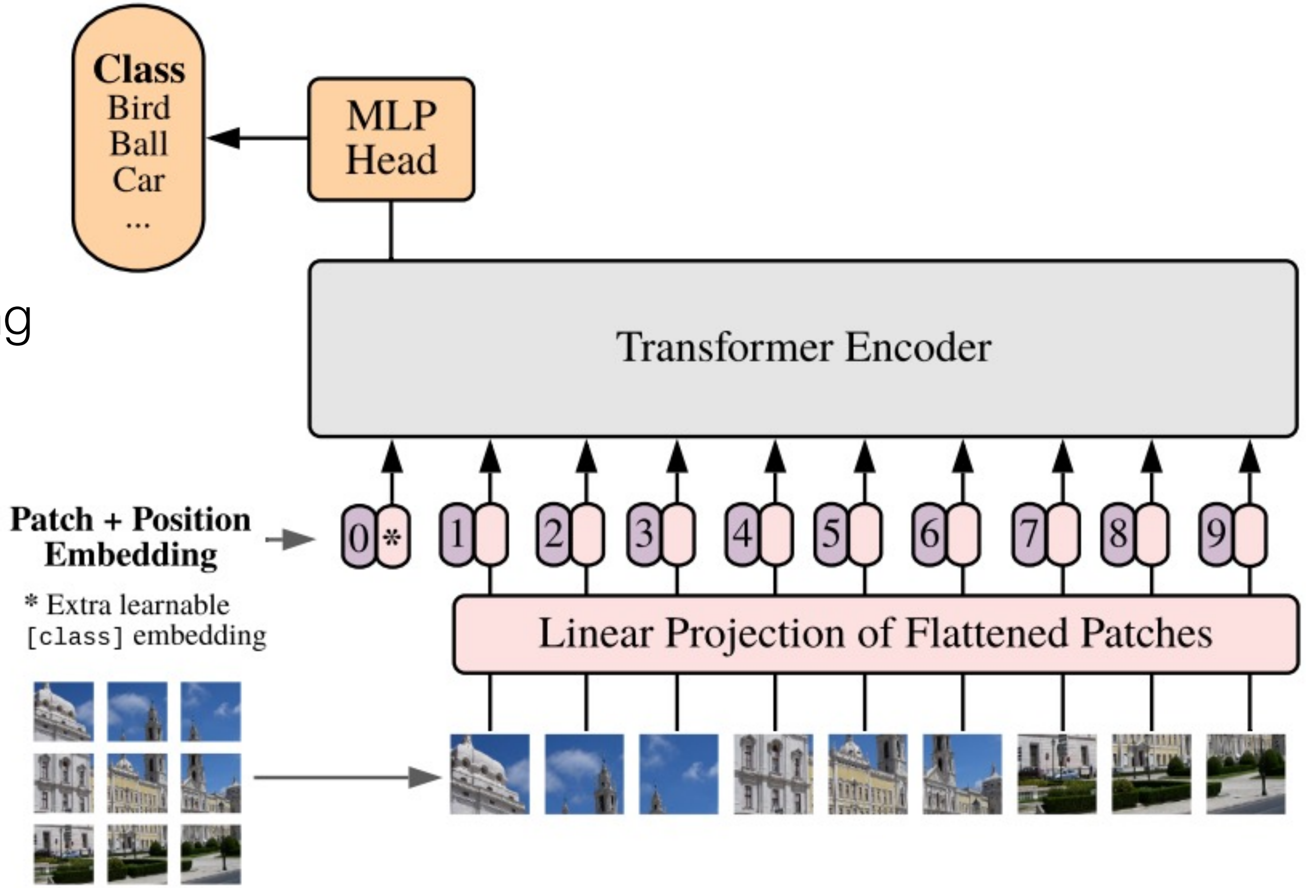
BERT-like: Transformers

- Vision Transformer (ViT)
    - Less inductive bias
    - Non-overlapping tokenization
        - Easier for masked auto-encoding



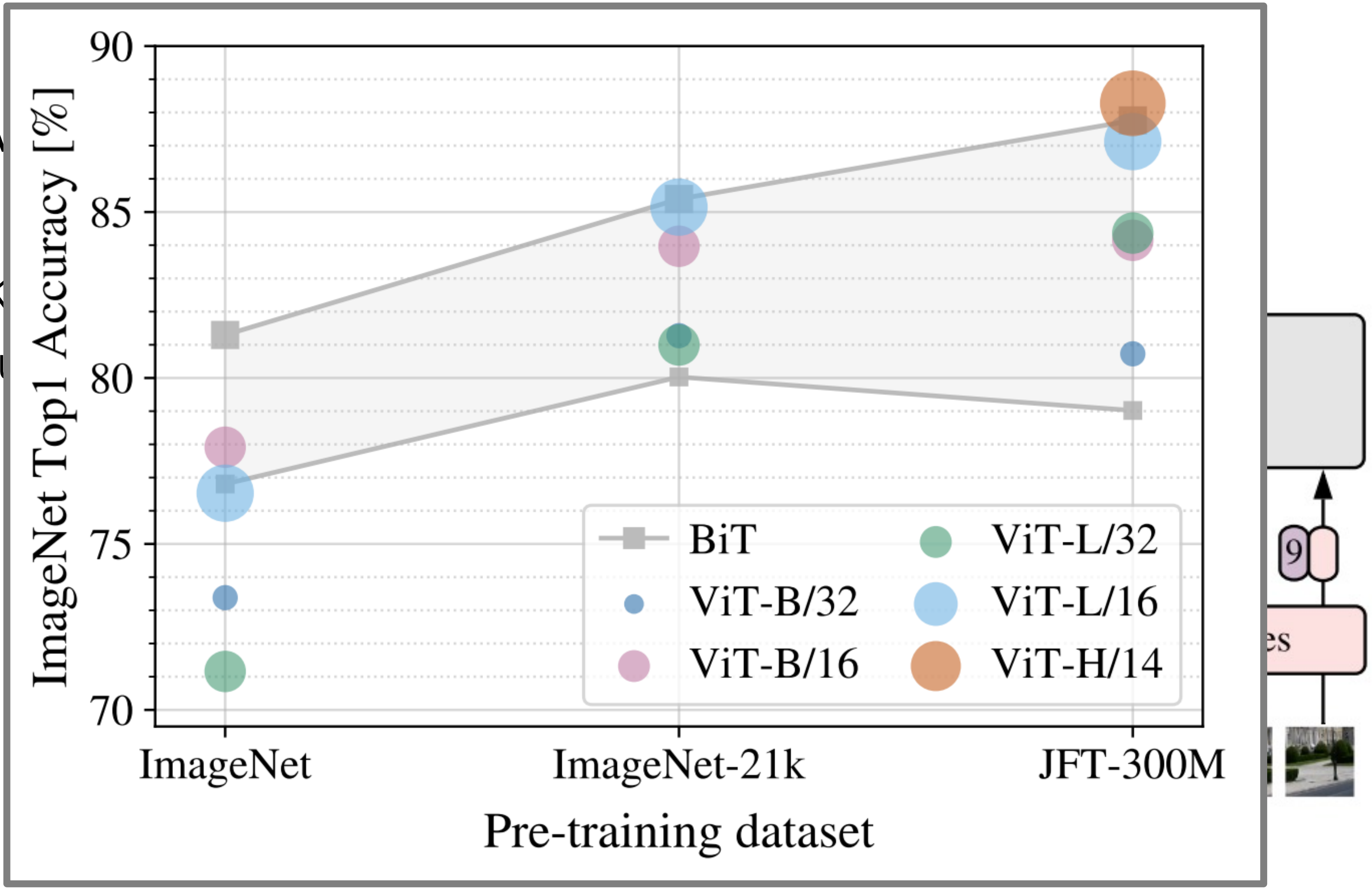
BERT-like: Transformers

- Vision Transformer (ViT)
  - Less inductive bias
  - Non-overlapping tokenization
    - Easier for masked auto-encoding
- *Scalable*
  - with larger models
  - on larger datasets



BERT-like: Transformers

- Vision Transformer (V
  - Less inductive bias
  - Non-overlapping tok
    - Easier for masked au
- *Scalable*
  - with larger models
  - on larger datasets



[Devlin et al, NAACL 2019]

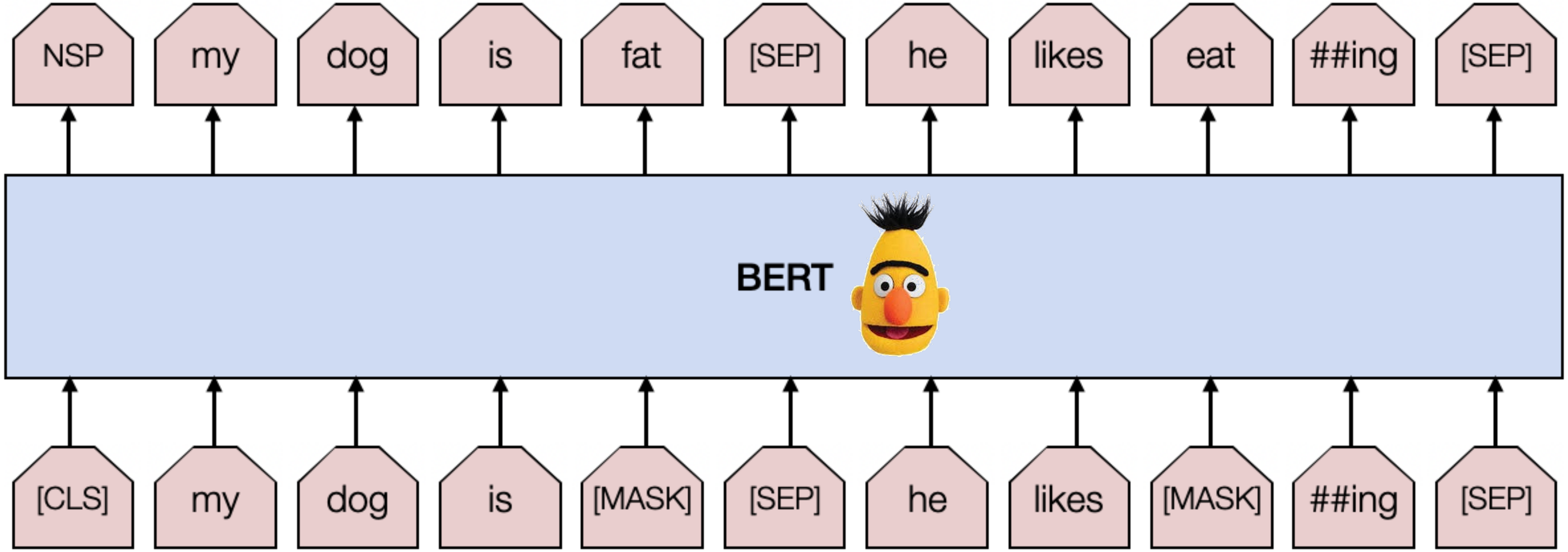
BERT-*unlike*: Mask Ratio

- BERT: 15% is enough to create a challenging task
- MAE: a high ratio of 75% - 80% to be meaningful

[Devlin et al, NAACL 2019]

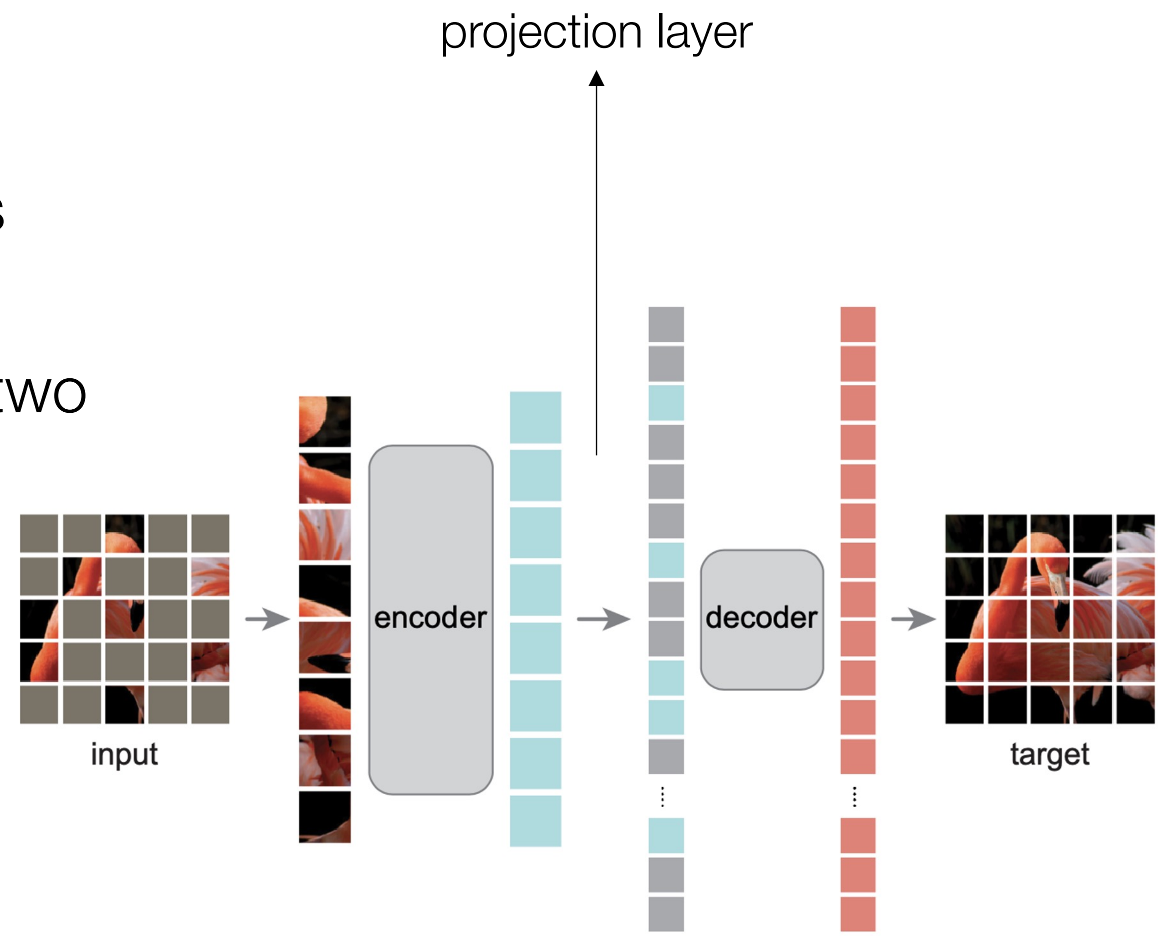
BERT-*unlike*: Encoder-Decoder

- BERT: encoder-*only* pre-training



BERT-*unlike*: Encoder-Decoder

- MAE:
  - *Large* encoder on *visible* tokens
  - Small decoder on *all* tokens
  - Projection layer to connect the two

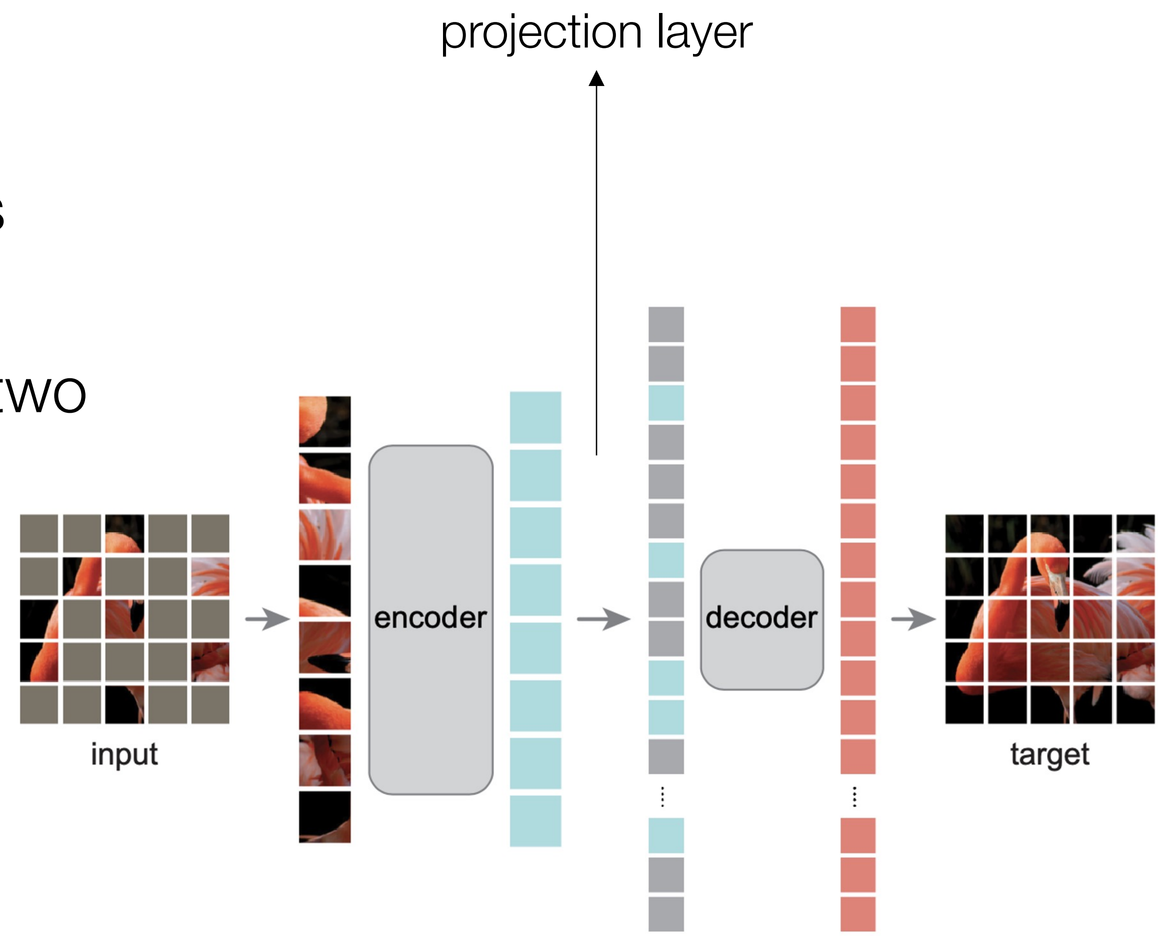


[Devlin et al, NAACL 2019]

BERT-*unlike*: Encoder-Decoder

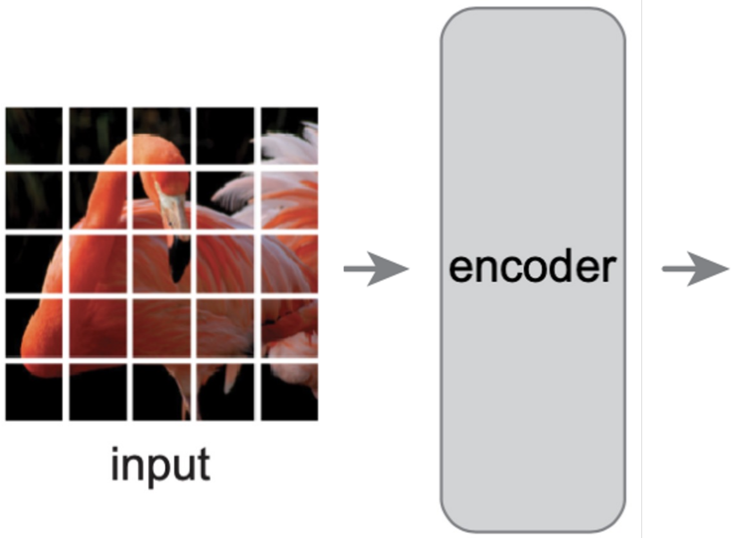
- MAE:
  - *Large* encoder on *visible* tokens
  - Small decoder on *all* tokens
  - Projection layer to connect the two

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



MAE for Downstream Tasks: *Encoder Only*

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



[Deng et al, CVPR 2009]

Experimental Protocols

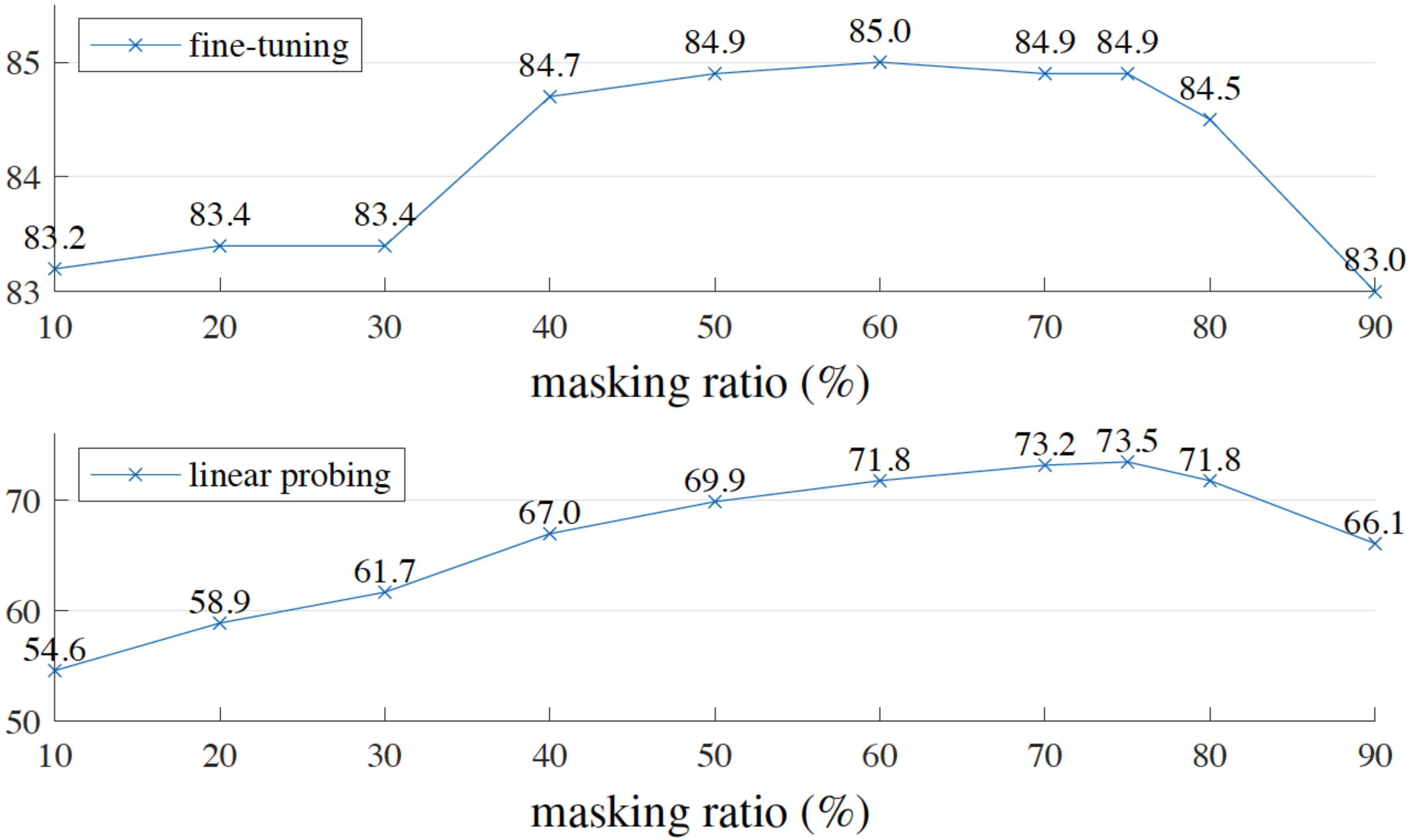
- Pre-training dataset: ImageNet-1K
- Architecture: ViT-*Large* encoder, 512-dim decoder

[Deng et al, CVPR 2009]

Experimental Protocols

- Pre-training dataset: ImageNet-1K
- Architecture: ViT-*Large* encoder, 512-dim decoder
- Transfer task: ImageNet-1K classification
  - “*ft*”: end-to-end tuning with MAE as an initialization
  - “*lin*”: linear probing, a single classifier on top of frozen encoder features

Analysis: Mask Ratio



Analysis: Decoder Size

- Encoder has 24-blocks, 1024-dimensional

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

Decoder depth

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

Decoder width

Analysis: Mask Token [M] 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 w/ [M] is default in BERT
- Big domain gap for linear probing
  - Pre-train sees 25% of the images only, while evaluation sees 100%

[Ramesh et al, ICML 2021] [Bao et al, ICLR 2022]

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

- Pixels with normalization: per-patch -- minus *mean*, divide by *std*
- PCA: only low-frequency component is retained
- dVAE token: from DALLE, expensive to compute

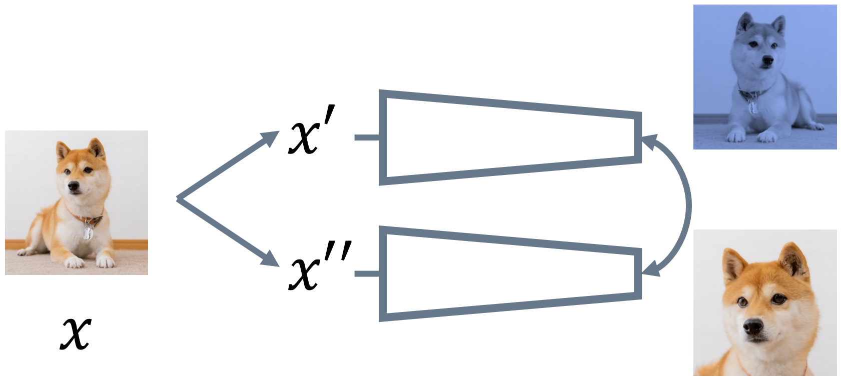
Analysis: Augmentations

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

Analysis: Augmentations

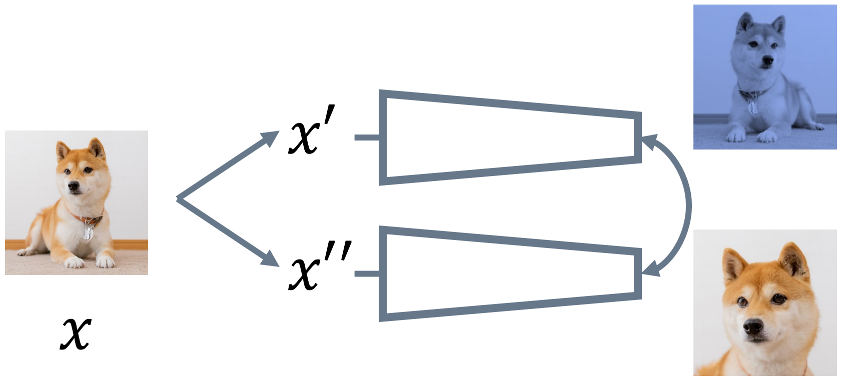
case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9



- MAE can work with minimal data augmentation
- For Contrastive / Siamese learning, augmentation is <u>crucial</u>

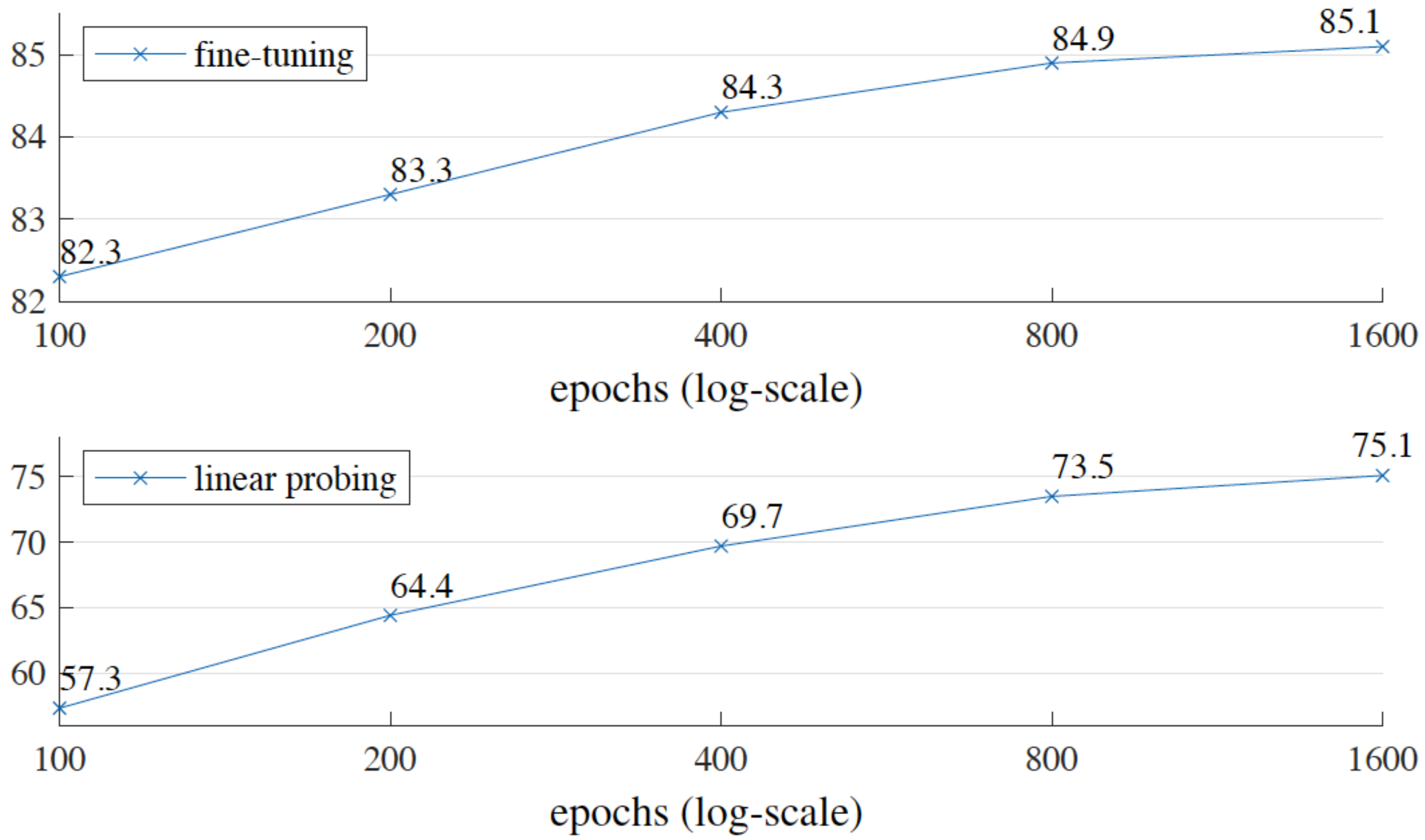
Analysis: Augmentations

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

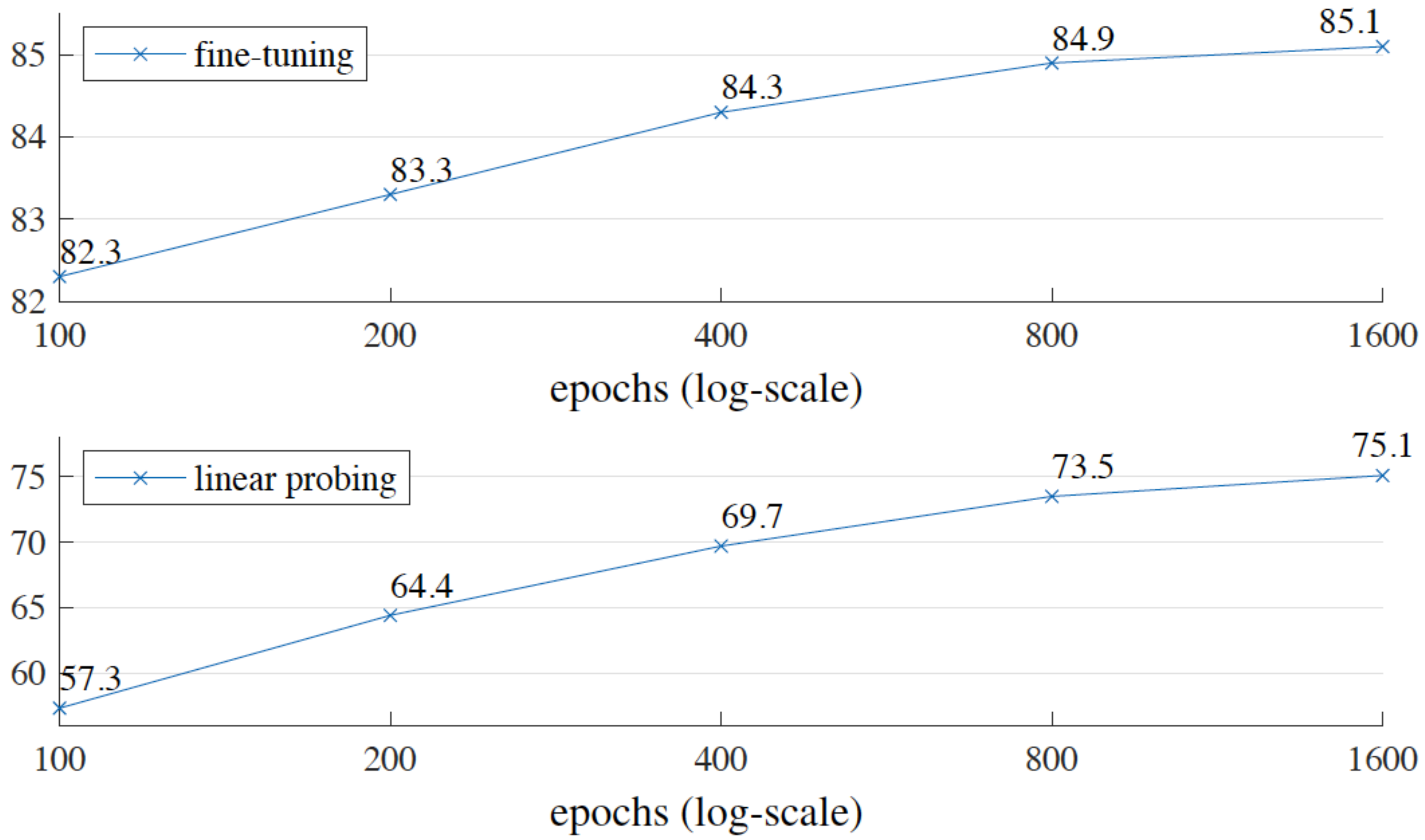


- MAE can work with minimal data augmentation
- For Contrastive / Siamese learning, augmentation is crucial
- Masking as a strong "augmentation": MSN, I-JEPA

Scalability: Longer Training

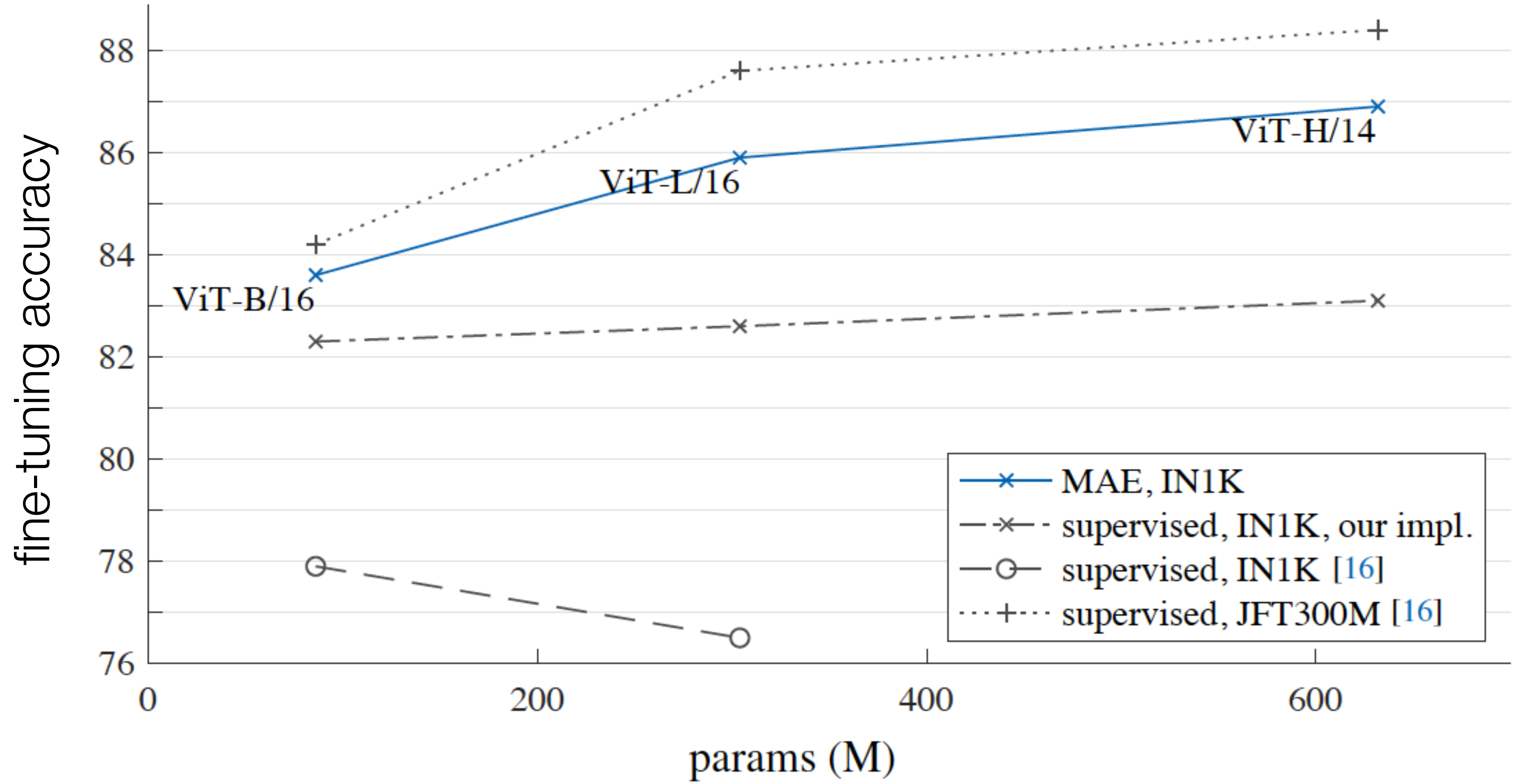


Scalability: Longer Training

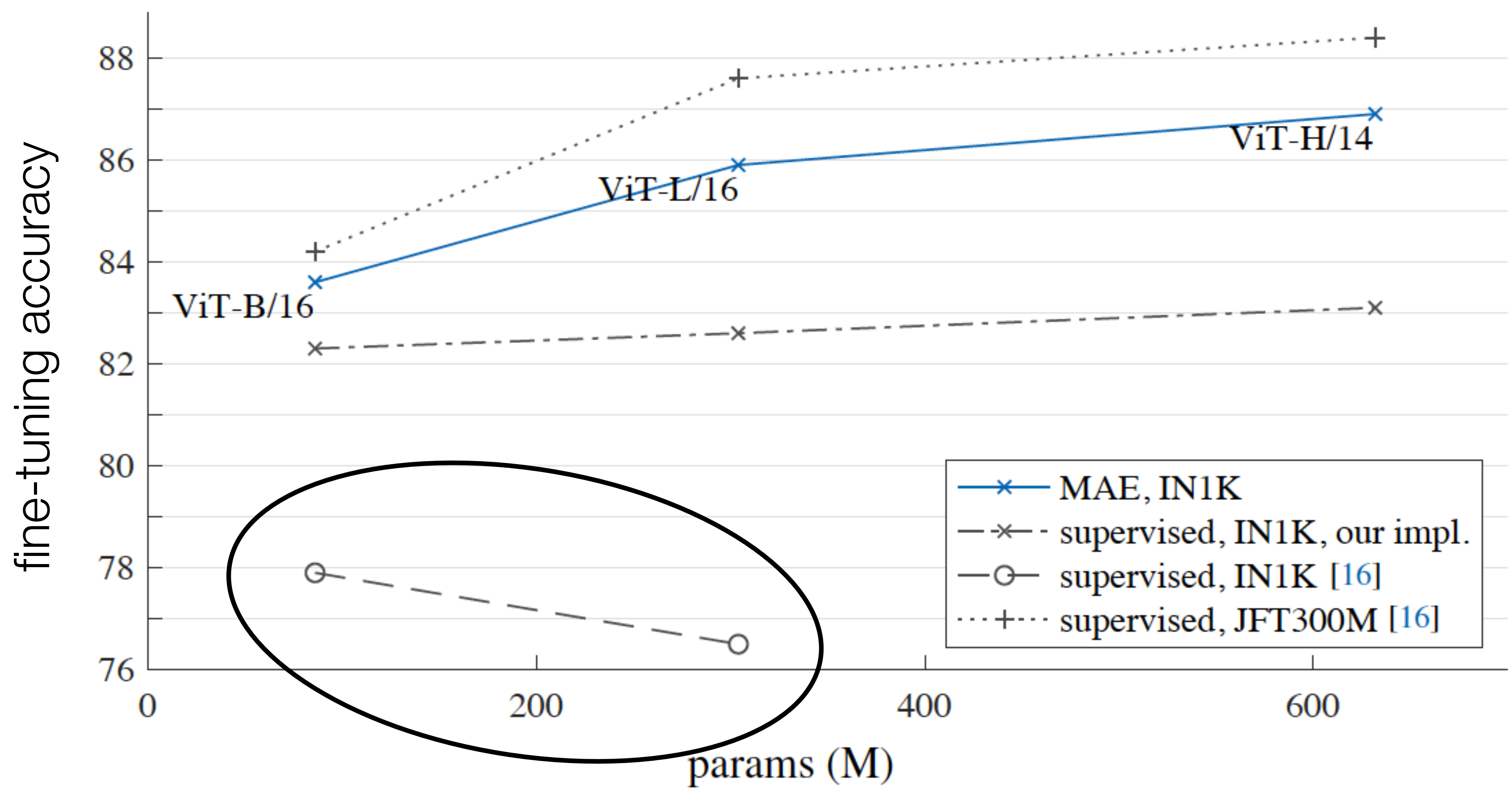


Wall-clock speed still efficient thanks to MAE design

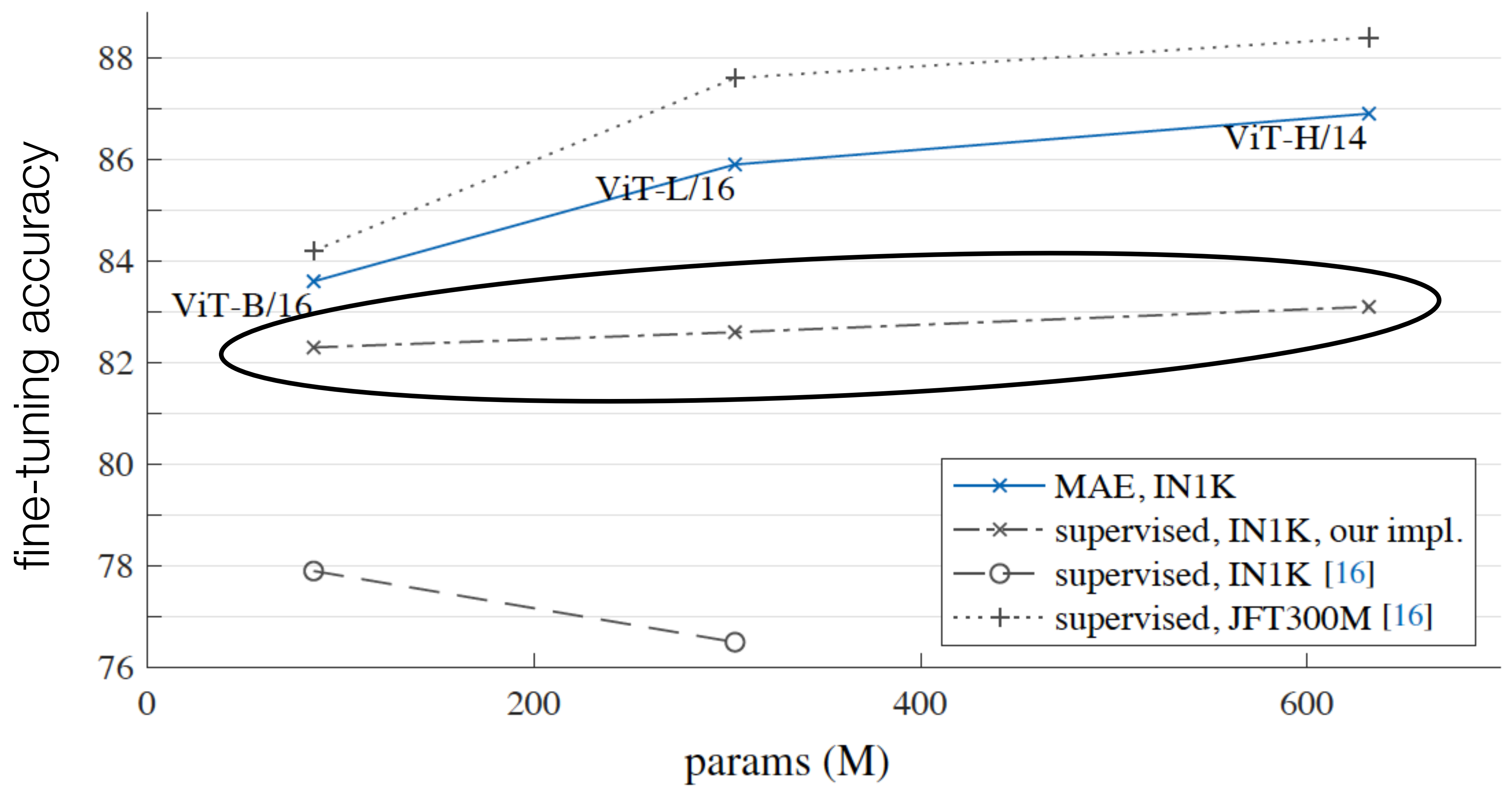
Scalability: Larger Models



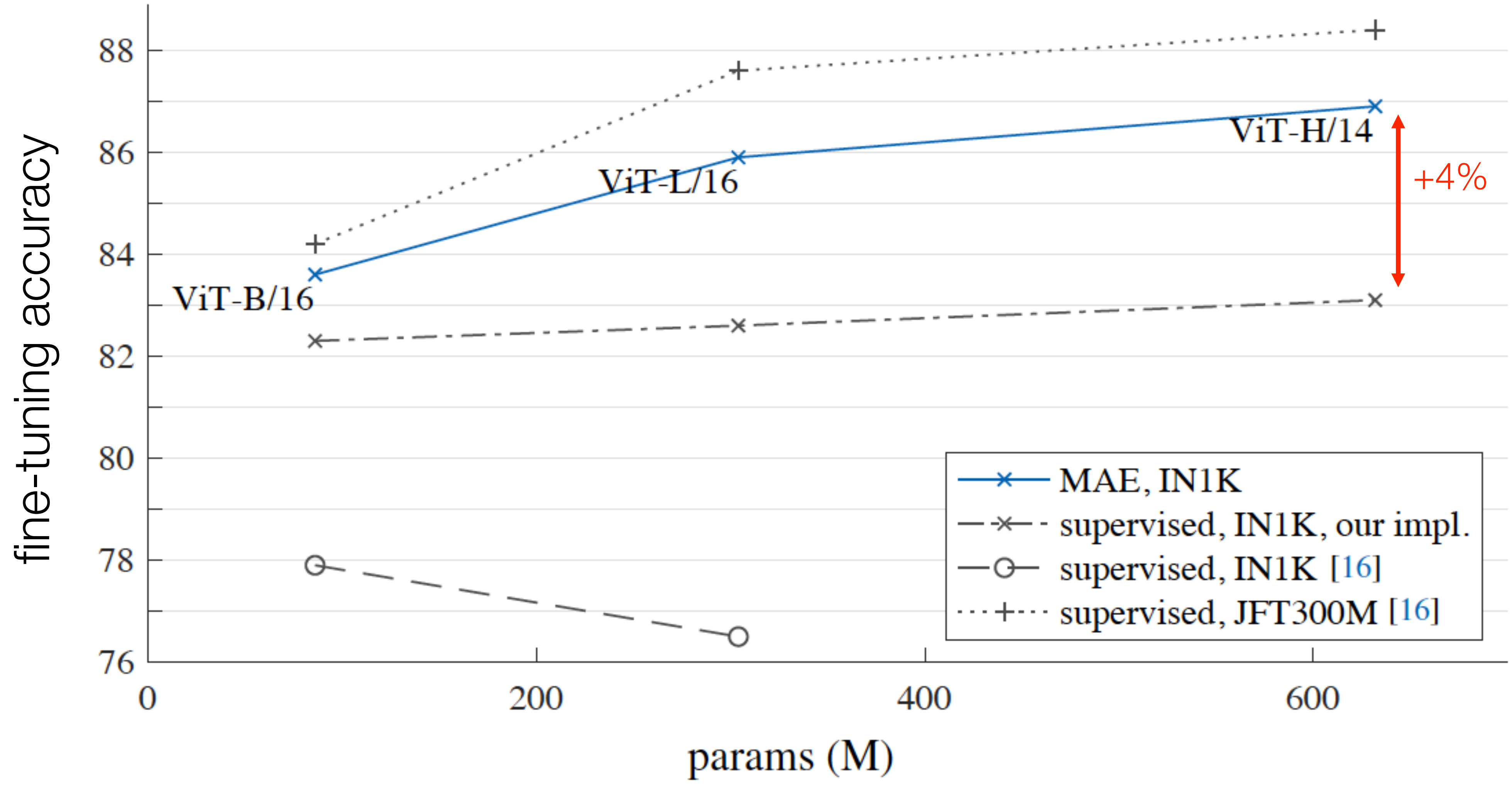
Scalability: Larger Models



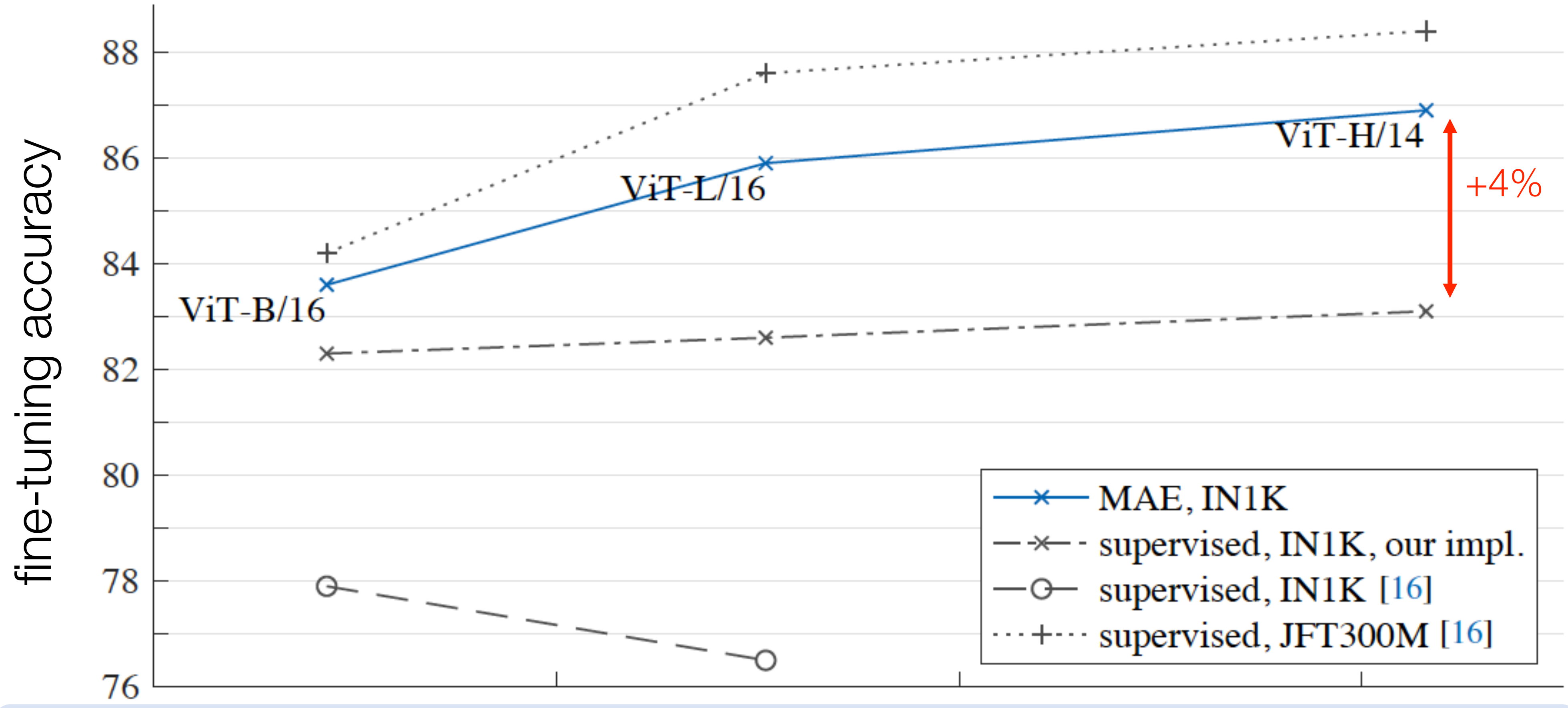
Scalability: Larger Models



Scalability: Larger Models



Scalability: Larger Models



new SOTA on ImageNet-1K (no extra data): **87.8%**

Scalability: Larger Models

dataset	ViT-B	ViT-L	ViT-H	ViT-H_{448}	prev best
iNat 2017	70.5	75.7	79.3	**83.4**	75.4 [50]
iNat 2018	75.4	80.1	83.0	**86.8**	81.2 [49]
iNat 2019	80.5	83.4	85.7	**88.3**	84.1 [49]
Places205	63.9	65.8	65.9	**66.8**	66.0 [19]†
Places365	57.9	59.4	59.8	**60.3**	58.0 [36]‡

new SOTA on 5 large-scale classification datasets

dataset	ViT-B	ViT-L	ViT-H	ViT-H_{448}	prev best
IN-Corruption ↓ [27]	51.7	41.8	**33.8**	36.8	42.5 [32]
IN-Adversarial [28]	35.9	57.1	68.2	**76.7**	35.8 [41]
IN-Rendition [26]	48.3	59.9	64.4	**66.5**	48.7 [41]
IN-Sketch [60]	34.5	45.3	49.6	**50.9**	36.0 [41]

new SOTA on 4 ImageNet robust evaluations

Scalability: Larger Models

method	pre-train data	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3
MoCo v3	IN1K	47.9	49.3
BEiT	IN1K+DALLE	49.8	**53.3**
MAE	IN1K	**50.3**	**53.3**

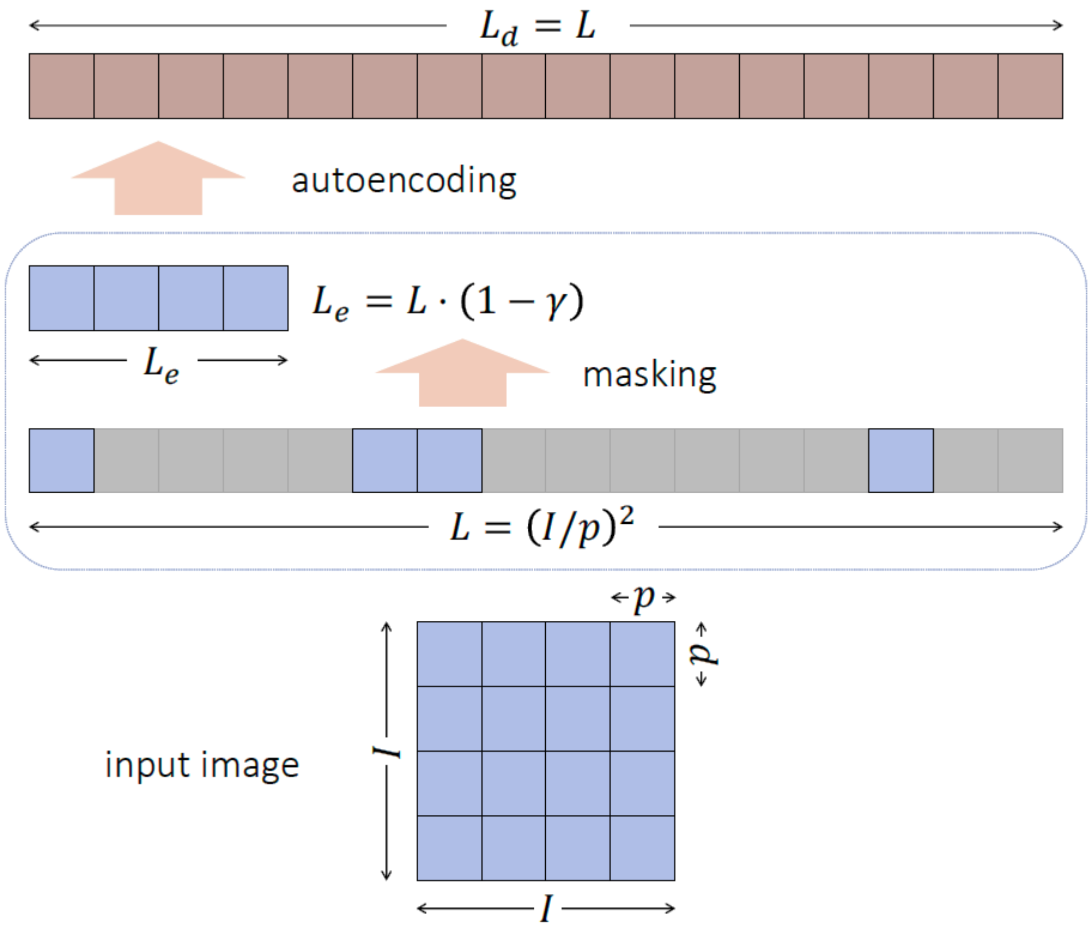
COCO detection: **+4.0%**

method	pre-train data	ViT-B	ViT-L
supervised	IN1K w/ labels	47.4	49.9
MoCo v3	IN1K	47.3	49.1
BEiT	IN1K+DALLE	47.1	53.3
MAE	IN1K	**48.1**	**53.6**

ADE20K segmentation: **+3.7%**

[Hu et al, arXiv 2202.04639]

Scalability: Sequence Length



- Input: $p \times p$ patches from $I \times I$ images as tokens
- Length of token sequence $L = (I/p)^2$
- Analysis: change sequence length for MAE, but fix length for downstream tasks

[Hu et al, arXiv 2202.04639]

Analysis of L, I and p, $L = (I/p)^2$

p	L	AP^b	AP^m	mIoU
64	49	44.0	39.8	35.0
32	196	49.5	44.2	48.0
16	784	**51.7**	**45.9**	**50.8**

image size $I = 448$

I	L	AP^b	AP^m	mIoU
112	49	47.3	42.1	42.2
224	196	50.4	45.1	49.4
448	784	**51.7**	**45.9**	**50.8**

patch size $p = 16$

I	p	AP^b	AP^m	mIoU
224	8	**51.7**	**46.0**	50.5
448	16	**51.7**	45.9	**50.8**
672	24	**51.7**	45.8	50.4

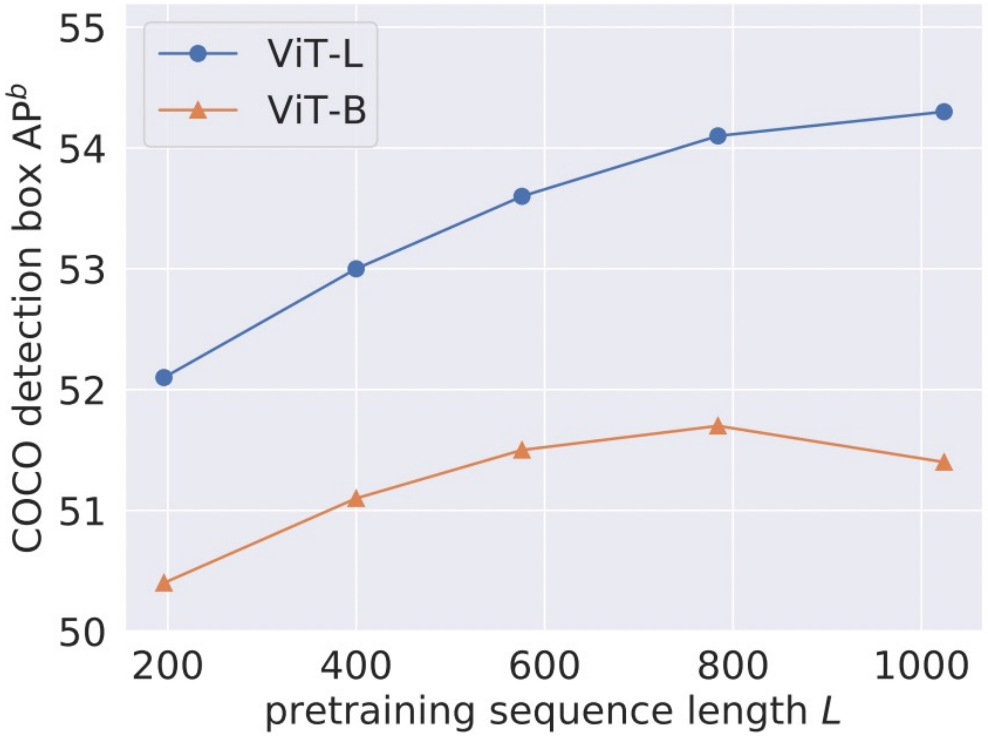
sequence length $L = 784$

COCO detection and ADE20K segmentation

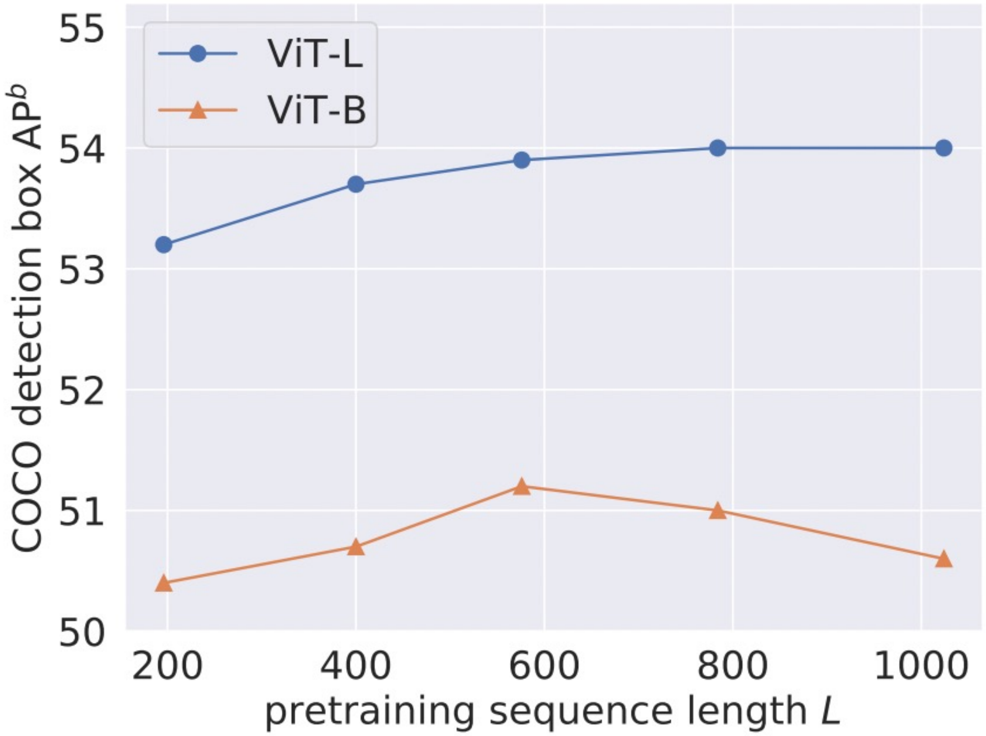
[Hu et al, arXiv 2202.04639]

Scalability: Sequence Length

- Sequence length helps more for larger models



COCO pre-training



ImageNet-1K pre-training

Take-aways

- Self-supervised learning allows representation learning at *scale*

- Masked auto-encoders as a step toward scalable vision learners

Take-aways

Large
Language
Models

- Self-supervised learning allows representation learning at *scale*

- Masked auto-encoders as a step toward scalable vision learners

- Still need to close the gap with large language models

MAE

Self-supervised learning from masked video and audio

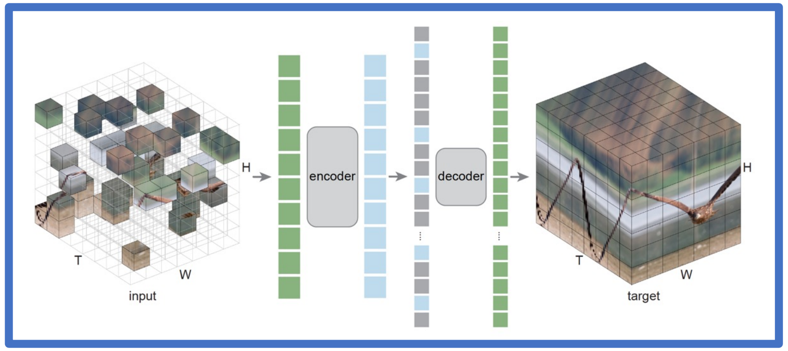
Christoph Feichtenhofer

Meta AI, FAIR

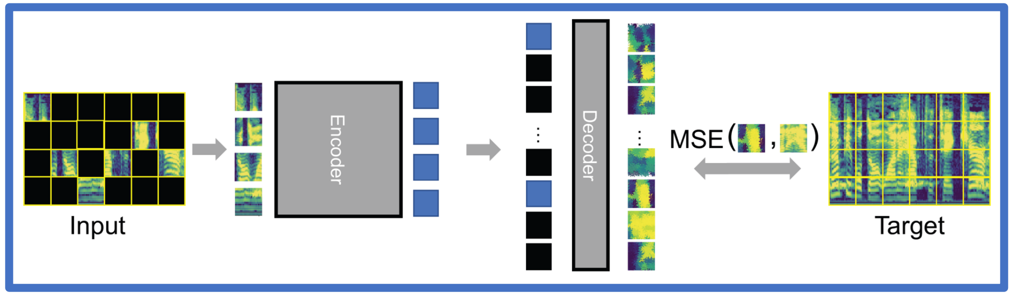
Outline: Advances in representation learning from video

- 4 topics on masked self-supervised learning from video (visual) and audio information

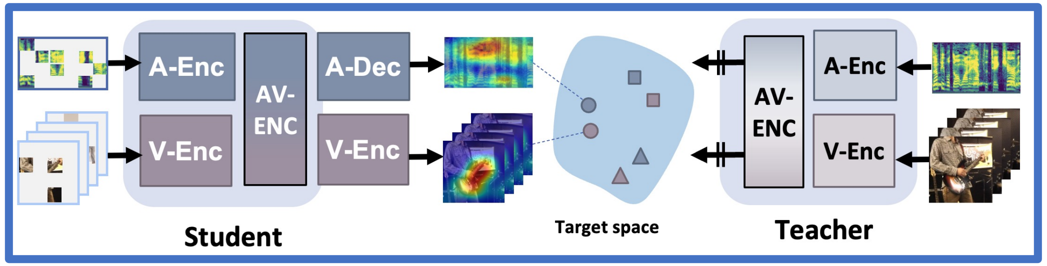
1. Video Masked Autoencoders



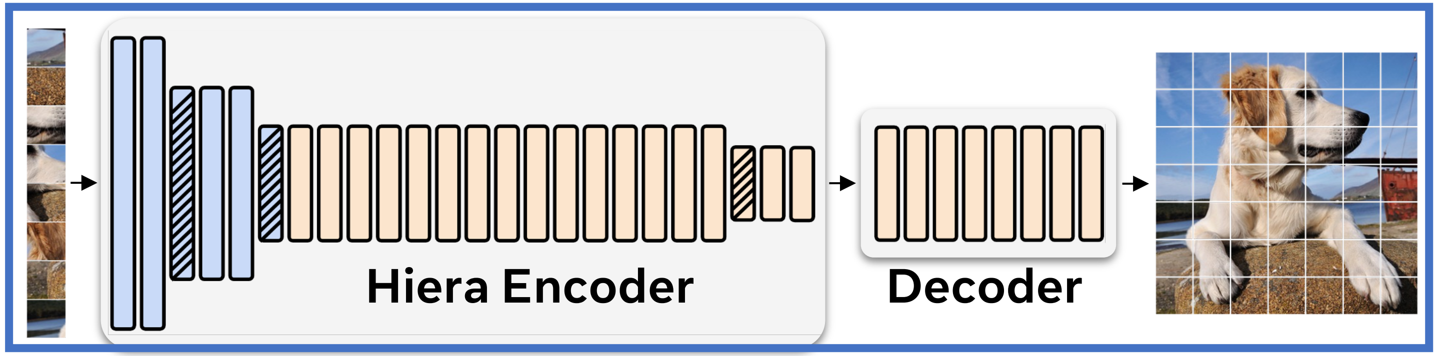
2. Audio Masked Autoencoders



3. Masked Audio-Video Learners



4. Hiera, a fast hierarchical transformer



Masked Autoencoders As Spatiotemporal Learners

Christoph Feichtenhofer*, Haoqi Fan*, Yanghao Li, Kaiming He

Meta AI, FAIR

github.com/facebookresearch/mae_st
github.com/facebookresearch/SlowFast

Masked Language Modeling

men basketball

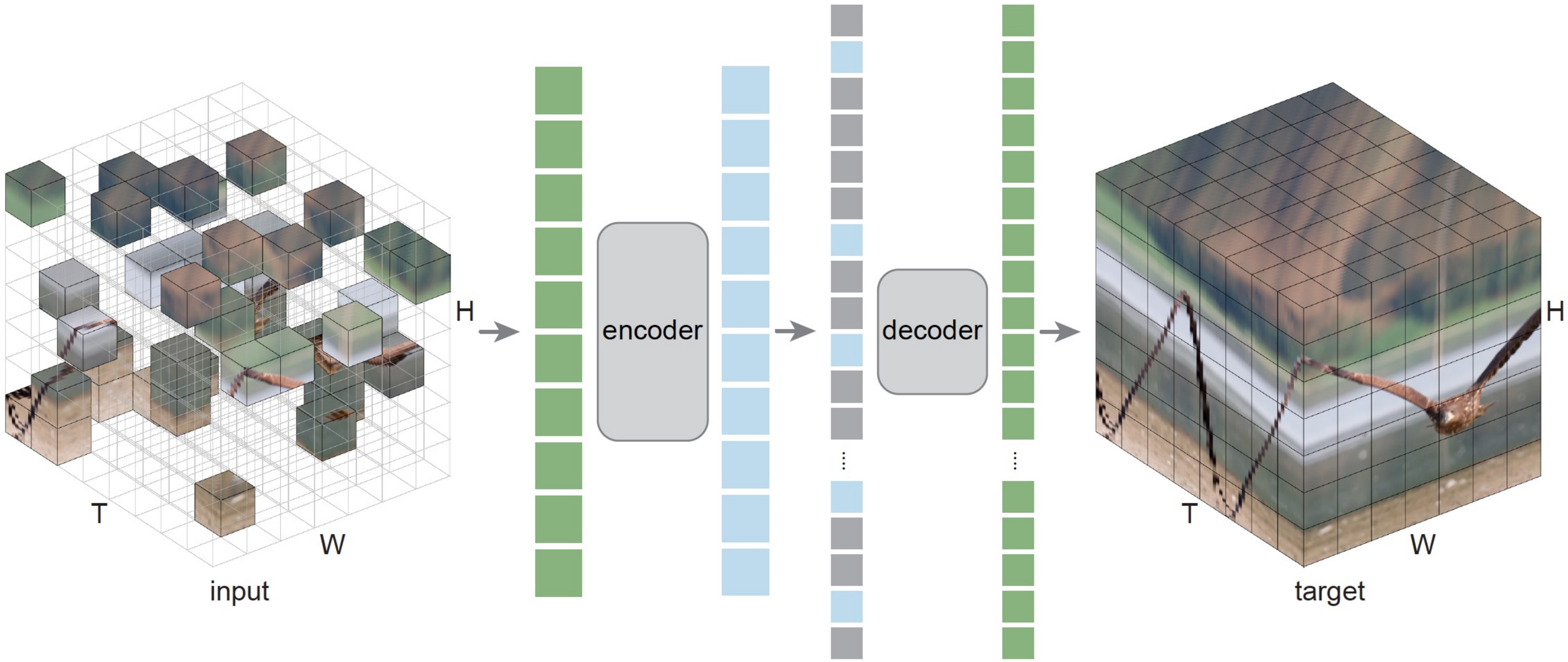
“Two playing a game of on an outside court.”

Devlin *et al.*, BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding

Masked Autoencoders (MAE) for visual learning

He *et al.*, Masked Autoencoders Are Scalable Vision Learners

Masked Autoencoders as spatiotemporal learners

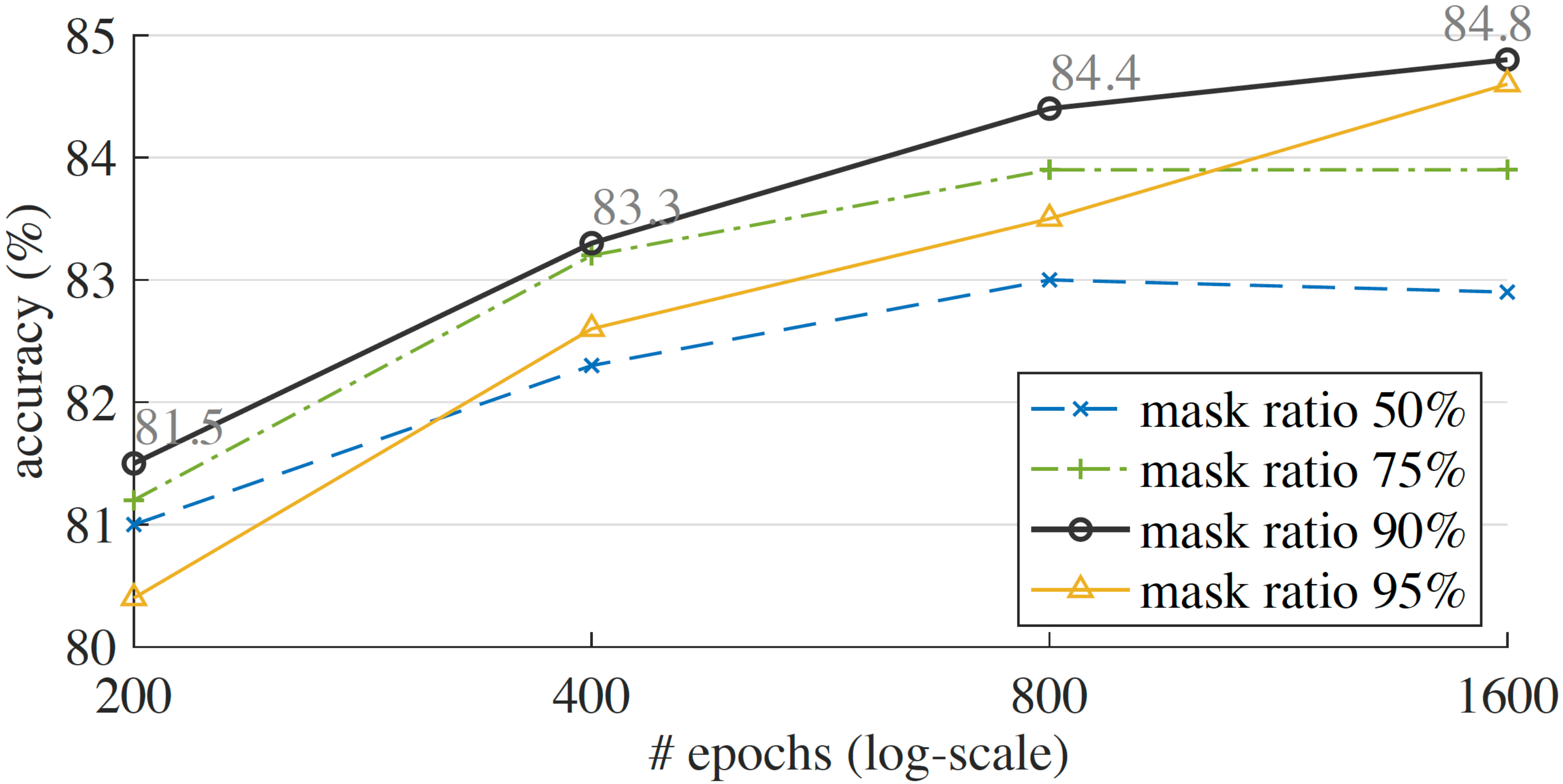


- Masking of random patches in spacetime
- Encoder operates on the set of visible patches
- A small decoder on encoded patches and mask tokens reconstruct input
- Except for patch and positional embeddings, no inductive bias

Masking ratio can be extremely high

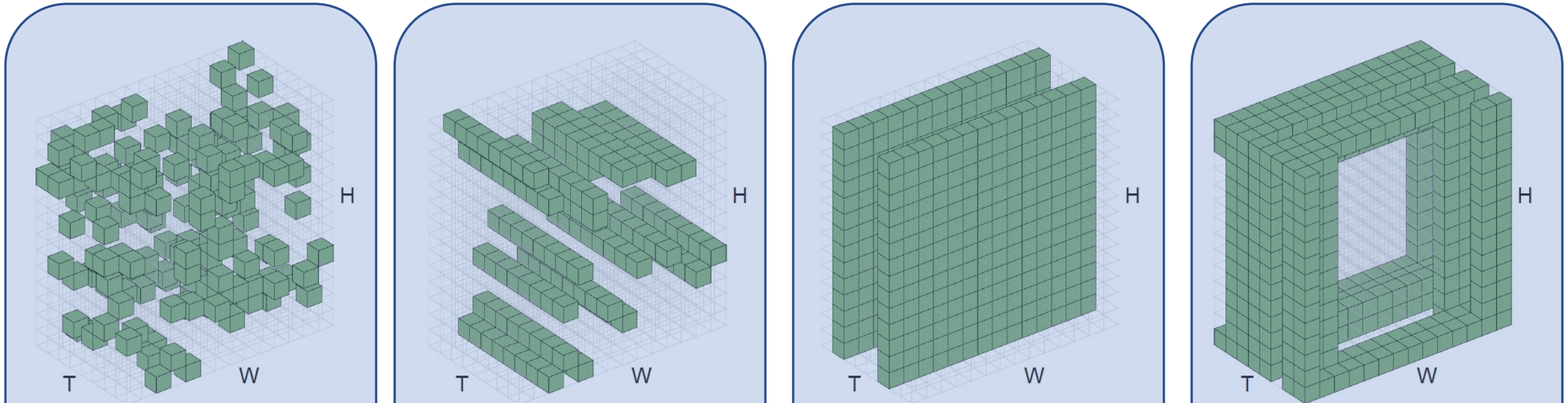
- **Task:**
  Kinetics-400 (K400)
  video classification
- Metric: accuracy (acc.)

- Model: ViT-L
- Pre-train: 200-1600 epochs
- Fine-tune: 100 epochs
- Training from scratch: 71.4%



- For image classification, 75% is the optimal value, but for video 90% is considerably better

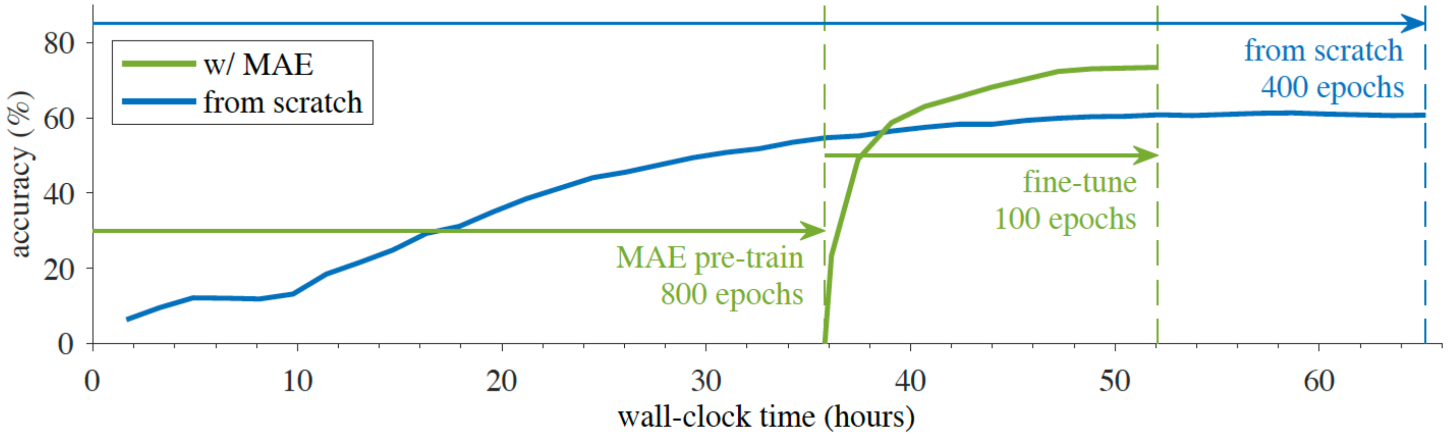
Masking can be agnostic in spacetime



(a) agnostic, 90% (b) space-only, 90% (c) time-only, 75% (d) block, 75%

case	ratio	acc.
agnostic	90	**84.4**
space-only	90	83.5
time-only	75	79.1
block	75	83.2

MAE is faster than pure supervised training



	scratch	MAE
1-view	60.7	**73.4** (+12.7)
multi-view	71.4	**84.4** (+13.0)

Figure 5: MAE pre-training plus fine-tuning is *much more accurate* and *faster* than training from scratch. Here the x-axis is the wall-clock training time (128 A100 GPUs), and the y-axis is the 1-view accuracy on Kinetics-400 validation. The table shows the final accuracy. The model is ViT-L.

Influence of data scale and curation

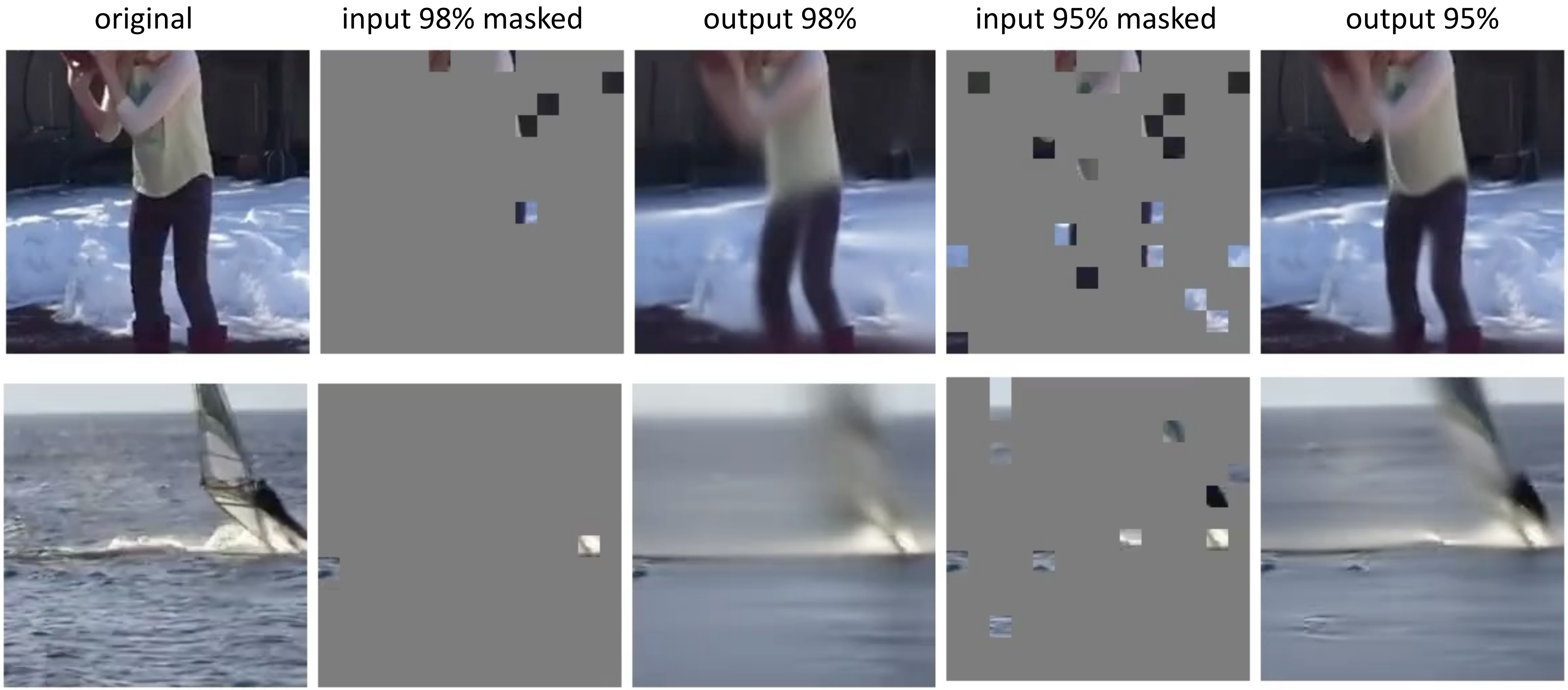
pre-train set	# pre-train data	pre-train method	K400	AVA	SSv2
-	-	none (from scratch)	71.4	-	-
K400	240k	supervised	-	21.6	55.7
K400	240k	MAE	84.8	31.1	72.1
K600	387k	MAE	**84.9**	32.5	73.0
K700	537k	MAE	n/a†	33.1	**73.6**
IG-uncurated	1M	MAE	84.4	**34.2**	**73.6**

Table 3: **Influence of pre-training data**, evaluated on K400, AVA, and SSv2 as the downstream tasks.

MAE visualizations

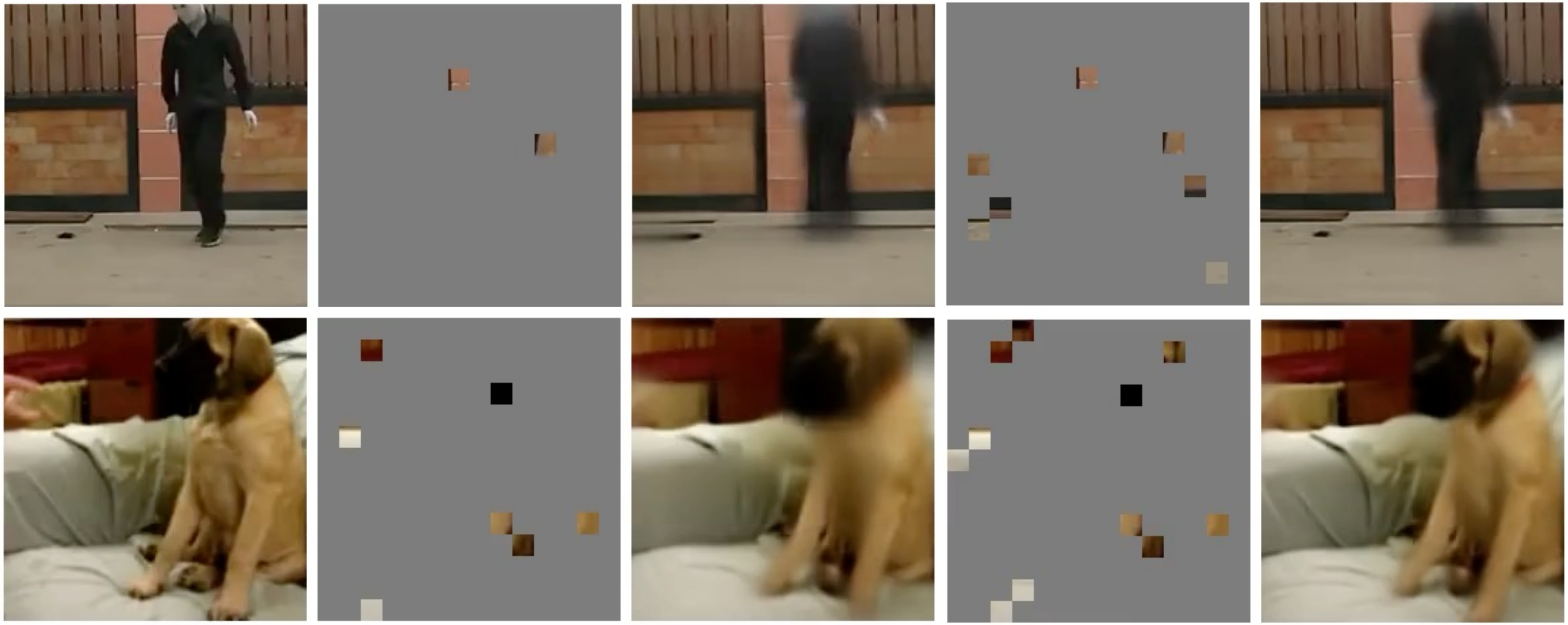
original	input 98% masked	output 98%	input 95% masked	output 95%

MAE visualizations



MAE visualizations

original | input 98% masked | output 98% | input 95% masked | output 95%



MAE visualizations

original | input 98% masked | output 98% | input 95% masked | output 95%

MAE visualizations

original | input 98% masked | output 98% | input 95% masked | output 95%

Masked Autoencoders that Listen

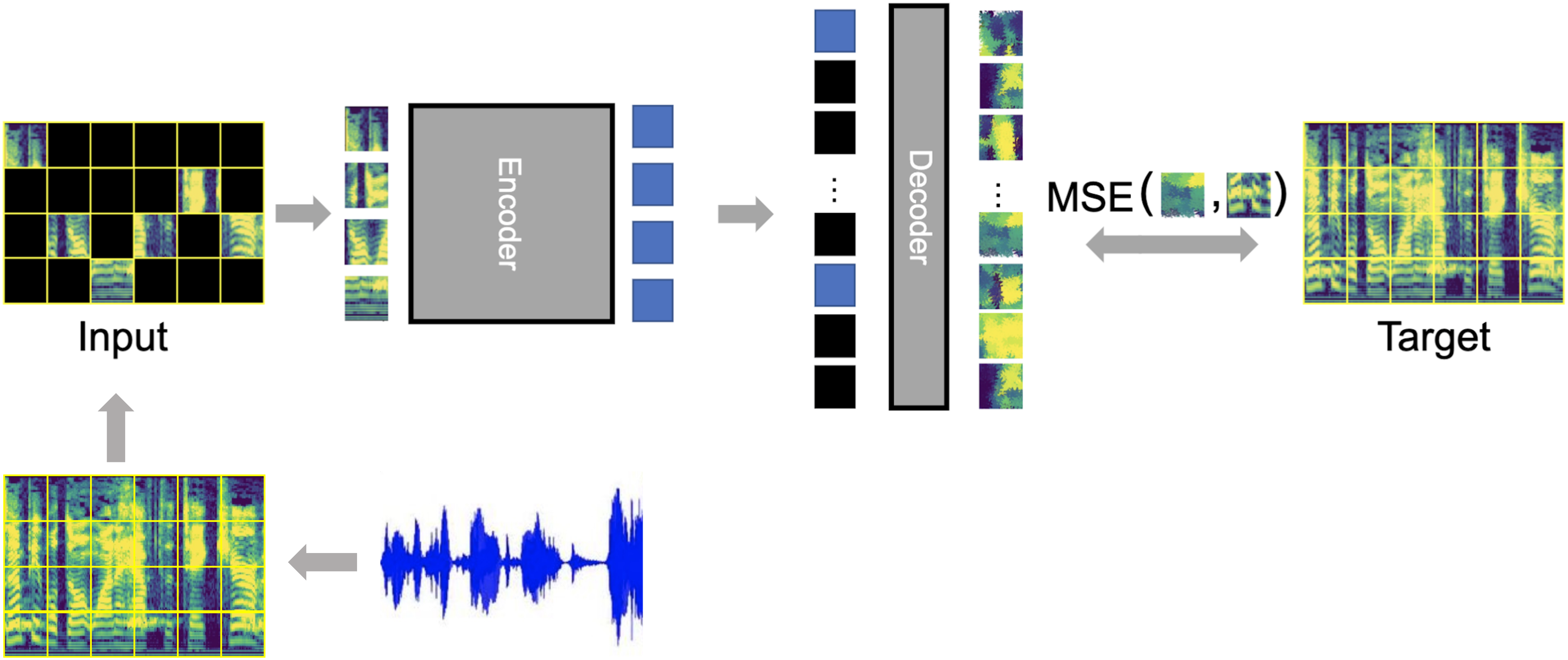
Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski
Michael Auli, Wojciech Galuba, Florian Metze, Christoph Feichtenhofer

Meta AI, FAIR

In NeurIPS 2022

github.com/facebookresearch/AudioMAE

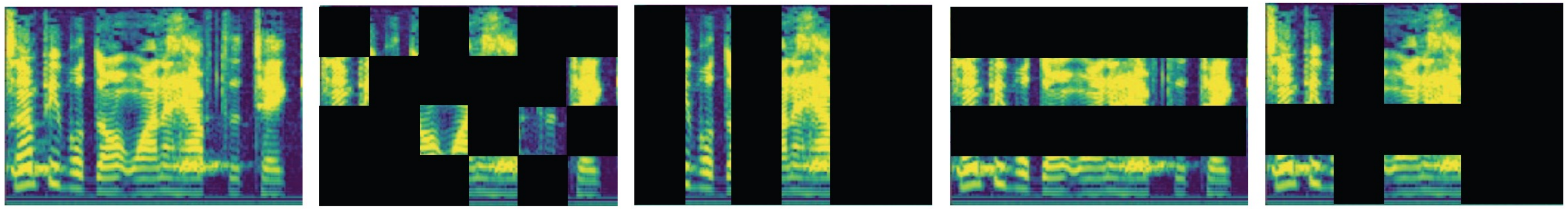
Audio-MAE



Experiments

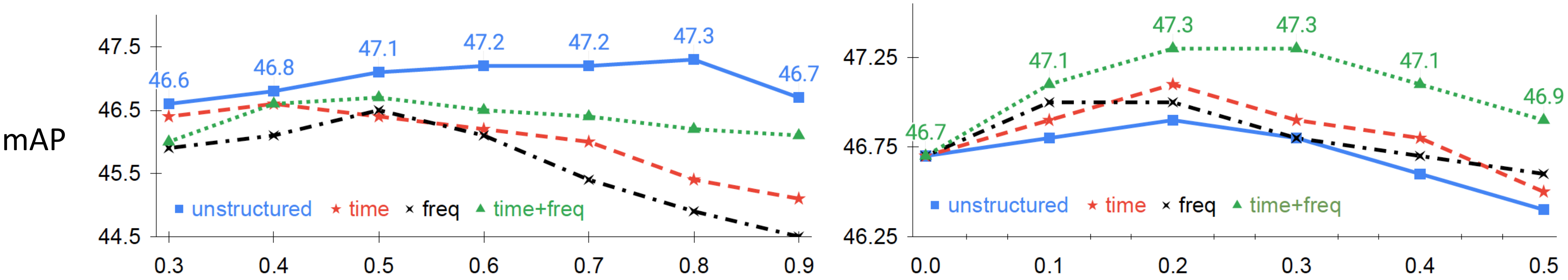
- Pre-training (PT)
  - Audioset-2M
    - 2 million 10-sec audio recordings in unbalanced 527 classes
      - Labels are not used (self-supervised pre-training)
    - For each 10-sec audio recording
      - 128 Mel-fbanks / 1024 time windows (stride 10 ms)
      - Shape: 1024x128x1

- Fine-tuning
  - Audioset-20K (balanced)
  - Audioset-2M (unbalanced)
  - ESC-50
  - Speech commands v1
  - Speech commands v2
  - SID (Voxceleb)



(a) Original (b) Unstructured (c) Time (d) Frequency (e) Time+frequency

Figure 2: Masking strategies for Audio-MAE.



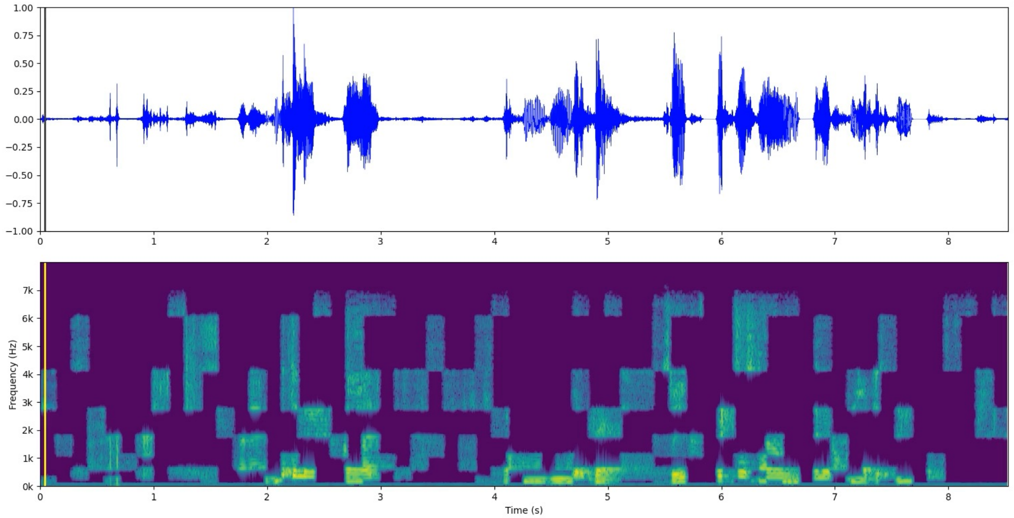
(a) Pre-training masking **ratio** (b) Fine-tuning masking **ratio**

Comparison to state-of-the-art

Model	Backbone	PT-Data	AS-20K	AS-2M	ESC-50	SPC-2	SPC-1	SID
No pre-training								
ERANN [57]	CNN	-	-	45.0	89.2	-	-	-
PANN [58]	CNN	-	27.8	43.1	83.3	61.8	-	-
In-domain self-supervised pre-training								
wav2vec 2.0 [33]	Transformer	LS	-	-	-	-	96.2^{*}	75.2^{*}
HuBERT [35]	Transformer	LS	-	-	-	-	96.3^{*}	81.4^{*}
Conformer [37]	Conformer	AS	-	41.1	88.0	-	-	-
SS-AST [18]	ViT-B	AS+LS	31.0	-	88.8	98.0	96.0	64.3
Concurrent MAE-based works								
MaskSpec [43]	ViT-B	AS	32.3	47.1	89.6	97.7	-	-
MAE-AST [38]	ViT-B	AS+LS	30.6	-	90.0	97.9	95.8	63.3
Audio-MAE (global)	ViT-B	AS	36.6$\pm$.11	46.8$\pm$.06	93.6$\pm$.11	**98.3**$\pm$.06	**97.6**$\pm$.06	94.1$\pm$.06
Audio-MAE (local)	ViT-B	AS	**37.1**$\pm$.06	**47.3**$\pm$.06	**94.1**$\pm$.10	**98.3**$\pm$.06	96.9$\pm$.00	**94.8**$\pm$.11
Out-of-domain supervised pre-training								
PSLA [30]	EffNet [59]	IN	31.9	44.4	-	96.3	-	-
AST [10]	DeiT-B	IN	34.7	45.9	88.7	98.1	95.5	41.1
MBT [11]	ViT-B	IN-21K	31.3	44.3	-	-	-	-
HTS-AT [29]	Swin-B	IN	-	47.1	$97.0^{\dagger}$	98.0	-	-
PaSST [28]	DeiT-B	IN	-	47.1	$96.8^{\dagger}$	-	-	-

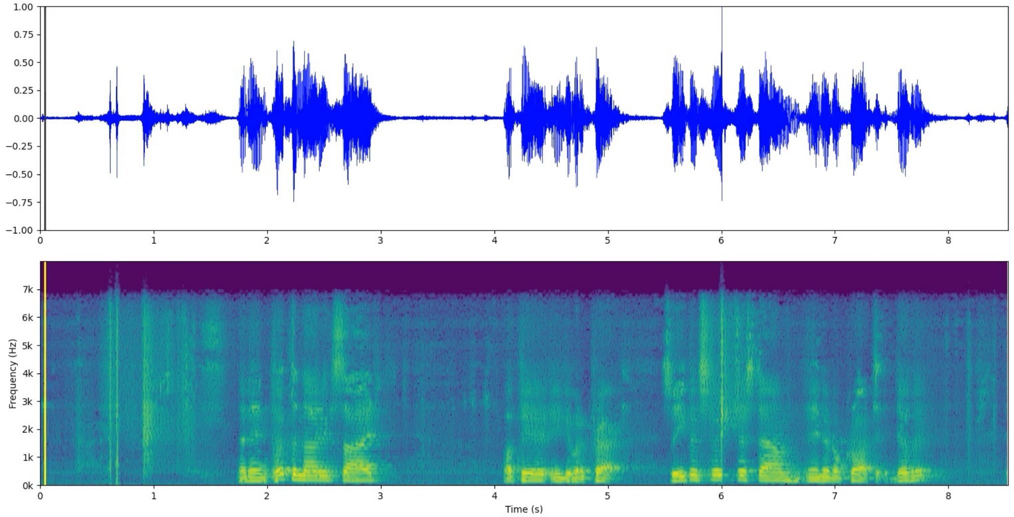
Audio-MAE speech sample

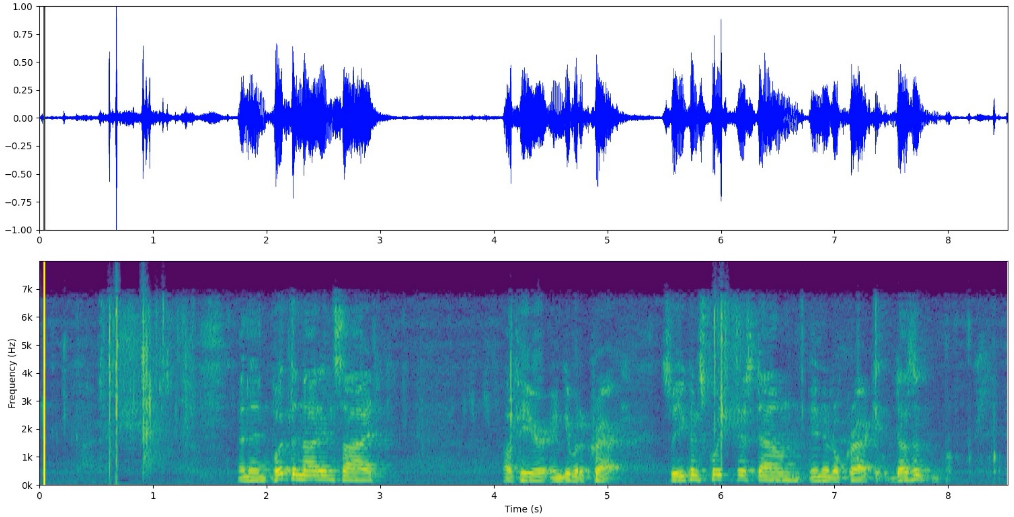
masked 80%



original

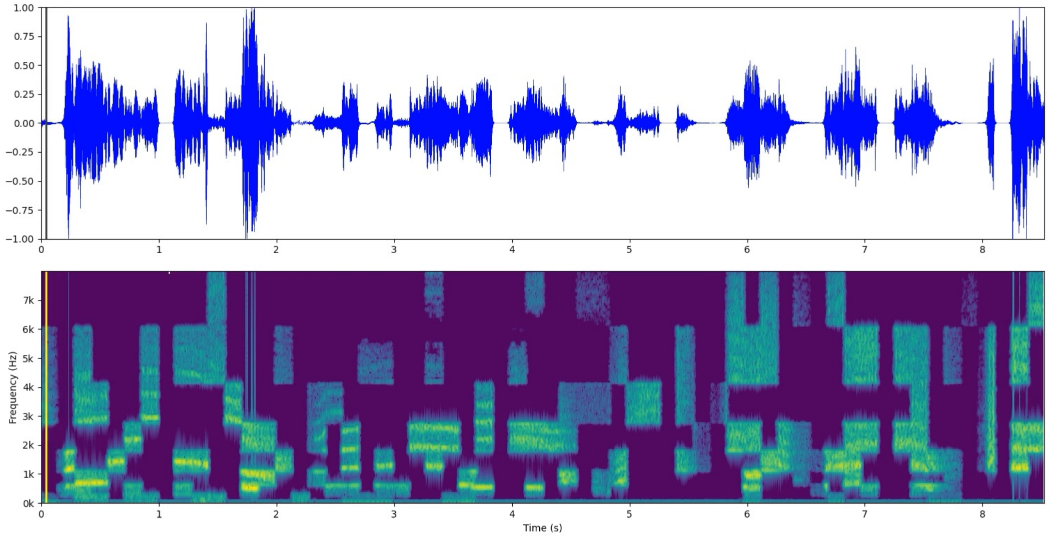
reconstruction output





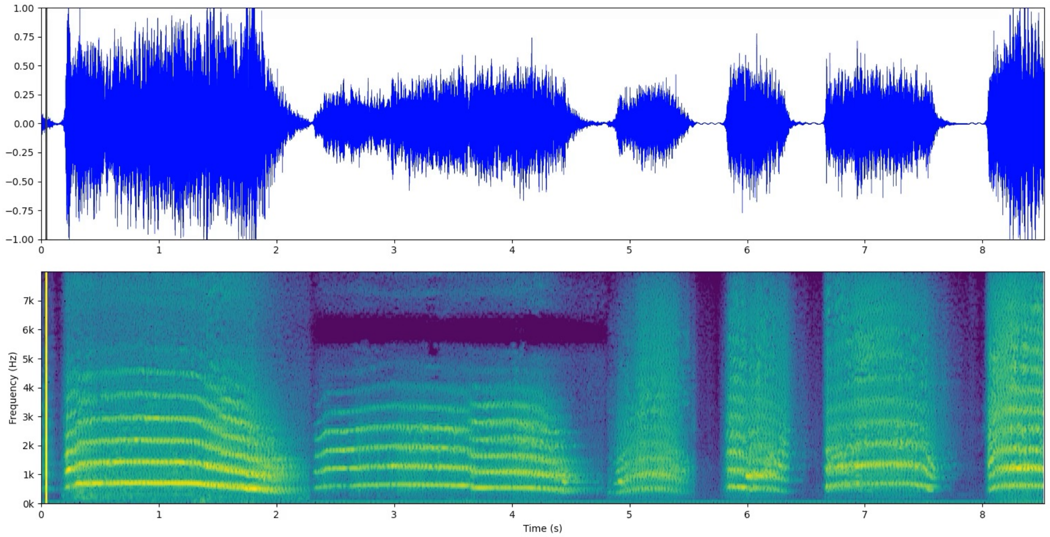
Audio-MAE misc sound sample

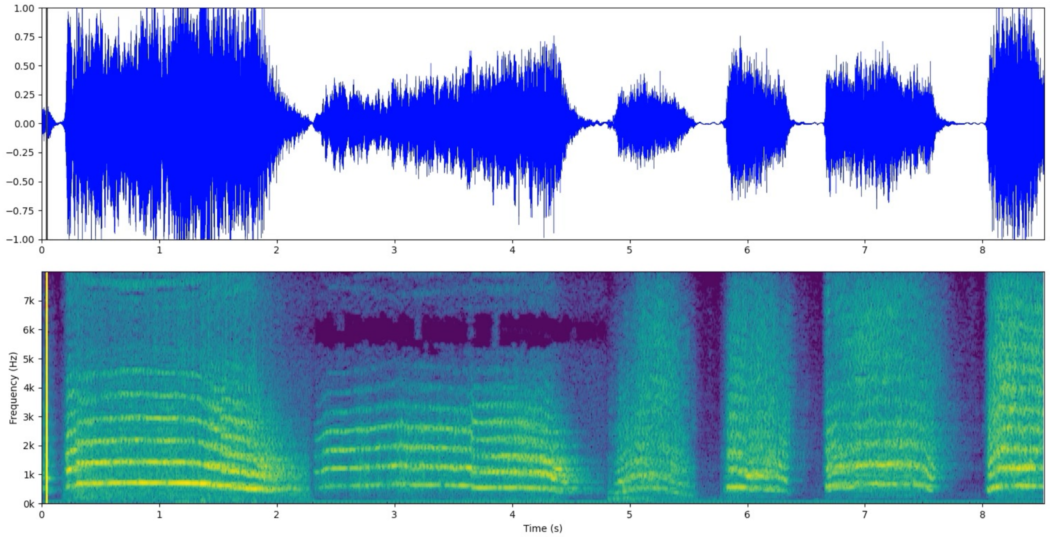
masked 80%



original

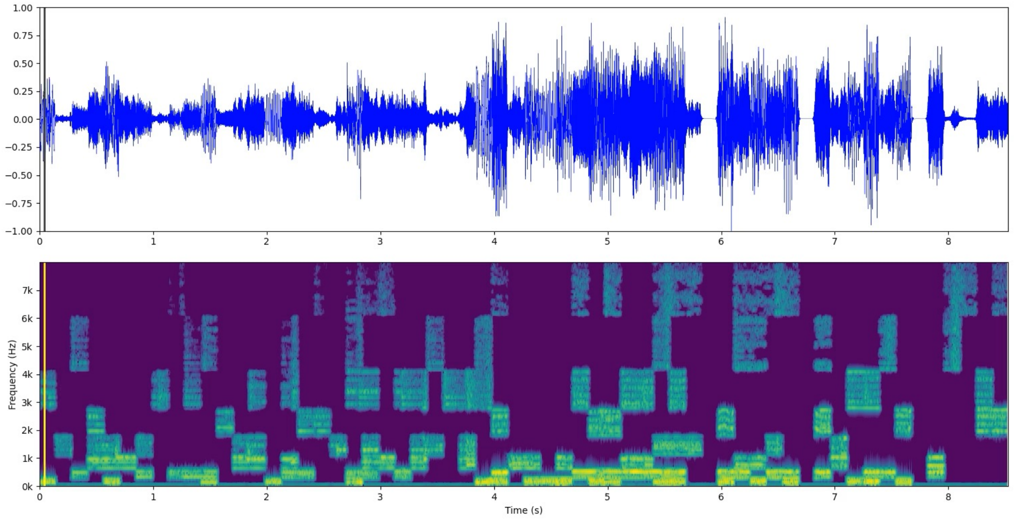
reconstructio
output





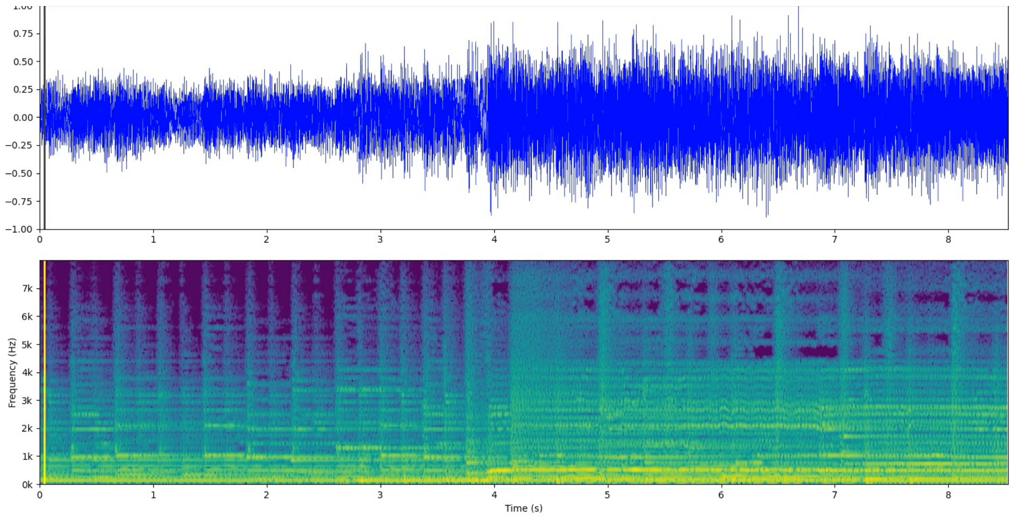
Audio-MAE music sample

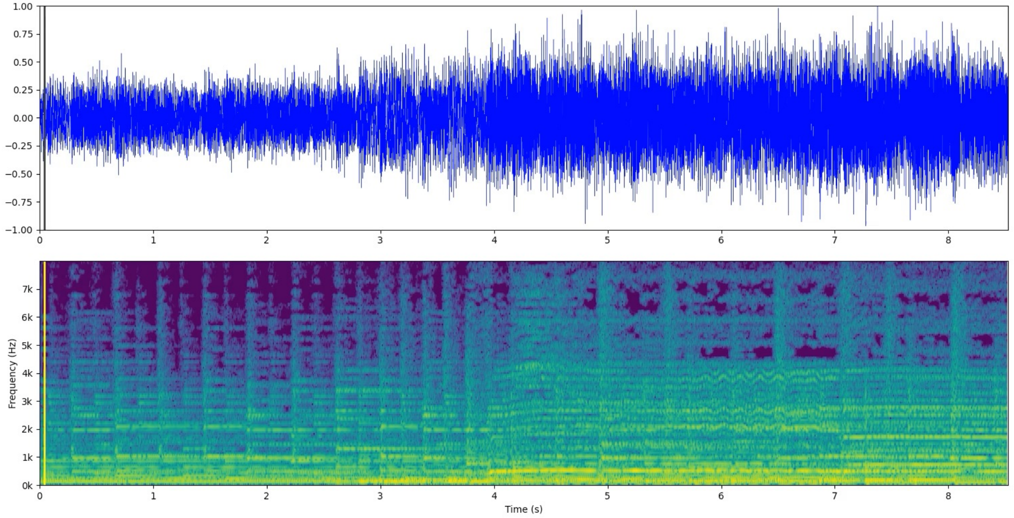
masked 80%



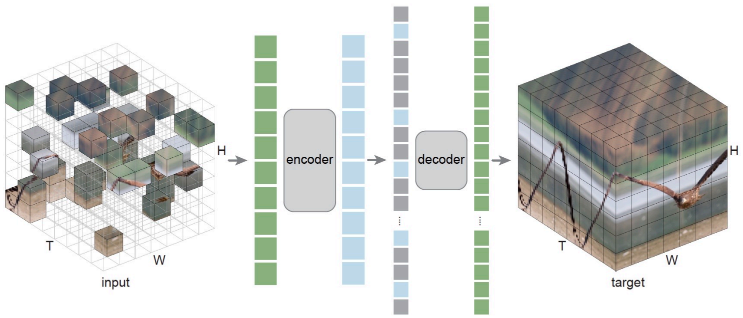
original

reconstruction output



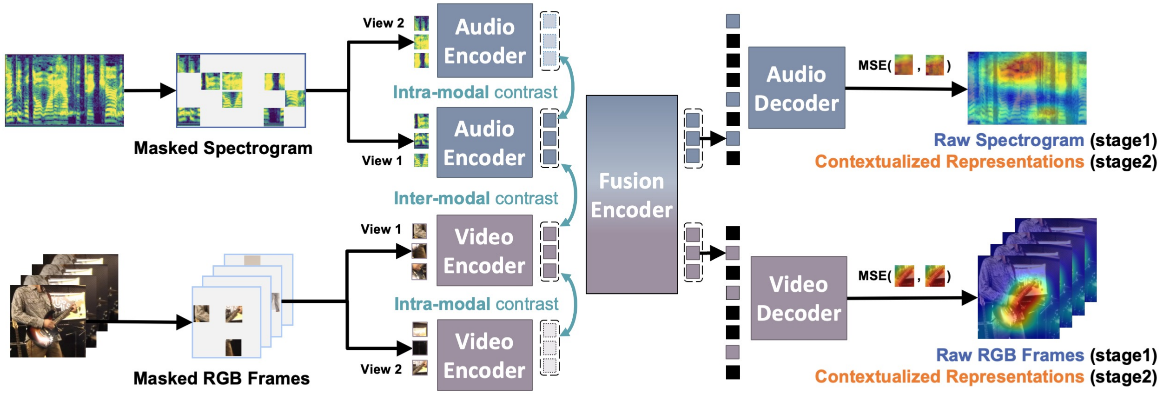


A unifying trend across Vision and Audio



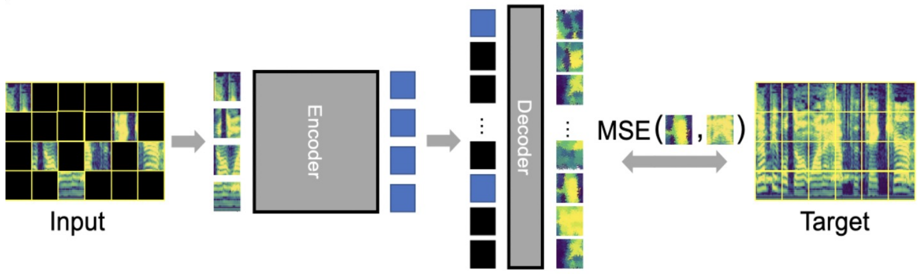
MAE-ST (Video-MAE)
NeurIPS 22

2022



MAViL: Masked Audio-Video Learners
arXiv 2023

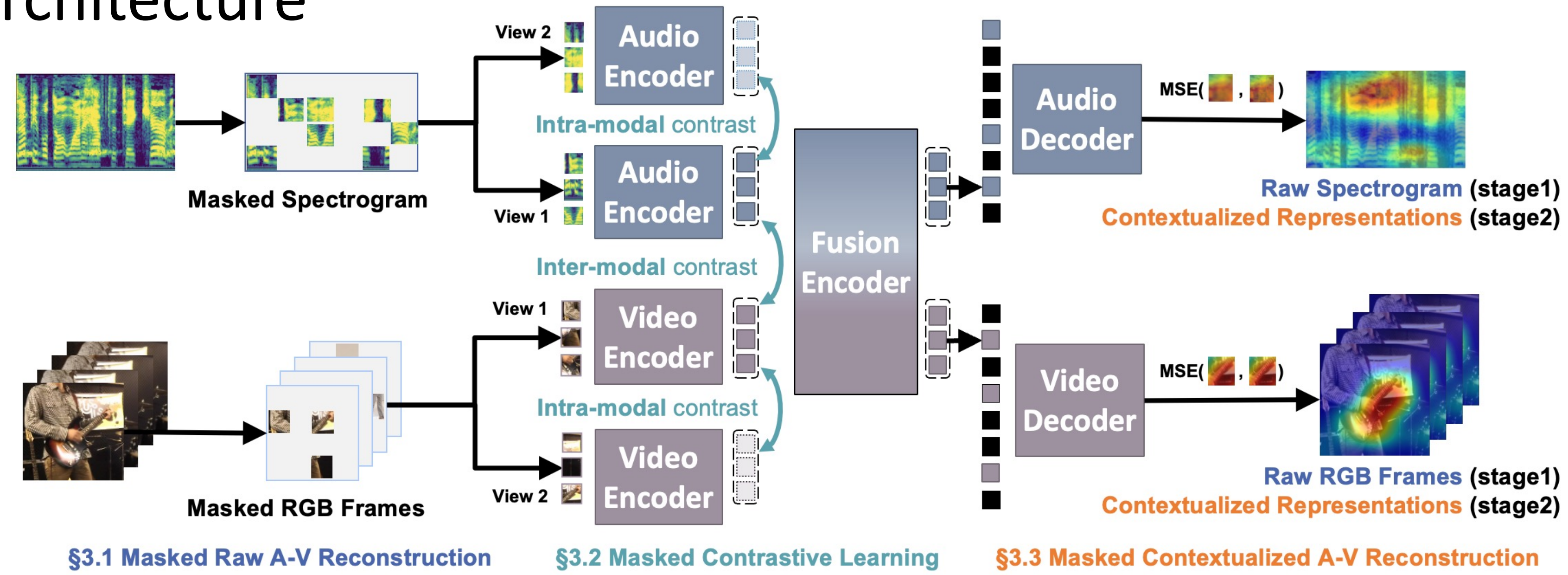
github.com/facebookresearch/MAViL



Audio-MAE
NeurIPS 22

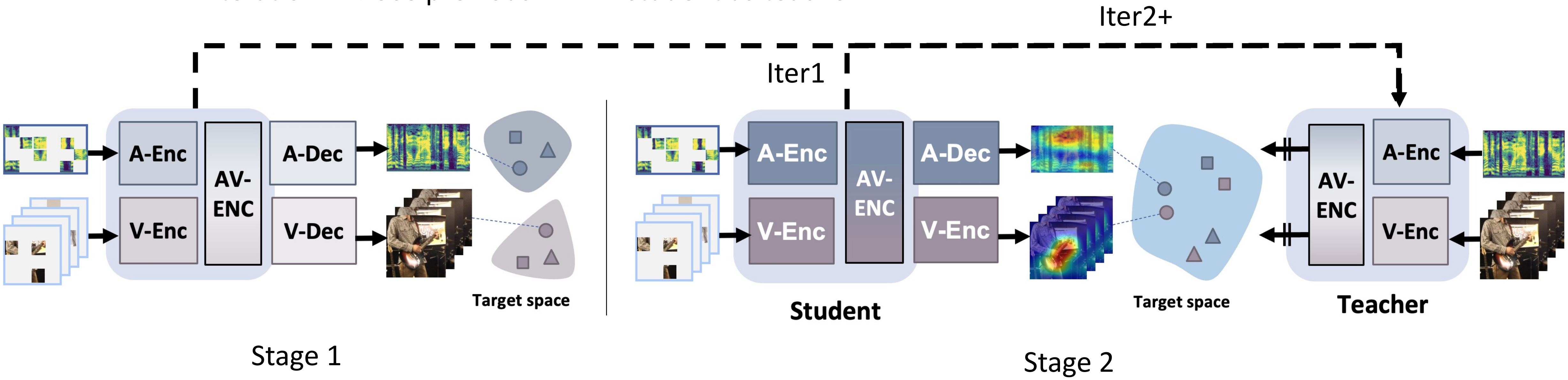
MAViL: Masked Audio-Video Learners

- Reconstructing aligned & contextualized representations
  - Inter-modal and intra-modal masked contrastive learning for promoting alignment between semantically correlated audio and/or video.
  - Train a student under masked view to predict contextualized representations in the aligned latent space generated by a teacher with full-view.
- Model Architecture



MAViL: Masked Audio-Video Learners

- Two-stage training:
  - Stage 1: Contrastive objectives and raw A-V reconstruction
  - Stage 2: Contrastive objectives and contextualized reconstruction from a Teacher
    - Iteration 1: Use Stage1 MAViL as teacher
    - Iteration 2+: Use previous MAViL student as teacher



Experiments

- Pre-training (PT)
  - Audioset-2M
    - 2 million 10-sec videos
      - Labels are not used (self-supervised pre-training)
    - For each 10-sec audio track
      - 128 Mel-fbanks/ 1024 time windows (stride 10 ms)
      - Shape: 1x1024x128
    - For each 10-sec video track
      - Sample 4-sec with 8 frames
      - Shape: 8x3x224x224
  - MAViL model
    - ViT-B backbone for Audio/Video
    - 80% Masking Ratio

- Fine-tuning
  - A-V Classification
    - Audioset-20K (balanced)
    - Audioset-2M (unbalanced)
    - VGGSound
  - Audio-only Classification
    - Speech commands v1
    - ESC-50
  - Audio-Video Retrieval
    - YouCook
    - MSR-VTT

Ablation Studies on Audioset-2M

Method	Audio	Video
A-MAE/V-MAE (baseline)	36.4	17.4
MAViL stage-1		
+ Joint AV-MAE	$36.8_{(+0.4)}$	$17.7_{(+0.3)}$
+ Inter contrast	38.4	21.0
+ Intra and Inter contrast	$39.0_{(+2.2)}$	$22.2_{(+4.5)}$
MAViL stage-2		
+ Student-teacher learning	$41.8_{(+2.8)}$	$24.8_{(+2.6)}$

Observation 1: Fusing multimodal info for MAE reconstruction improve 0.3-0.4 mAP

Observation 2: Both Inter-modal and Intra-modal contrastive learning helps!

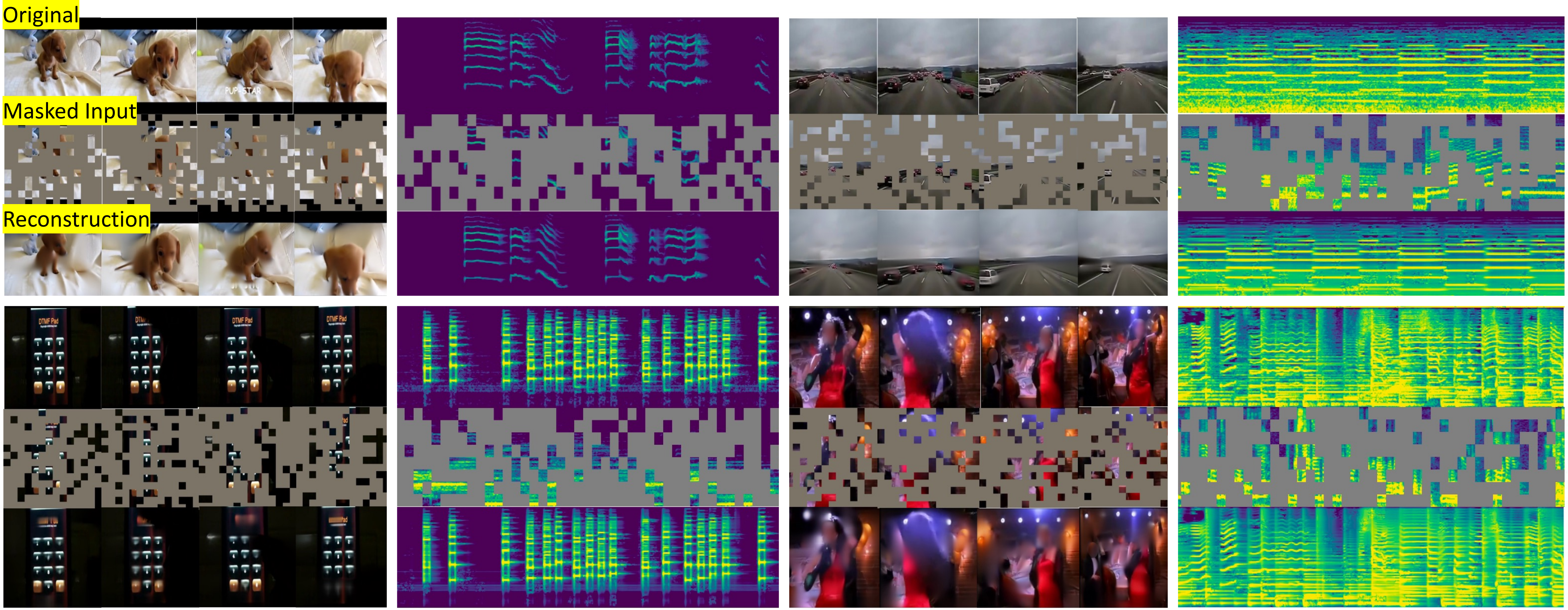
Observation 3: Reconstructing aligned and contextualized representations provides additional 2.6-2.8 mAP gains!

A-V Classification

Method	PT	AS-20K (mAP↑)			AS-2M (mAP↑)			VGGSound (Acc.↑)		
		A	V	A+V	A	V	A+V	A	V	A+V
Audio-only Models										
Aud-SlowFast [68]	-	-	-	-	-	-	-	50.1	-	-
VGGSound [58]	-	-	-	-	-	-	-	48.8	-	-
PANNs [69]	-	27.8	-	-	43.9	-	-	-	-	-
AST [64]	IN-SL	34.7	-	-	45.9	-	-	-	-	-
HTS-AT [70]	IN-SL	-	-	-	47.1	-	-	-	-	-
PaSST [71]	IN-SL	-	-	-	47.1	-	-	-	-	-
Data2vec [51]	AS-SSL	34.5	-	-	-	-	-	-	-	-
SS-AST [72]	AS-SSL	31.0	-	-	-	-	-	-	-	-
MAE-AST [73]	AS-SSL	30.6	-	-	-	-	-	-	-	-
Aud-MAE [4]	AS-SSL	37.0	-	-	47.3	-	-	-	-	-
Audio-Video Models										
G-Blend [74]	-	29.1	22.1	37.8	32.4	18.8	41.8	-	-	-
Perceiver [75]	-	-	-	-	38.4	25.8	44.2	-	-	-
Attn AV [76]	IN-SL	-	-	-	38.4	25.7	44.2	-	-	-
CAV-MAE [41]	IN-SSL, AS-SSL	37.7	19.8	42.0	46.6	26.2	51.2	59.5	47.0	65.5
MBT* [27]	IN21K-SL	31.3	27.7	43.9	41.5	31.3	49.6	52.3	**51.2**	64.1
MAViL	AS-SSL	41.6	23.7	44.6	**48.7**	28.3	51.9	60.6	50.0	66.5
MAViL	IN-SSL, AS-SSL	**41.8**	**24.8**	**44.9**	**48.7**	**30.3**	**53.3**	**60.8**	50.9	**67.1**

MAViL not only learns strong joint audio-video representations (A+V), but can also improve single modality encoders *without* using the other modality during fine-tuning (A, V).

Qualitative Results:



Hiera: A Hierarchical Vision Transformer without the Bells-and-Whistles

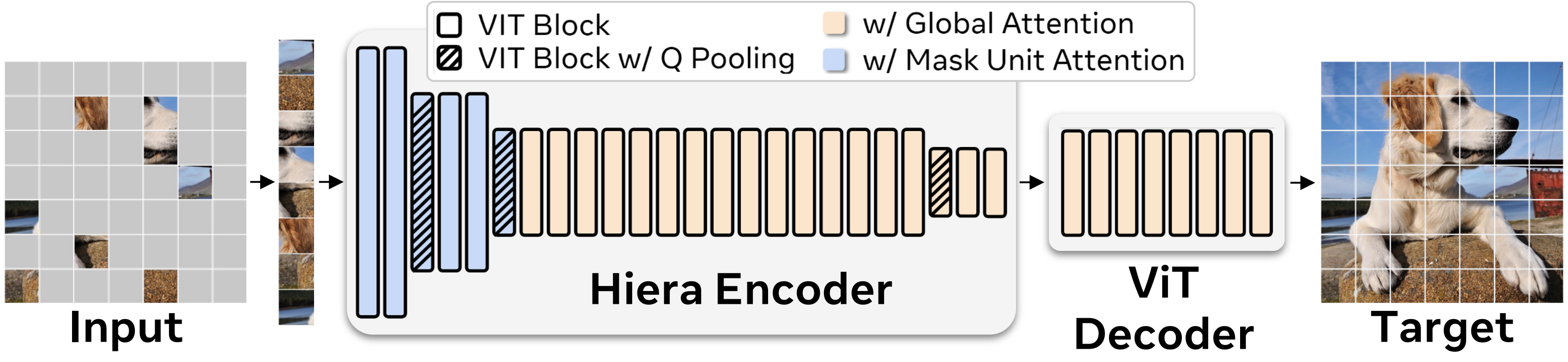
Chaitanya Ryali*, Yuan-Ting Hu*, Daniel Bolya*, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li*, Christoph Feichtenhofer*

Meta AI, FAIR

In ICML 2023
Oral A2 Computer Vision and Efficient ML **Tue 25 Jul 5:30 p.m.**
Poster**: Wed 26 Jul 2 p.m. — 3:30 p.m.**

github.com/facebookresearch/Hiera

Hiera: A Hierarchical Vision Transformer without the Bells-and-Whistles



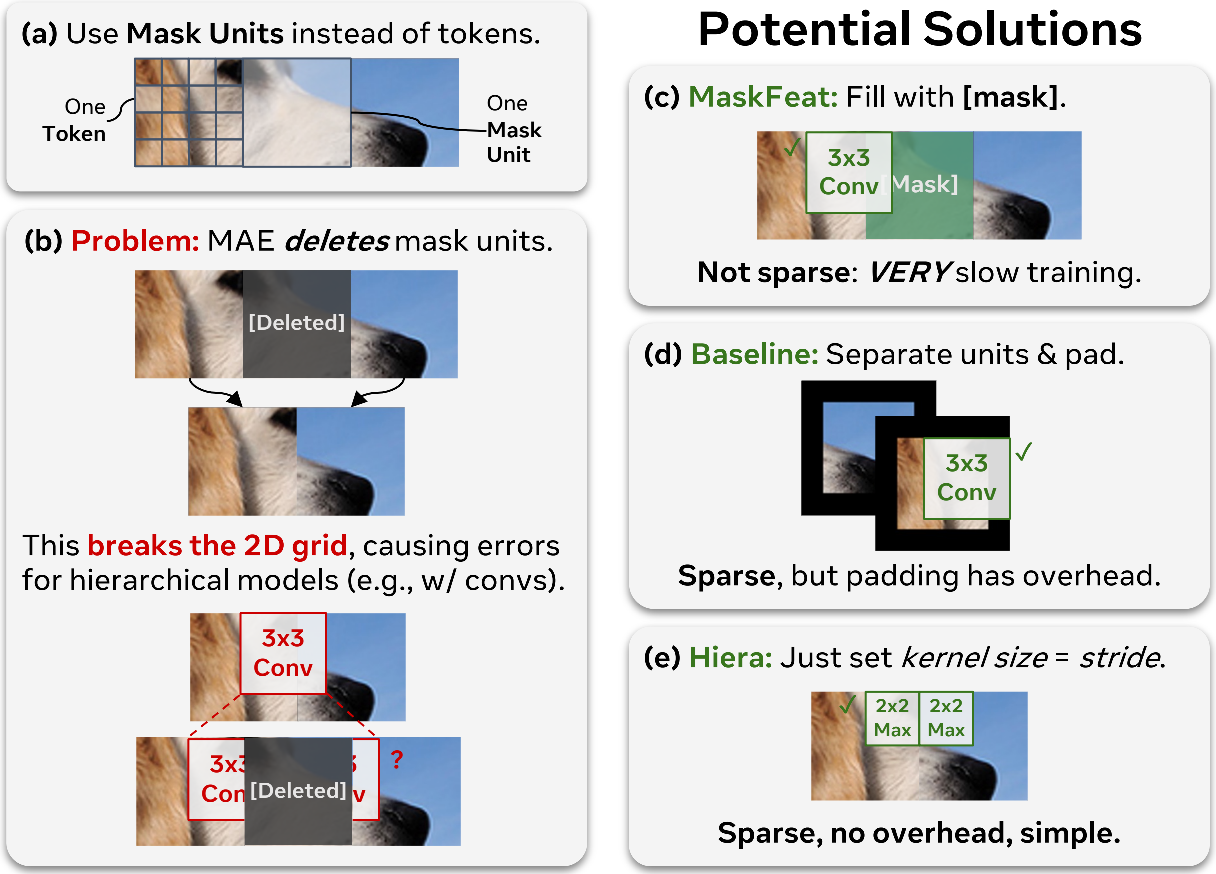
- A simple hierarchical vision transformer
- Created by removing the bells-and-whistles from an existing one (MViTv2)
- Works if we supply the model with spatial bias through MAE pre-training
- Decoder can be multi-scale, important for video accuracy

multi-scale	image	video
✗	85.0	83.8
✔	**85.6**	**85.5**

(a) **Multi-Scale Decoder.** Hiera being *hierarchical*, using multi-scale information for decoding brings significant gains.

Hiera: Mask Unit Attention

- MAE is incompatible with multi-scale models.
- MAE masks tokens, but tokens in multi-scale transformers start very small (e.g., 4 x 4 pixels).
- **(a)** We mask coarser “mask units” (32x32 pixels) instead of tokens directly.
- **(b)** MAE deletes what it masks (a problem for spatial modules like conv).
- **(c)** Keeping masked tokens fixes this but gives up 4 – 10x training speed-up.
- **(d)** We can solve the issue with undesirable padding.
- **(e)** In Hiera, we side-step the problem entirely by changing the architecture so the kernels can’t overlap between mask units.



Bells-and-whistles are unnecessary when training with a strong pretext task (MAE)

Setting	Image acc.	Image im/s	Video acc.	Video clip/s
MViTv2-L Supervised	85.3	219.8	80.5	20.5
Hiera-L MAE				
a. replace rel pos with absolute *	85.6	253.3	85.3	20.7
b. replace convs with maxpools *	84.4	99.9†	84.1	10.4†
c. delete stride=1 maxpools *	85.4	309.2	84.3	26.2
d. set kernel size equal to stride	**85.7**	369.8	**85.5**	29.4
e. delete q attention residuals	85.6	374.3	**85.5**	29.8
f. replace kv pooling with MU attn	85.6	**531.4**	**85.5**	**40.8**

Without MAE pre-training:

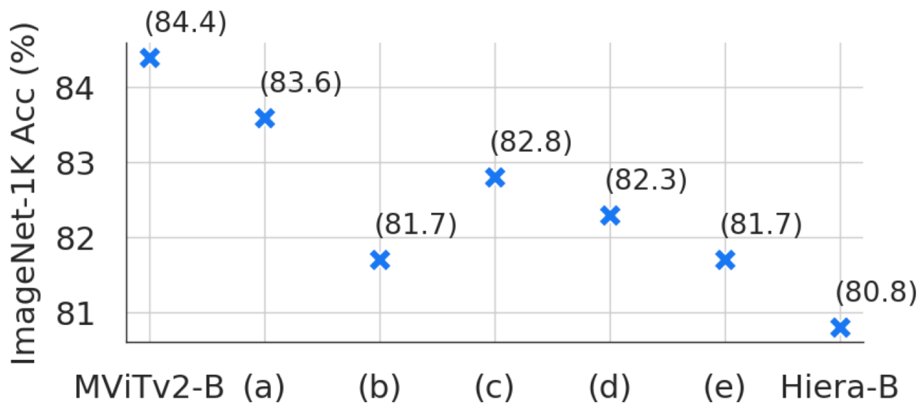
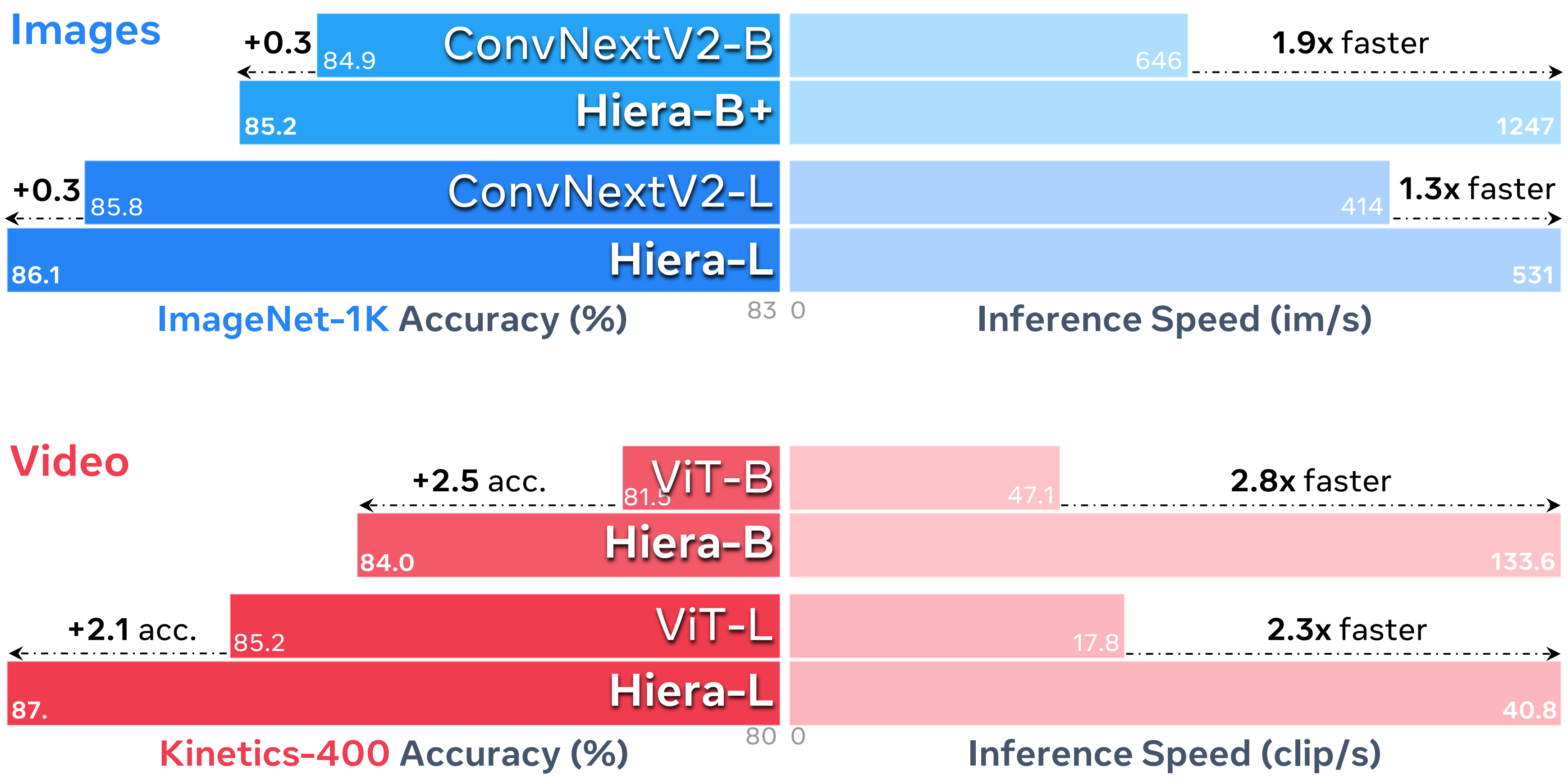
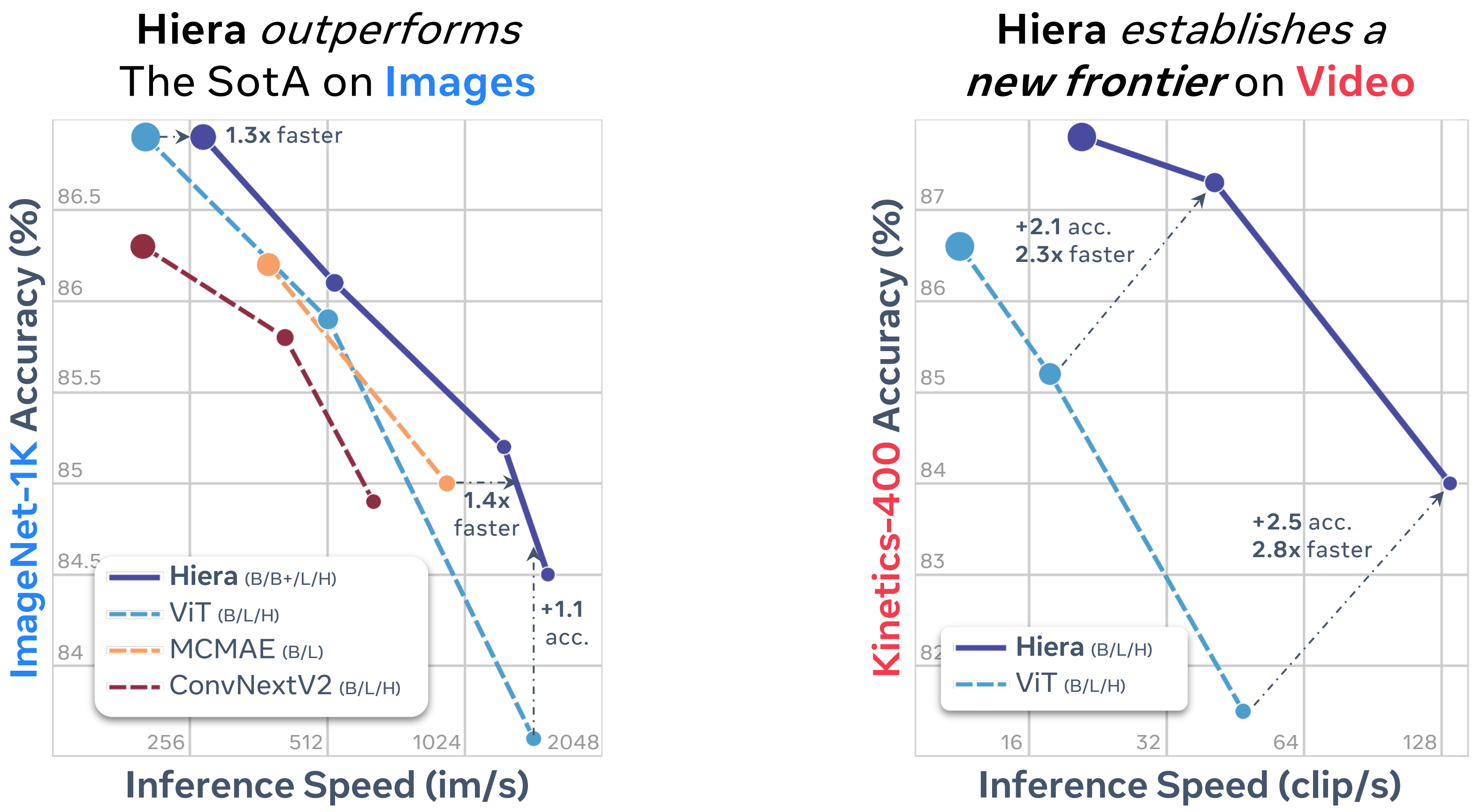


Figure 8. **Training on classification *from scratch*.** Here we repeat the experiment in Tab. 1 but without MAE pretraining, using MViTv2's supervised recipe instead. As expected, the bells-and-whistles that Hiera removes are actually *necessary* when training from scratch—hence their introduction in prior work in the first place. Hiera *learns* spatial biases instead.

Significant speedup over concurrent work



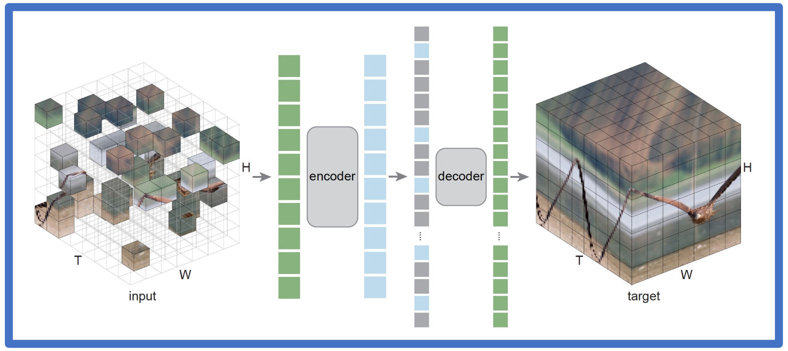
Hiera: Simple and fast



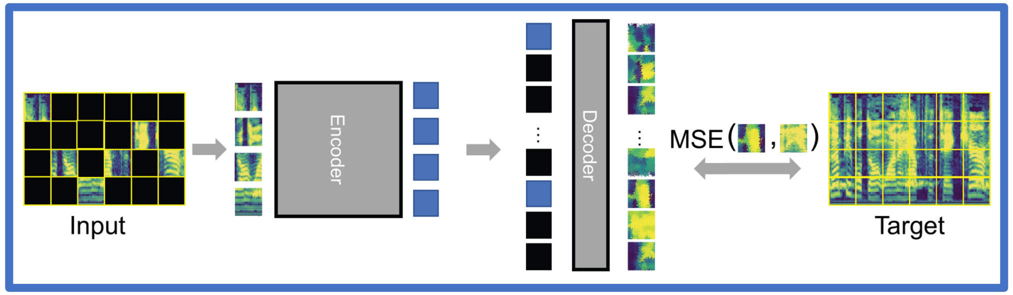
Summary: Self-supervised learning from video

- Video offers to learn by space-time prediction of appearance/shape, motion
- Video allows learning from spatiotemporal associations (across modalities)

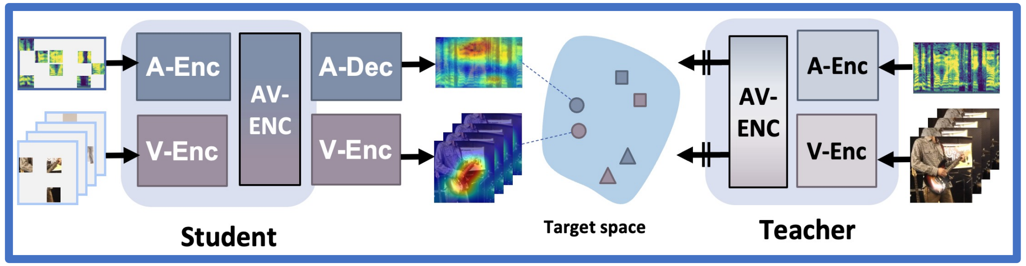
1. Video MAE



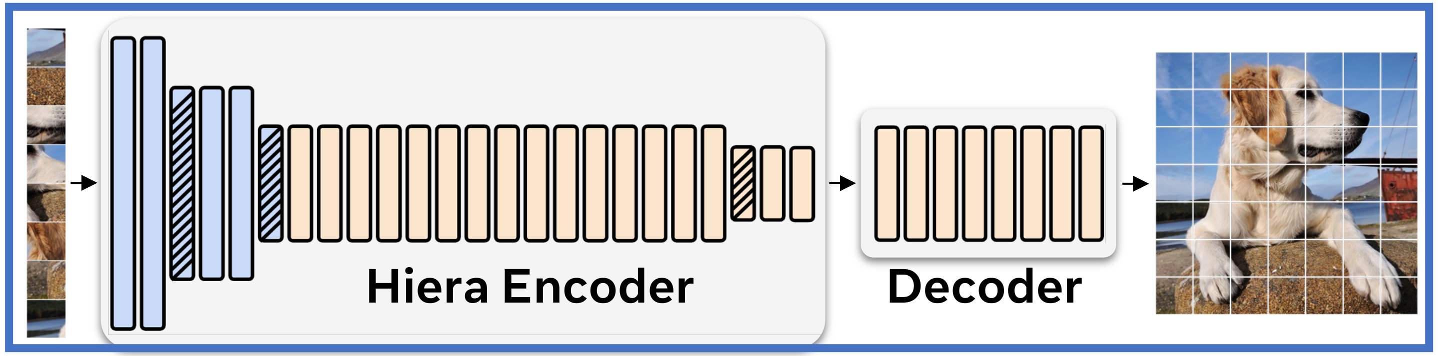
2. Audio MAE



3. Masked Audio-Video Learners



4. Hiera, a fast hierarchical transformer

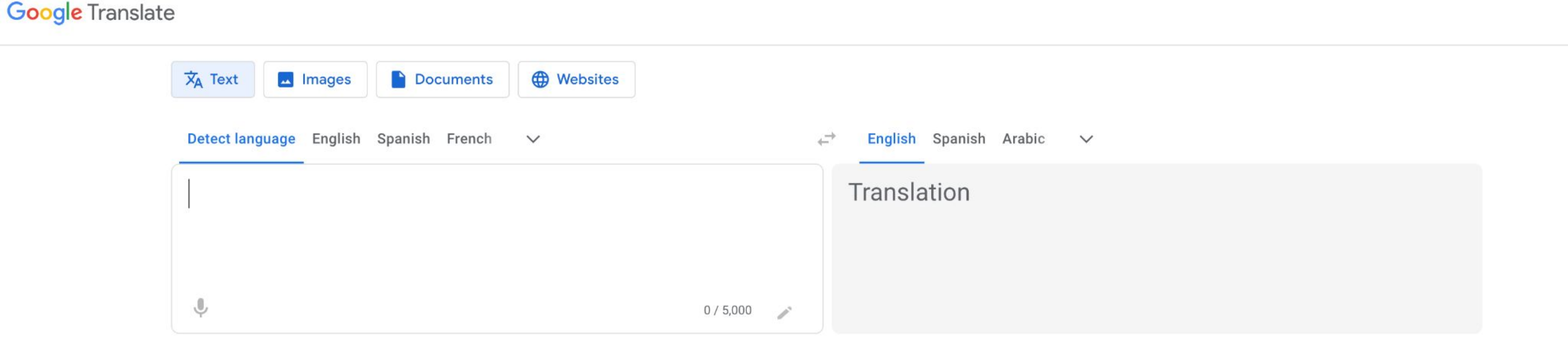


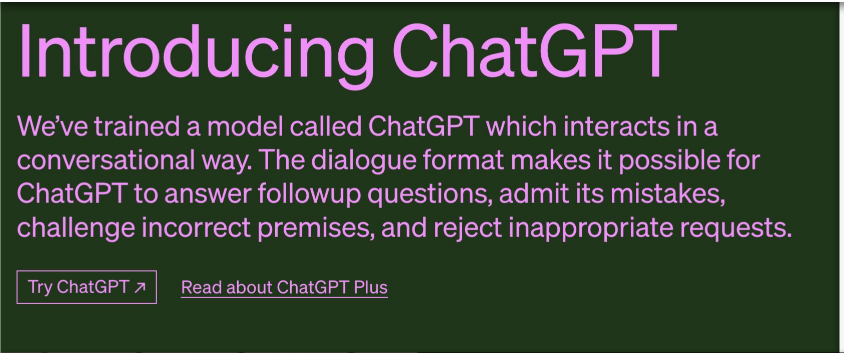
Self-Supervised Learning
from Research Advances to Best Practices

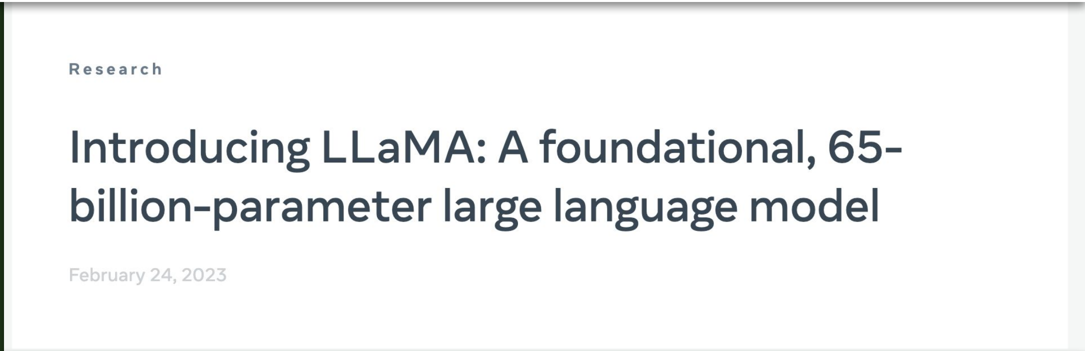
Part 2

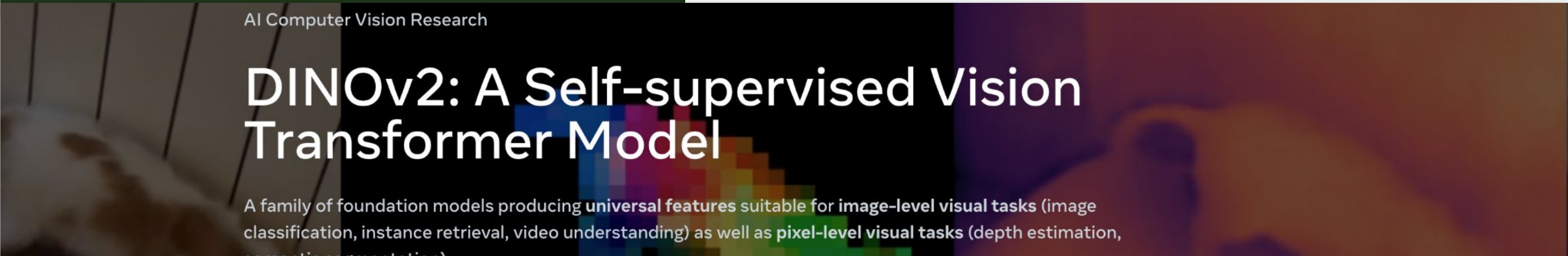
Ishan Misra, Mathilde Caron, Mark Ibrahim, Randall Balestriero

ICML 2023


Google Translate
Text
Images
Documents
Websites
Detect language
English
Spanish
French
English
Spanish
Arabic
Translation
0 / 5,000


Introducing ChatGPT
We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.
Try ChatGPT ↗
Read about ChatGPT Plus


Research
Introducing LLaMA: A foundational, 65-billion-parameter large language model
February 24, 2023


AI Computer Vision Research
DINOv2: A Self-supervised Vision Transformer Model
A family of foundation models producing universal features suitable for image-level visual tasks (image classification, instance retrieval, video understanding) as well as pixel-level visual tasks (depth estimation,

“The Dark Matter of Intelligence”
—Yann LeCun

One of the most promising ways to build background knowledge and approximate common sense.

https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/

Method?

SimCLR

MoCO

DINO

Masked Autoencdoer

BYOL

iBOT

JEPA

Barlow Twins

VICReg

&

Numerous other decisions

Projector type & Dimension

Softmax Temperature

Representation Dimension

Model Architecture

Model Representation Size

Training Length

Batch Size

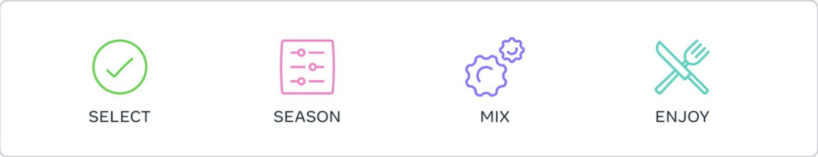
Augmentations

Evaluation method

Wisdom of Many
Self-Supervised Learning
Chefs

8 institutions

dozen+ researchers



1. Navigating the families of self-supervised learning methods

— Ishan

2. Recipes of best practices for training self-supervised learning methods

— Mathilde, Mark, Randall

Navigating the families of self-supervised learning methods

Ishan Misra

GenAI @ Meta AI

What is "self" supervision?

- Obtain "labels" from the data itself by using a "semi-automatic" process
- Predict part of the data from other parts
- Train a network using such a prediction task

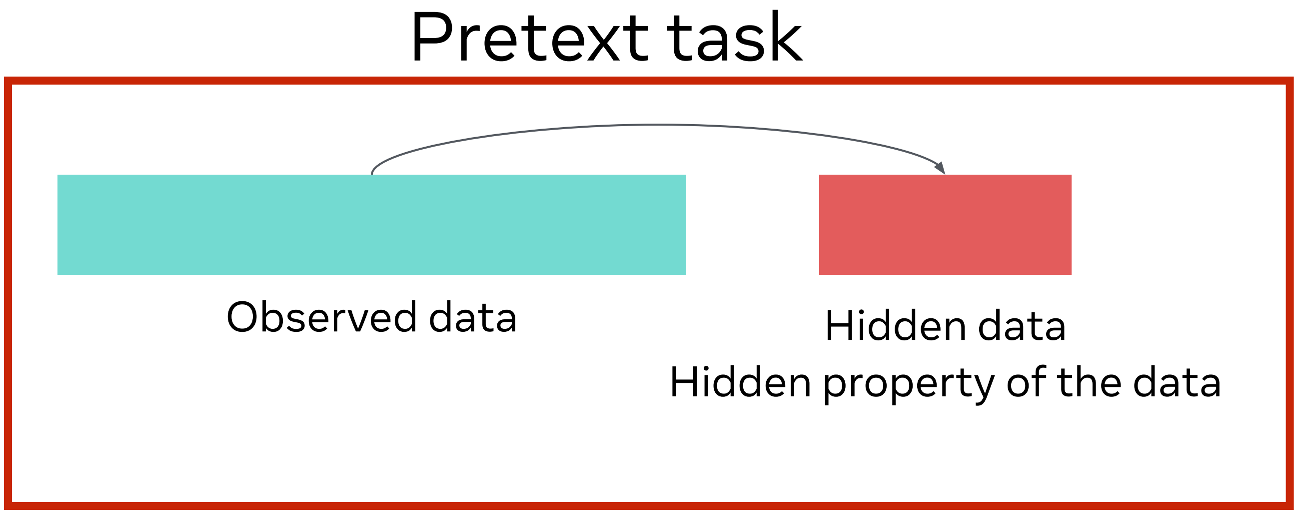
Why is it useful?

- Training data is "automatically generated"
- Ideally, for a downstream task that we care about, need less human supervision

In the context of Computer Vision

Pretext task

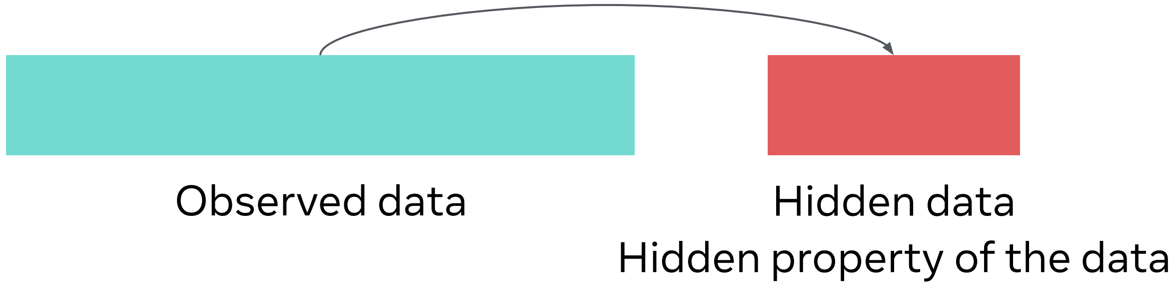
- Self-supervised task used for learning representations
- Often, not the "real" task (like image classification) we care about



Unsupervised visual representation learning by context prediction. Doersch et al., 2015

Pretext task

- Using images
- Using video
- Using video and sound



How to categorize SSL approaches?

Observed data

Hidden data
Hidden property of the data

Training

- Type of hidden data/property
- Loss function/Training mechanism

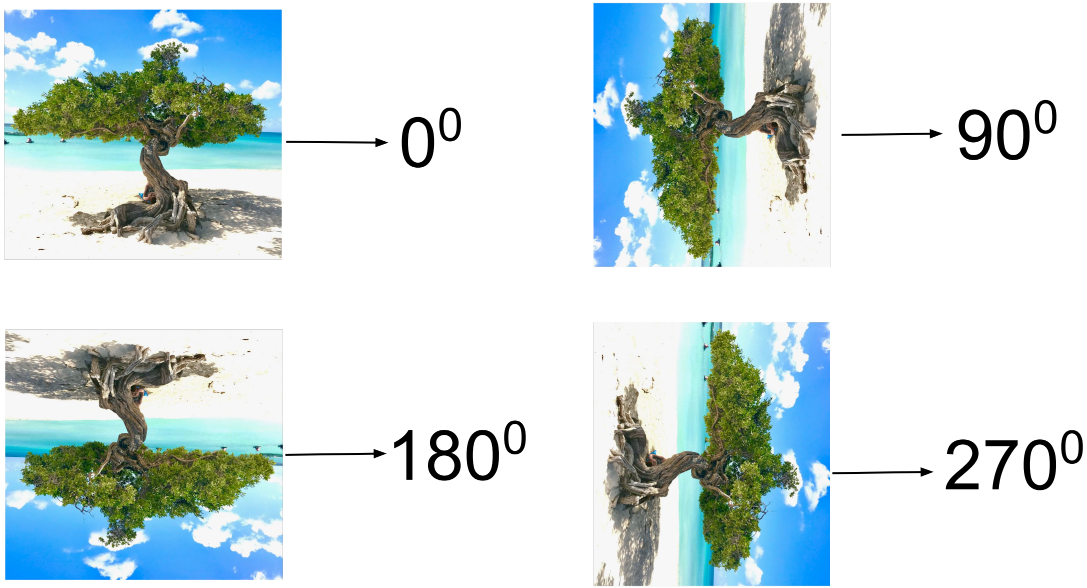
Performance based

- Ease of use
- Amount of training/supervision needed for downstream application

Type of hidden data/property

Doersch et al., 2015, Unsupervised visual representation learning by context prediction

Type of hidden data/property



Input: image rotated by [0, 90, 180, 270]

Output: 4-way classification

Gidaris et al., 2018, Predicting Image Rotations

Type of hidden data/property

- Masked Image Modeling (MIM)
- Predict missing pixel values

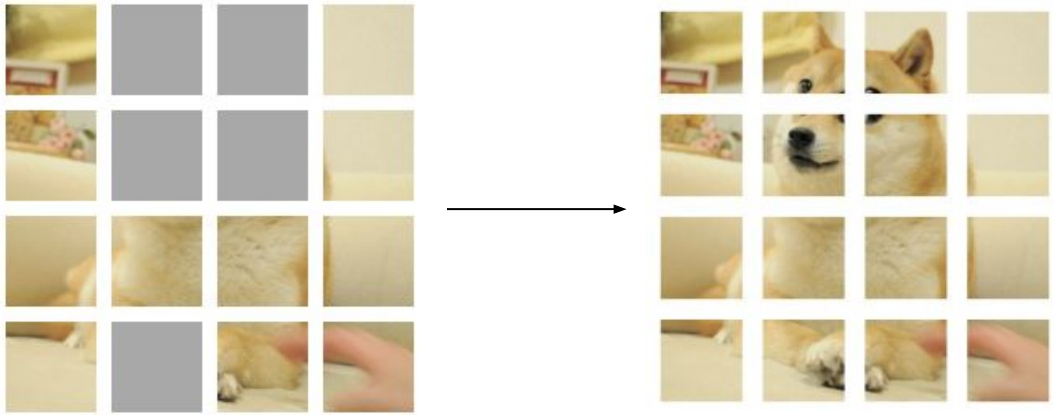


Image from: BEIT: BERT Pre-Training of Image Transformers - Bao et al., 2021
MAE - He et al., 2021

Type of hidden data/property

- Masked Image Modeling (MIM)
- Be robust to missing pixels in the input

Similar
Features

MSN Masked Siamese Networks - Assran et al., 2022

Type of hidden data/property

- Invariance
- Be robust to a large class of data augmentations

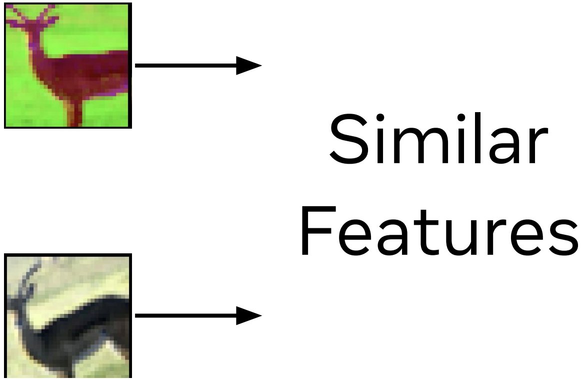
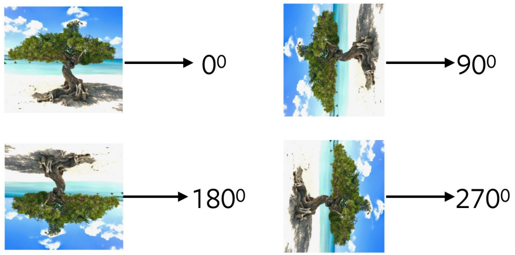


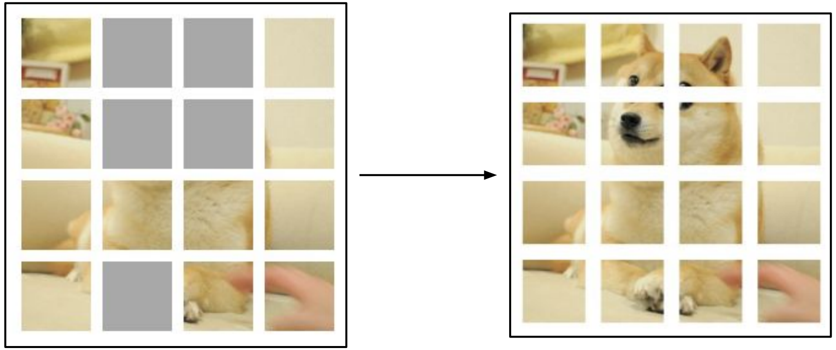
Figure from Dosovitskiy et al., 2014

Training/Loss Function

- Can be quite simple if the target is computed algorithmically



Discrete classification: Cross Entropy

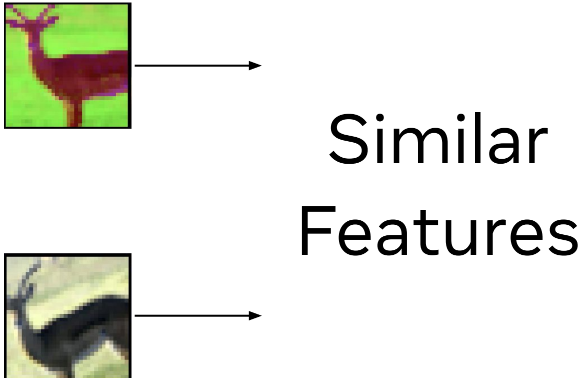


Discrete classification: Cross Entropy

Reconstruction of pixels: MSE

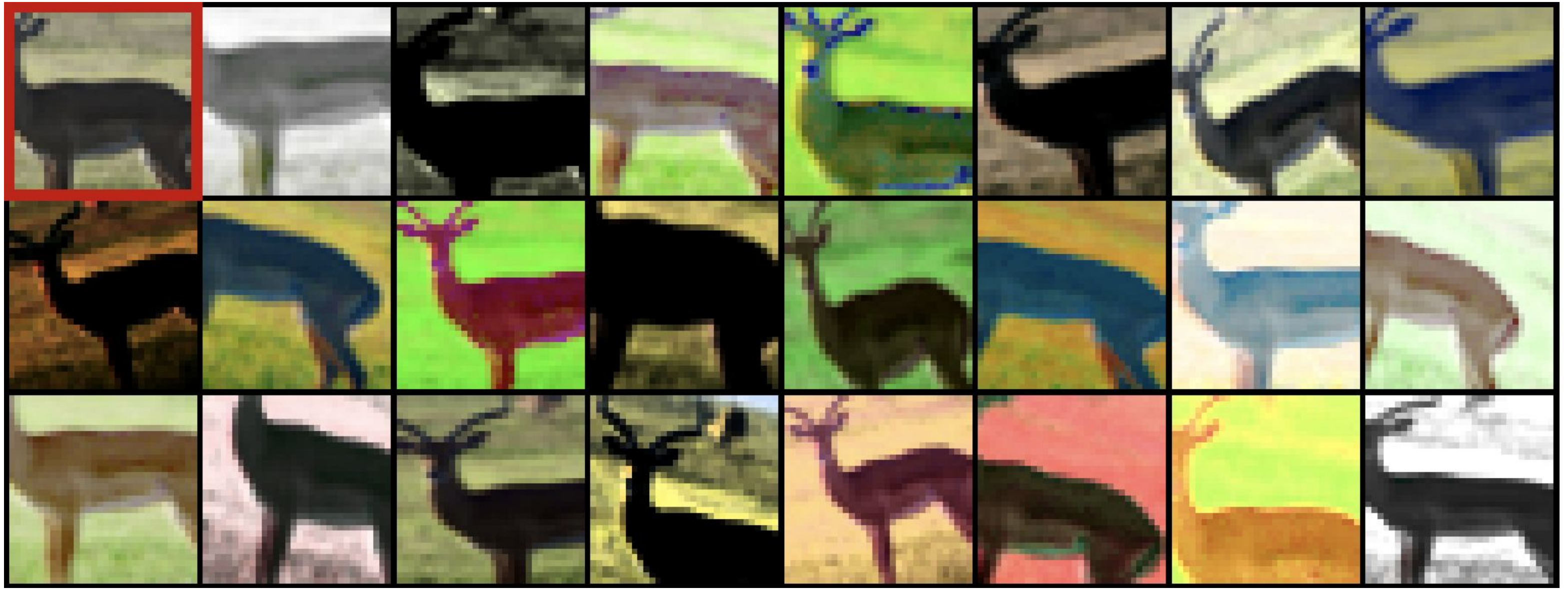
Training/Loss Function — Invariance methods

- Can be involved



Invariance based learning

- Being invariant to the data augmentation

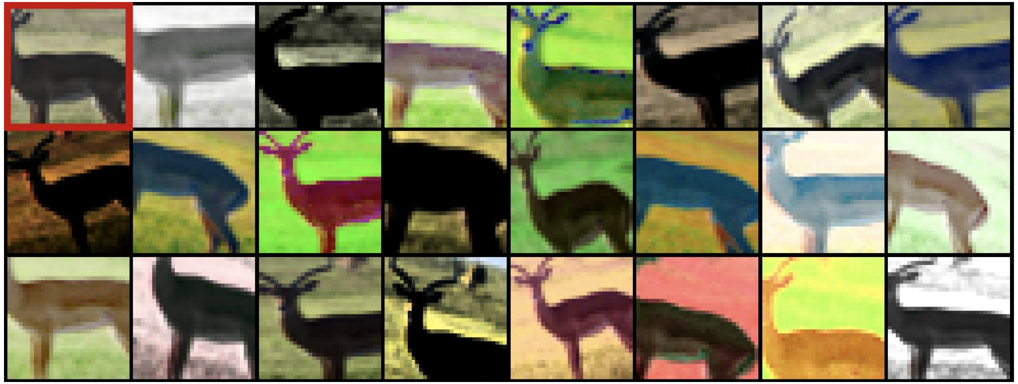


Learn features such that:

$$f_\theta(I) = f_\theta(\mathrm{augment}(I))$$

Figure from Dosovitskiy et al., 2014

Why is it useful?



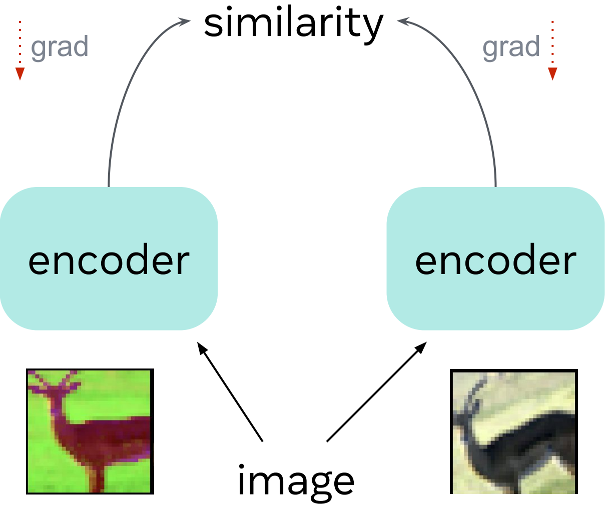
Learn features such that:

$$f_\theta(I) = f_\theta(\mathrm{augment}(I))$$

Learned features are invariant to "nuisance factors" or data augmentation

Can it work?

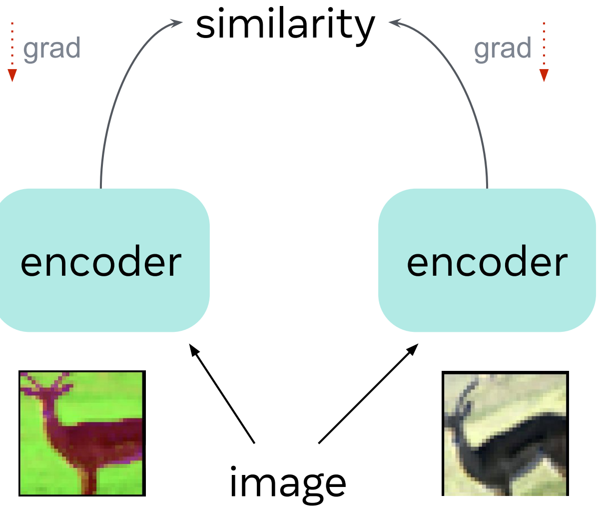
$$f_\theta(I) = f_\theta(\mathrm{augment}(I))$$



Trivial Solutions

$$f_\theta(I) = f_\theta(\text{augment}(I))$$

$$f_\theta(I) = \text{constant}$$



Satisfies the invariance property, but not useful

Invariant feature learning - Training/Loss categorization

Based on ways that they avoid trivial solutions

Invariant feature learning: ways to avoid trivial solutions

Similarity Maximization Objective

- Contrastive learning
  - MoCo, PIRL, SimCLR
- Clustering
  - DeepCluster, SeLA, SwAV
- Distillation
  - BYOL, SimSiam, DINO

Redundancy Reduction Objective

- Redundancy Reduction
  - Barlow Twins, VICReg

Many ways to avoid trivial solutions

Similarity Maximization Objective

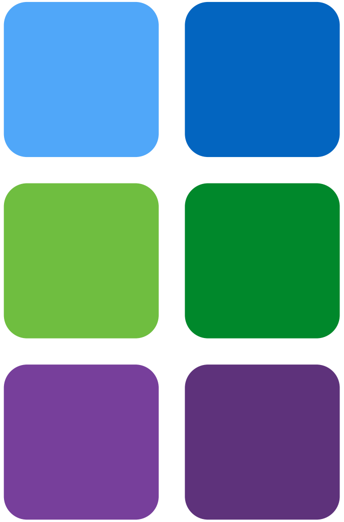
- Contrastive learning
  - MoCo, PIRL, SimCLR
- Clustering
  - DeepCluster, SeLA, SwAV
- Distillation
  - BYOL, SimSiam

Redundancy Reduction Objective

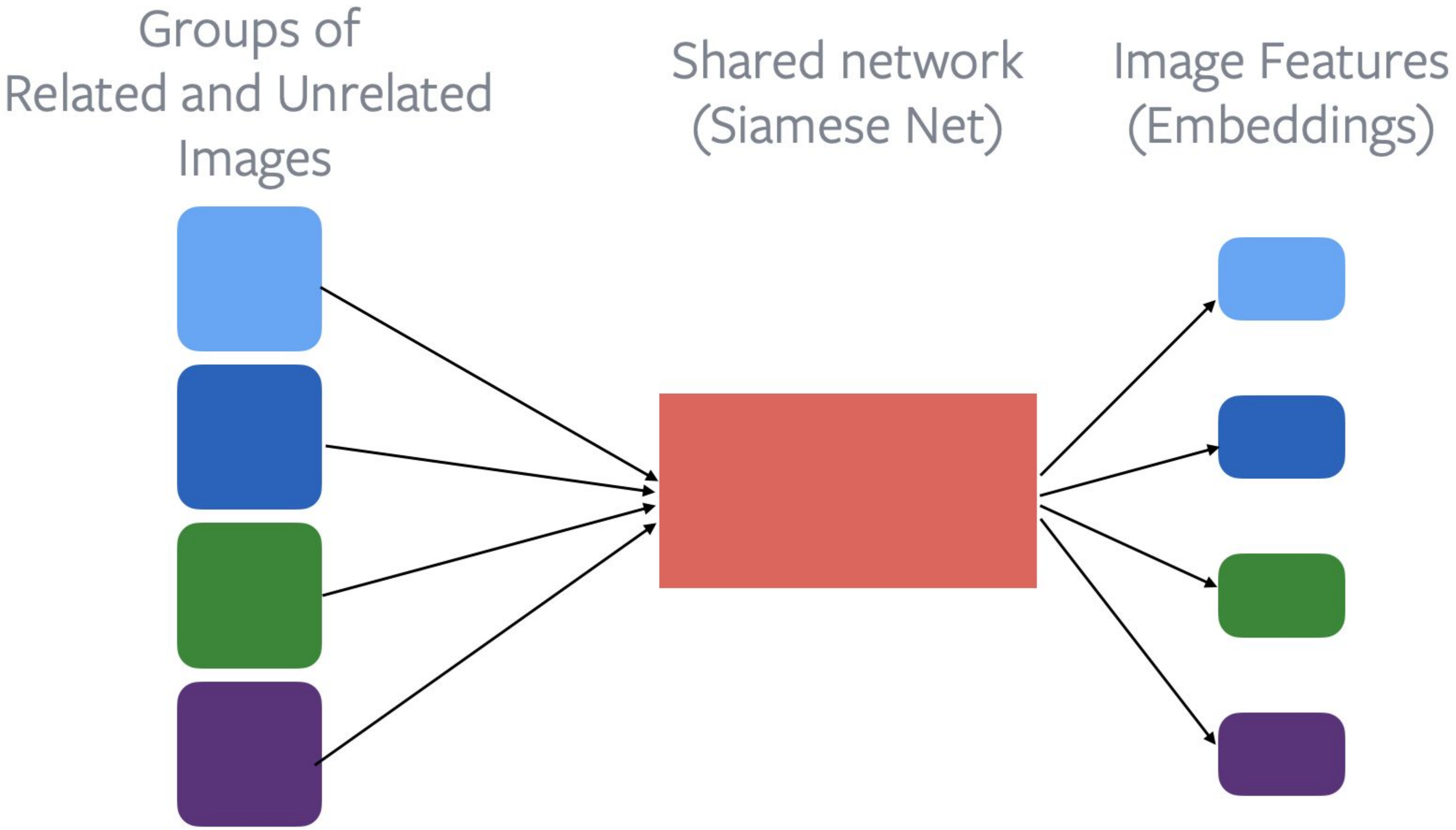
- Redundancy Reduction
  - Barlow Twins

Contrastive Learning

Groups of
Related and Unrelated
Images



Contrastive Learning

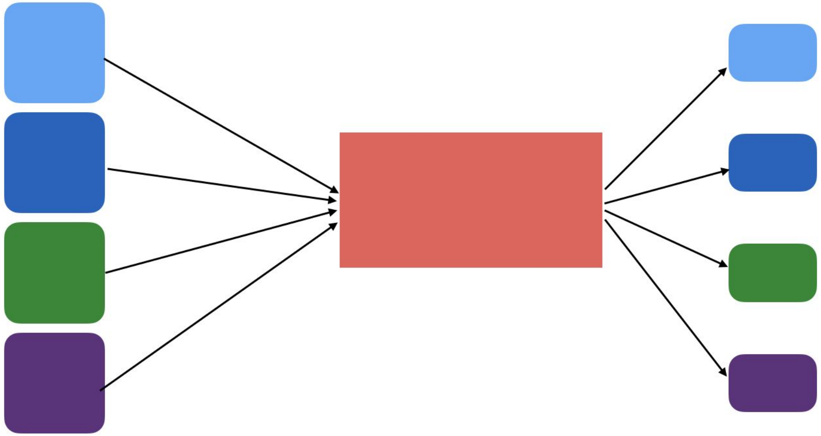


Contrastive Learning

Groups of Related and Unrelated Images

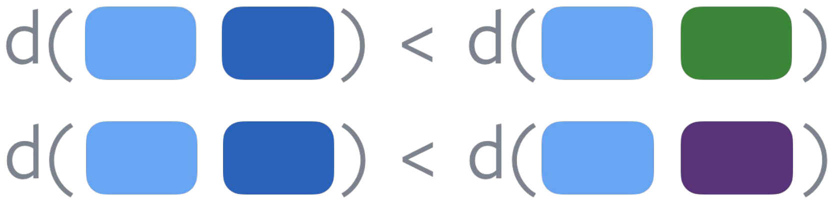
Shared network (Siamese Net)

Image Features (Embeddings)



Loss Function

Embeddings from related images should be closer than embeddings from unrelated images



Contrastive Learning in PIRL

Dataset

Loss Function

Image Feature &
Patch Features

Random Images

$\mathbf{I}$ $\mathbf{I}^t$

Self-Supervised Learning of Pretext-Invariant Representations. Misra et al. 2019

Nearby patches vs. distant patches of an Image

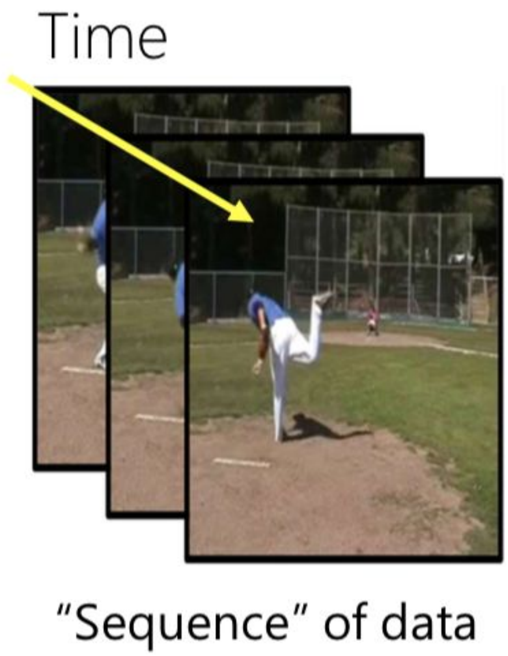
Related
(Positives)

Unrelated
(Negative)

van der Oord et al., 2018,
Henaff et al., 2019
Contrastive Predictive
Coding

Frames of a video

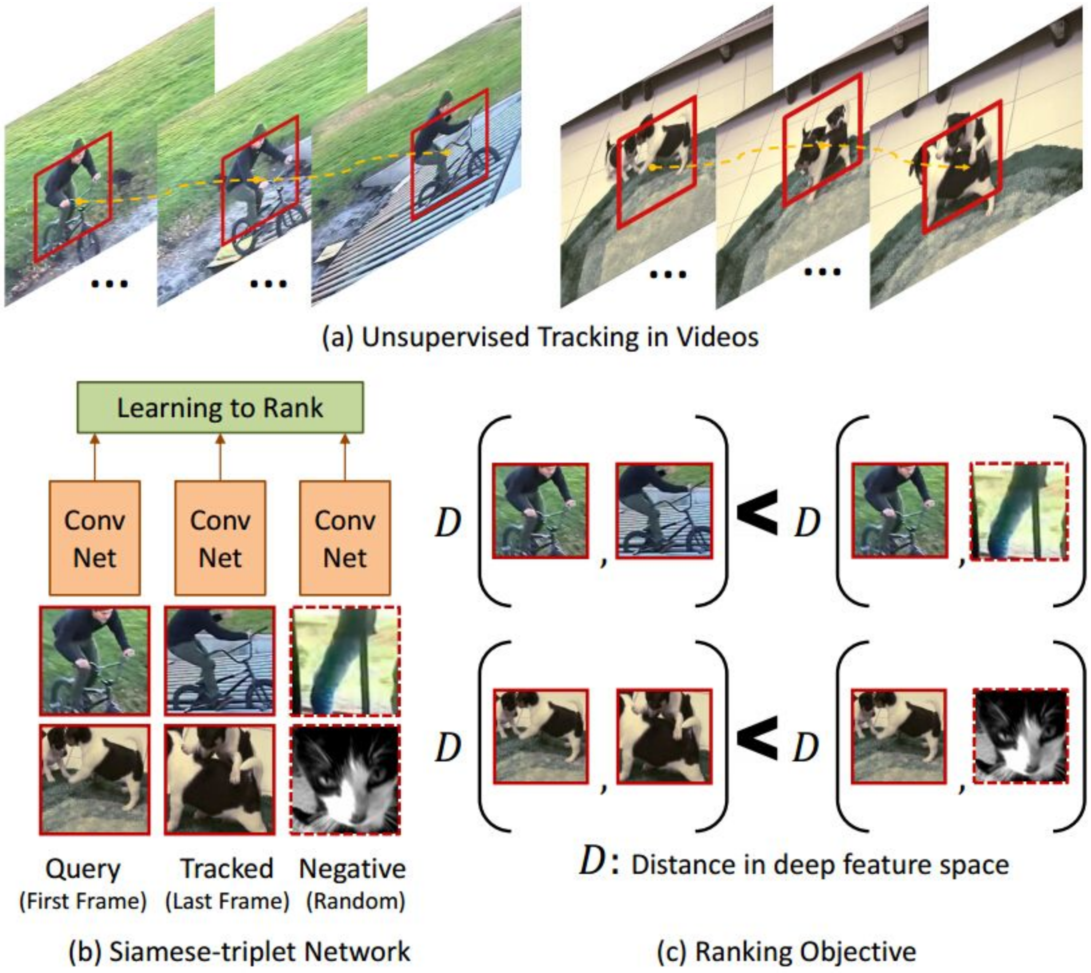
Video & Audio



Hadsell et al., 2005, DrLim
van der Oord et al., 2018, CPC

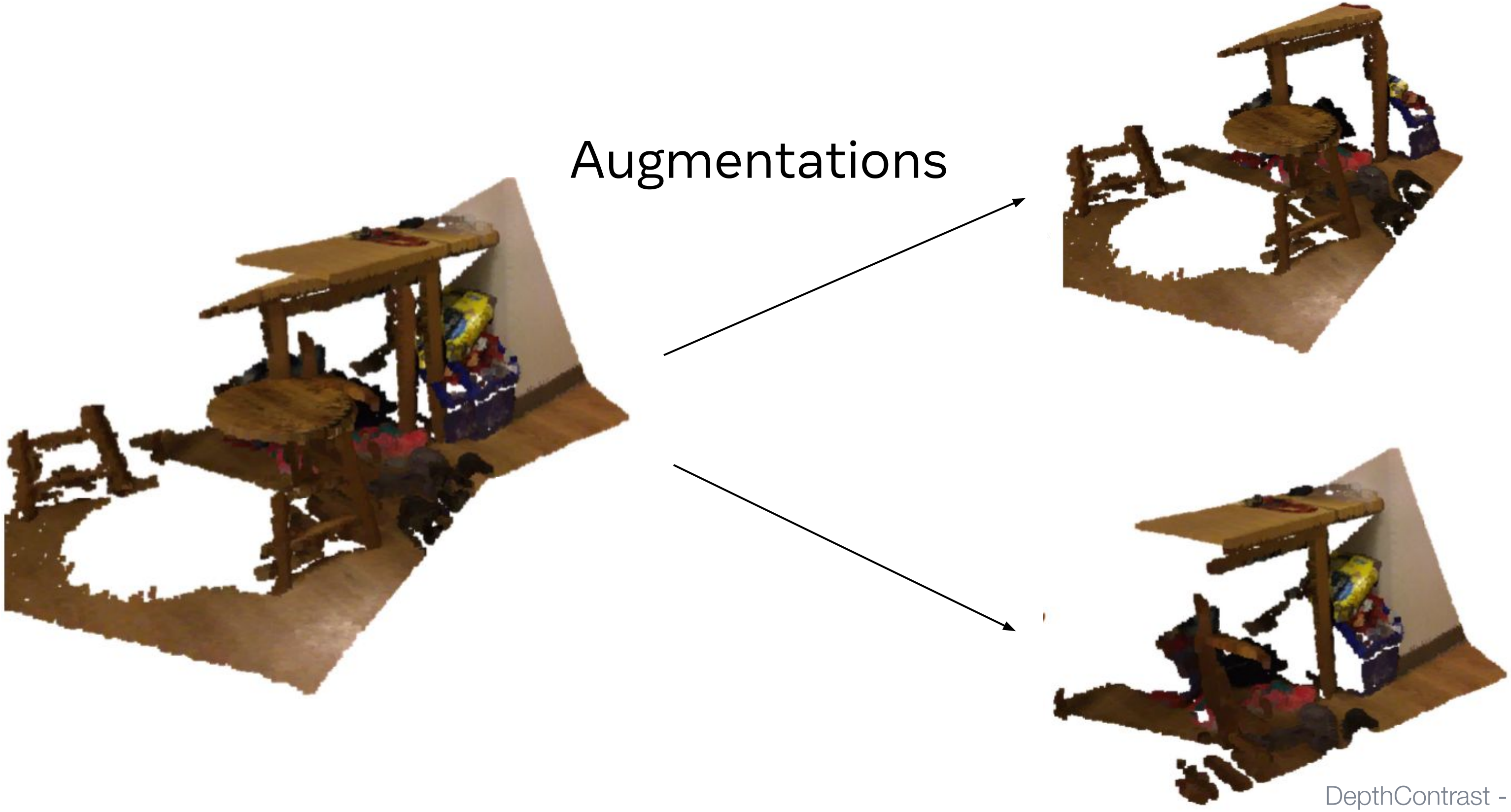
AVID+CMA - Morgado et al., 2020
GDT - Patrick et al., 2020

Tracking Objects



Unsupervised Learning of Visual Representations using Videos. Wang & Gupta, 2015.

3D Point Clouds

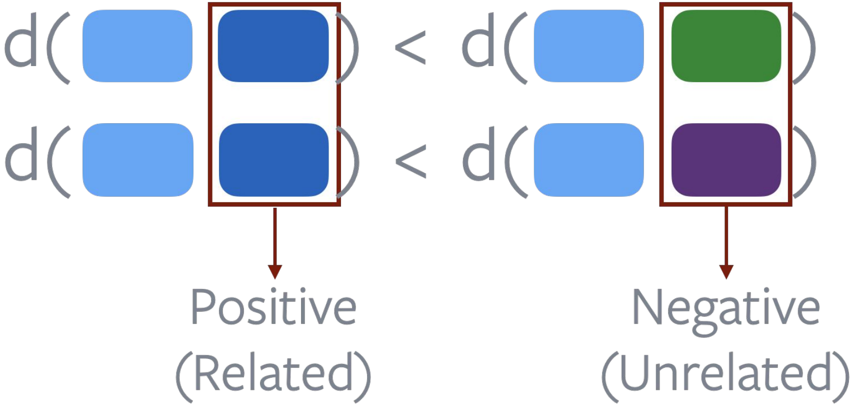


DepthContrast - Zhang et al., ICCV 2021
PointContrast Xie et al., CVPR 2020

Good negatives are necessary

Loss Function

Embeddings from related images should be closer than embeddings from unrelated images

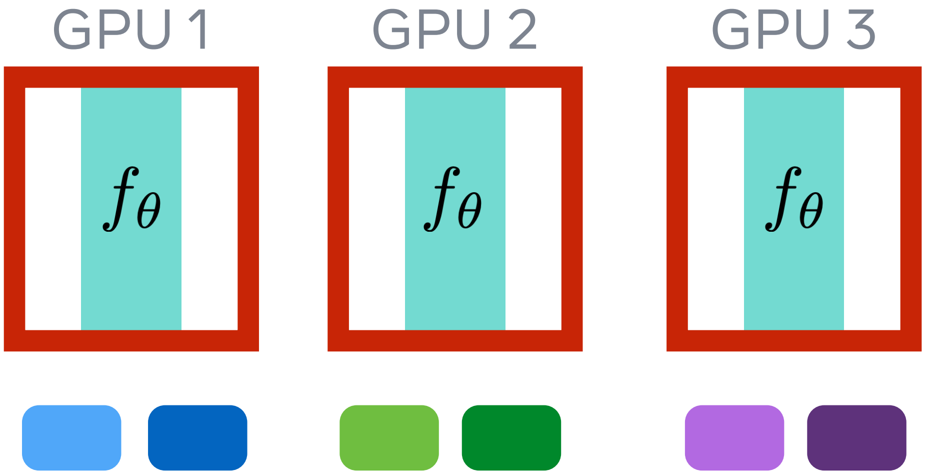


Good negatives are *very* important in contrastive learning

Hadsell et al., 2005, DrLim

SimCLR

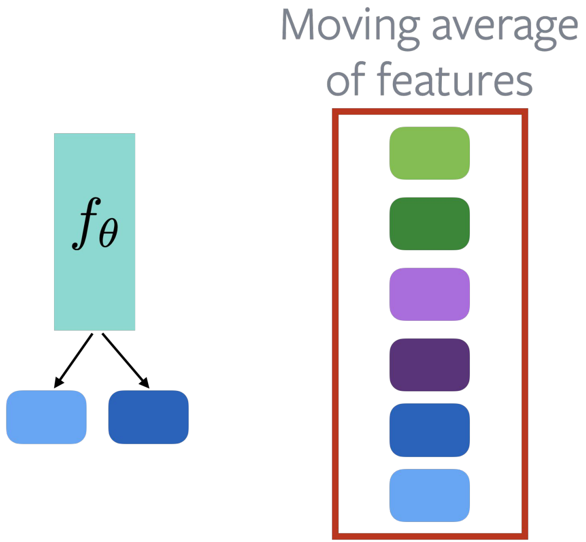
- Large batch size - e.g. in SimCLR
- Pros - Simple to implement
- Cons - Large batch size



A simple framework for contrastive learning - Chen et al., 2020

Memory Bank

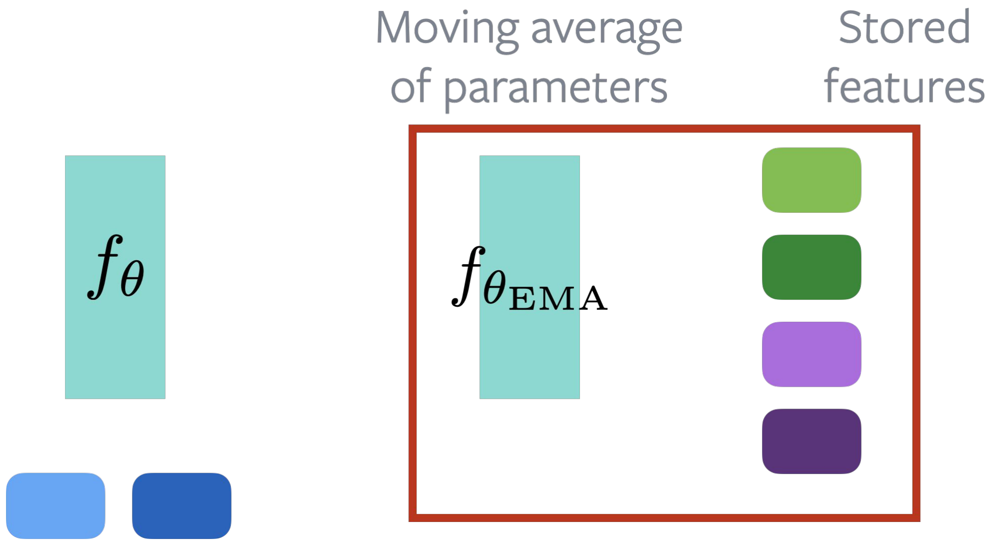
- Maintain a "memory bank" -- momentum of activations
- Pros - compute efficient
- Cons - Needs large memory, not "online"



Non parametric Instance Discrimination - Wu et al., 2018

MoCo

- Maintain "momentum" network - MoCo
- Pros - online, improved performance
- Cons - extra memory for parameters/stored features, extra fwd pass compared to memory bank



Momentum Contrast - He et al., 2019

Many ways to avoid trivial solutions

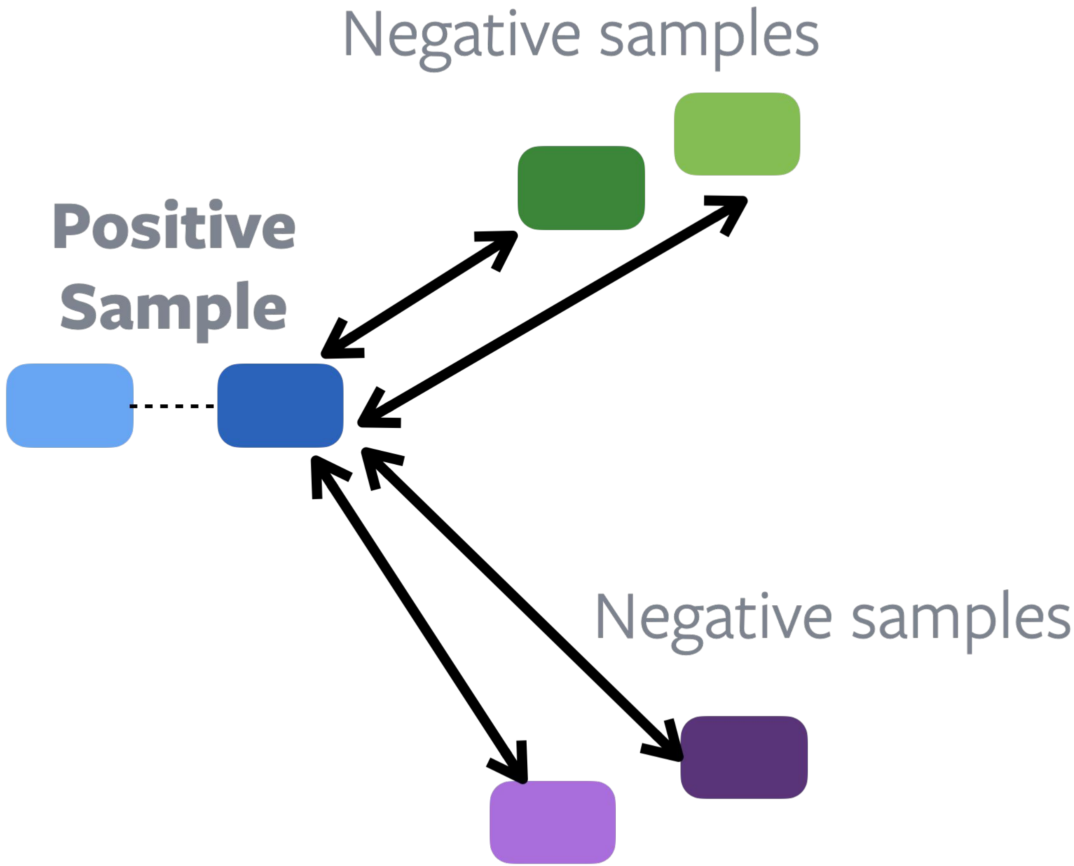
Similarity Maximization Objective

- Contrastive learning
  - MoCo, PIRL, SimCLR
- Clustering
  - DeepCluster, SeLA, SwAV
- Distillation
  - BYOL, SimSiam

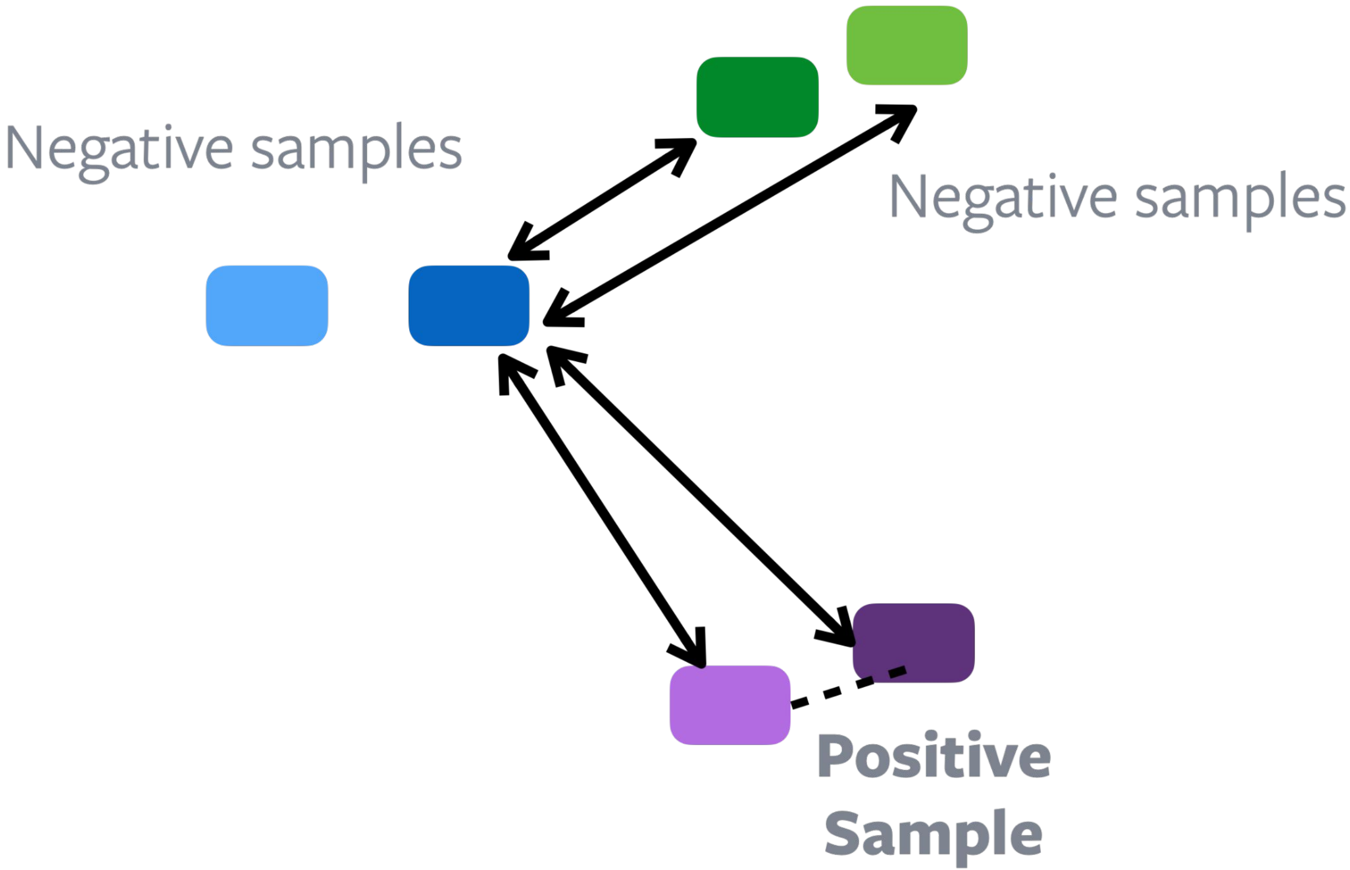
Redundancy Reduction Objective

- Redundancy Reduction
  - Barlow Twins

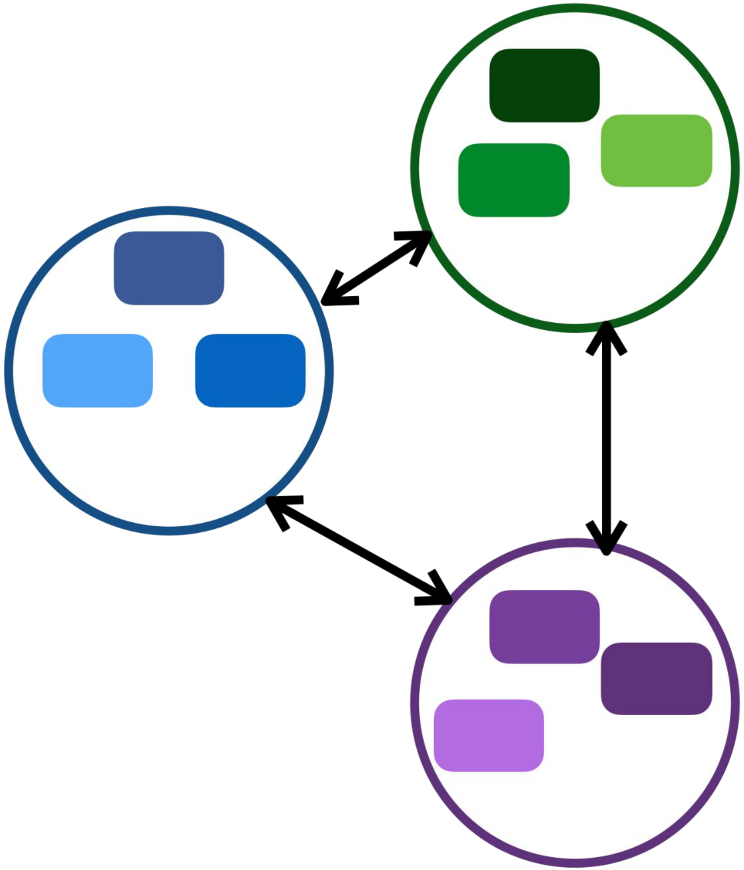
Contrastive learning -- what does it do?



Contrastive learning -- what does it do?

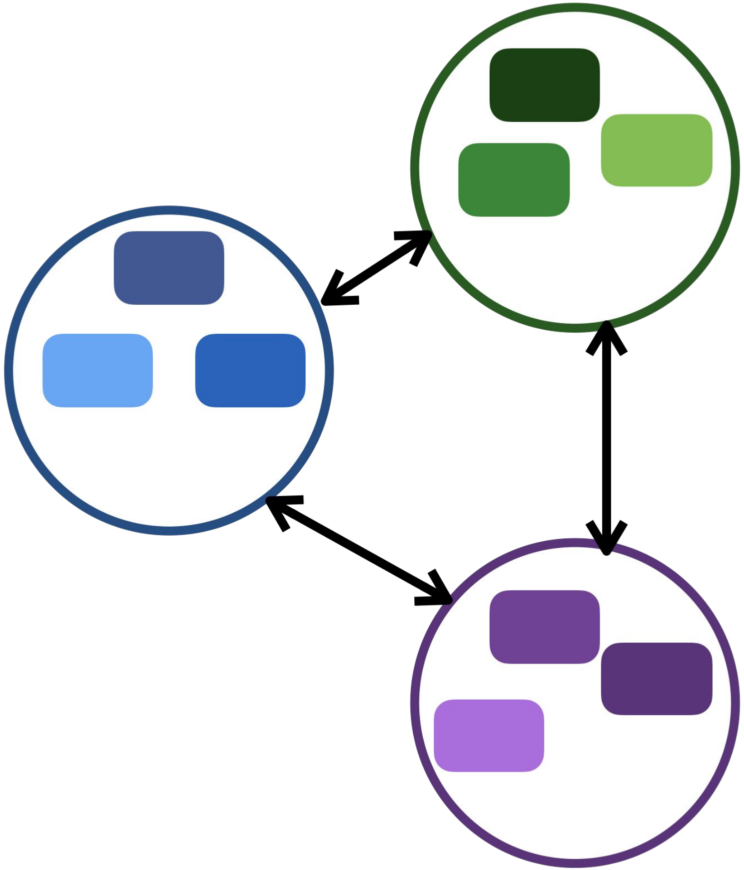


Contrastive Learning => Groups in feature space



Creates groups
in the feature space

Clustering creates groups too



Creates groups
in the feature space

So does **clustering**?!

Many ways to avoid trivial solutions

Similarity Maximization Objective

- Contrastive learning
  - MoCo, PIRL, SimCLR
- Clustering
  - DeepCluster, SeLA, SwAV
- Distillation
  - BYOL, SimSiam, DINO

Redundancy Reduction Objective

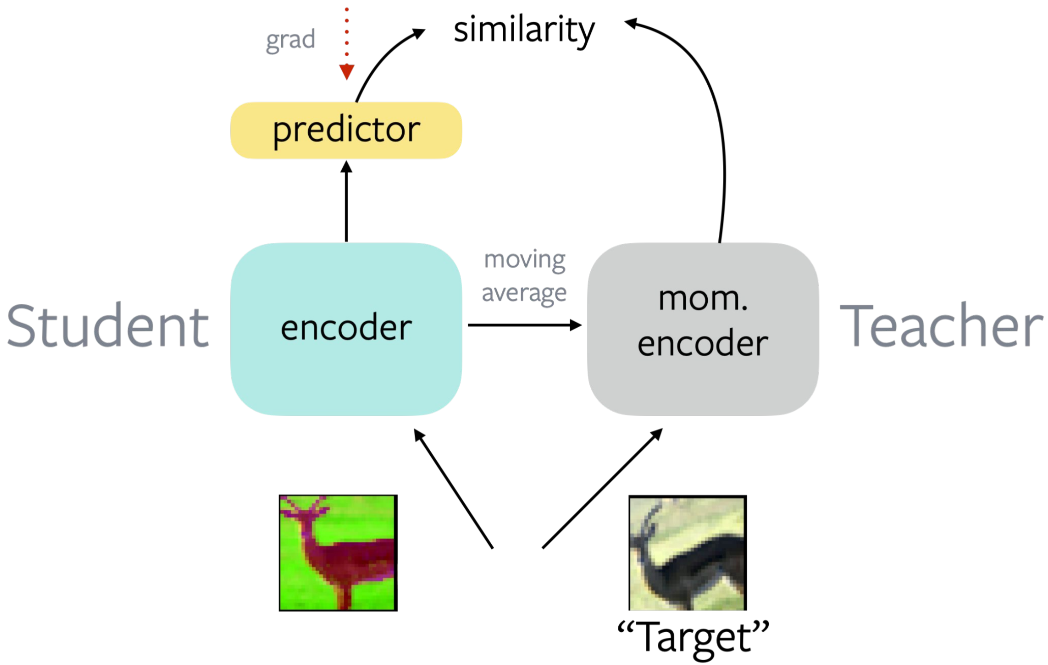
- Redundancy Reduction
  - Barlow Twins

“Self” Distillation

- What we want $f_\theta(I) = f_\theta(\text{augment}(I))$
- How we do it $f_\theta^{\text{student}}(I) = f_\theta^{\text{teacher}}(\text{augment}(I))$
- Prevent trivial solutions by asymmetry
  - Asymmetric **learning rule** between student teacher
  - Asymmetric **architecture** between student teacher

BYOL

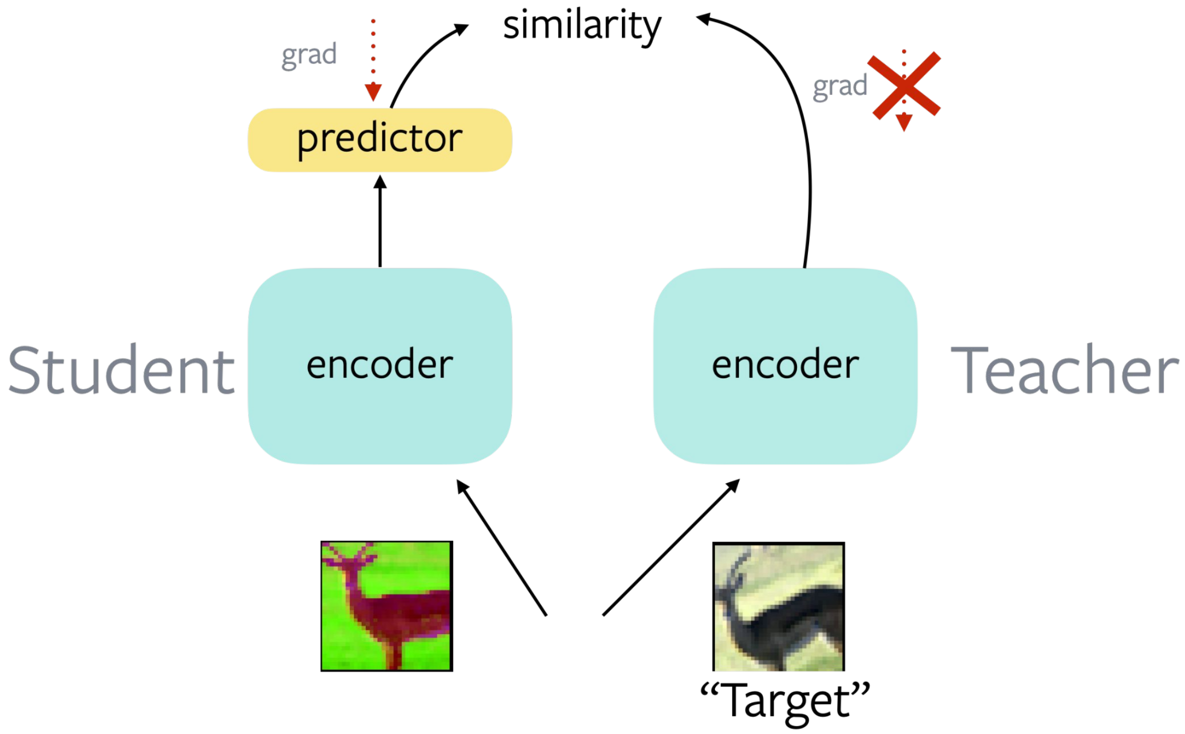
- What we want $f_\theta(I) = f_\theta(\text{augment}(I))$
- How we do it $f_\theta^{\text{student}}(I) = f_\theta^{\text{teacher}}(\text{augment}(I))$



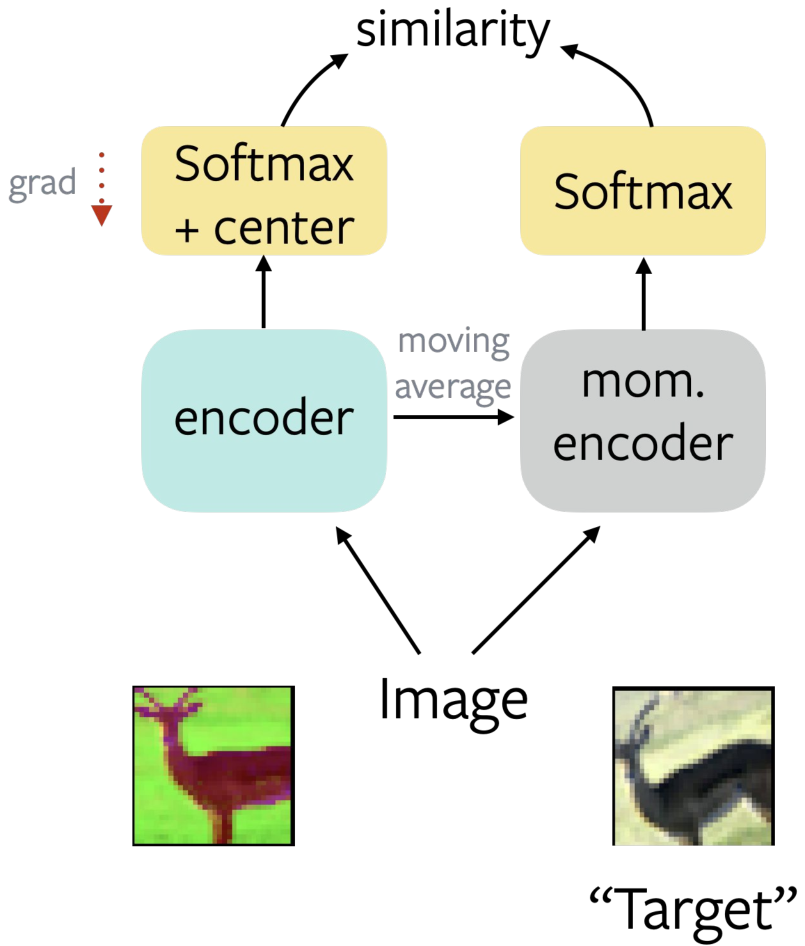
BYOL - Grill et al., 2020

SimSiam

- What we want $$f_\theta(I) = f_\theta(\text{augment}(I))$$

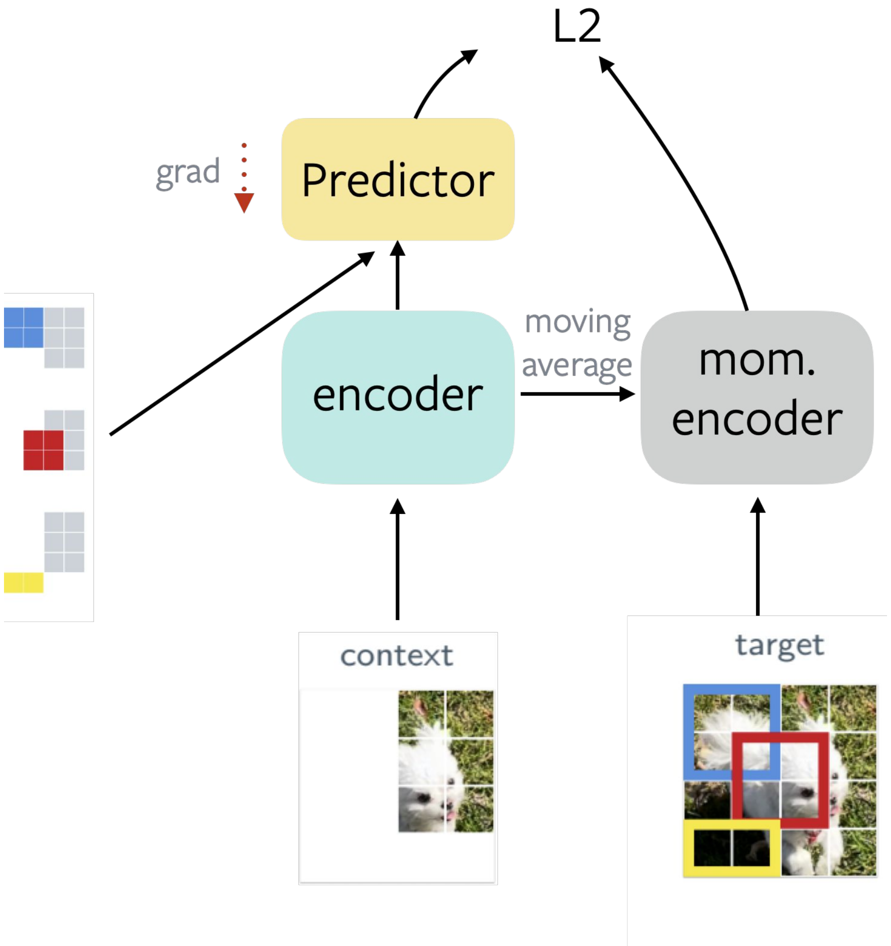


DINO - Main idea



DINO - Caron et al., 2022

I-JEPA - Main idea



I-JEPA - Assran et al., 2023

Many ways to avoid trivial solutions

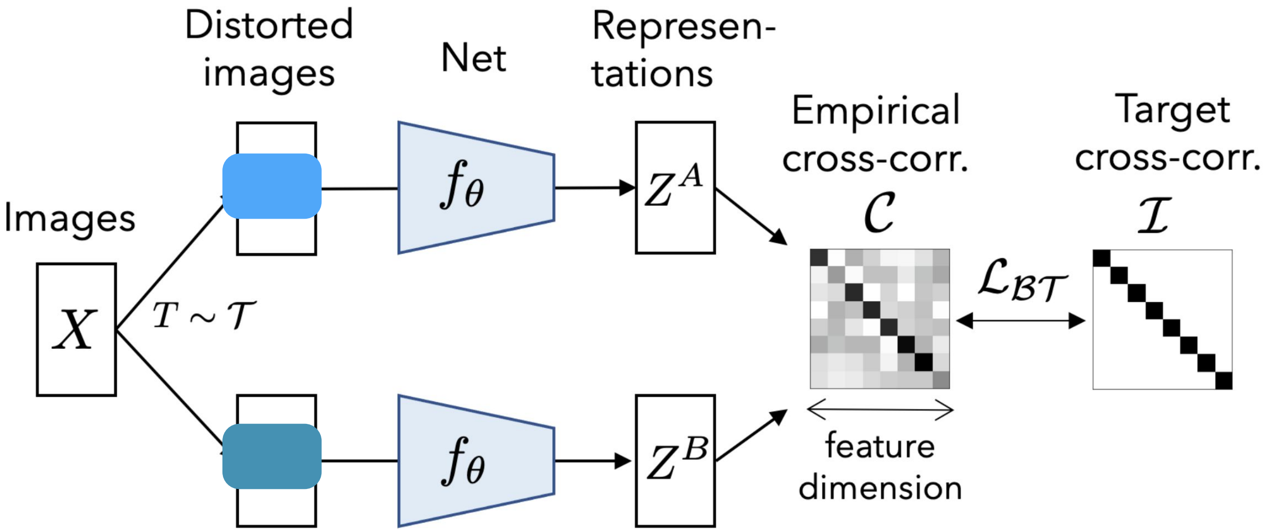
Similarity Maximization Objective

- Contrastive learning
  - MoCo, PIRL, SimCLR
- Clustering
  - DeepCluster, SeLA, SwAV
- Distillation
  - BYOL, SimSiam

Redundancy Reduction Objective

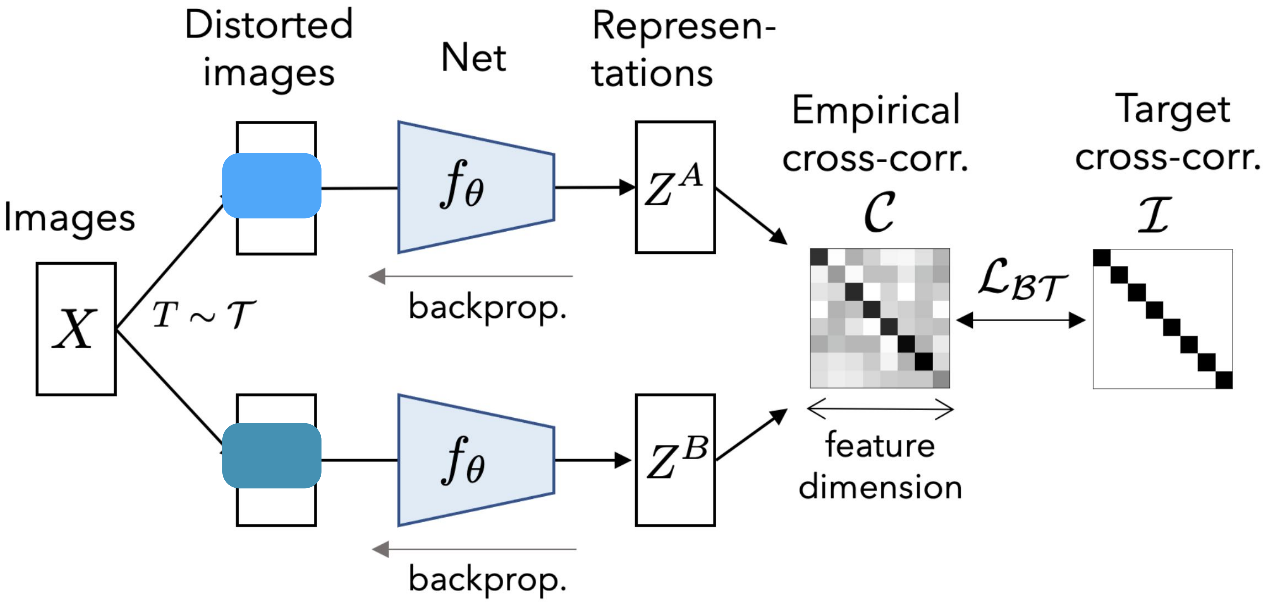
- Redundancy Reduction
  - Barlow Twins, VICReg

Barlow Twins - Loss



Barlow Twins - Zbontar et al., 2020

Barlow Twins - Loss



Barlow Twins - Zbontar et al., 2020

How to evaluate?

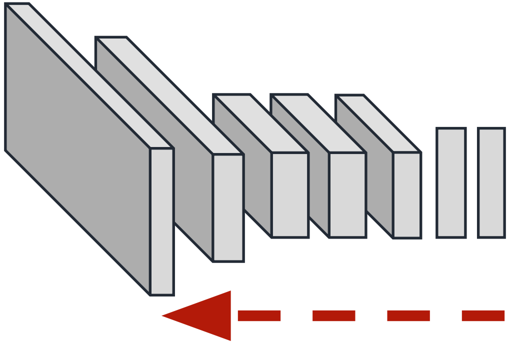
Most standard way

Use the pretrained network from self-supervised learning

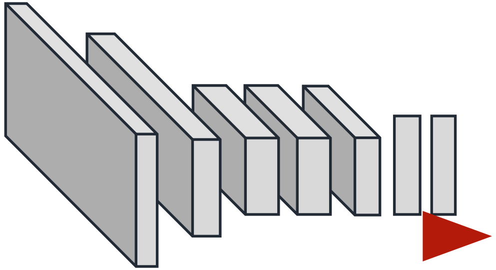
Use some amount of labeled data for the downstream task

Measure performance

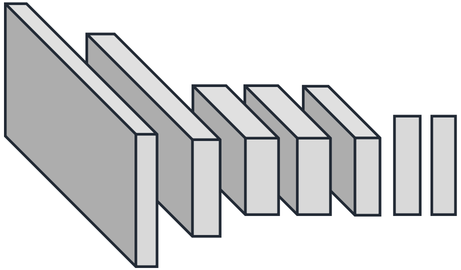
How to use the labeled data?



Fine-tune all layers



Linear classifier



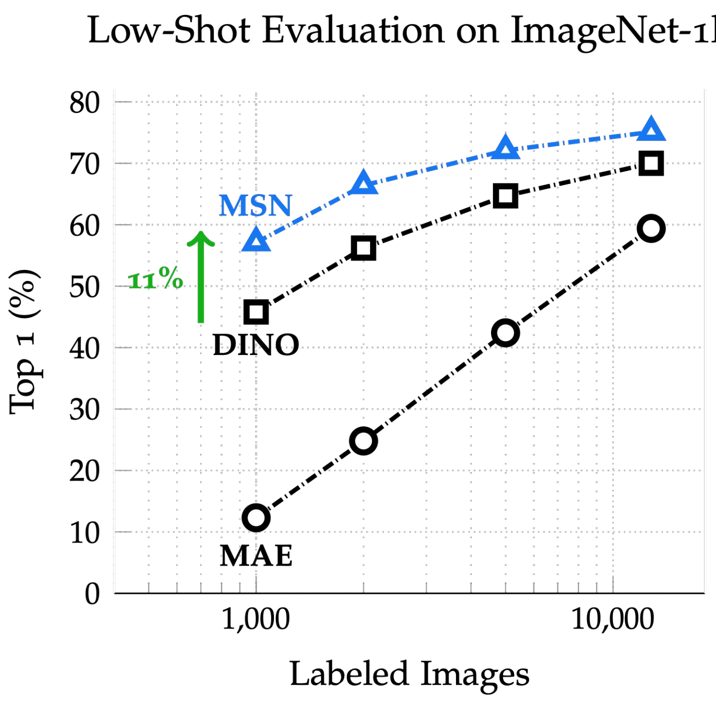
kNN

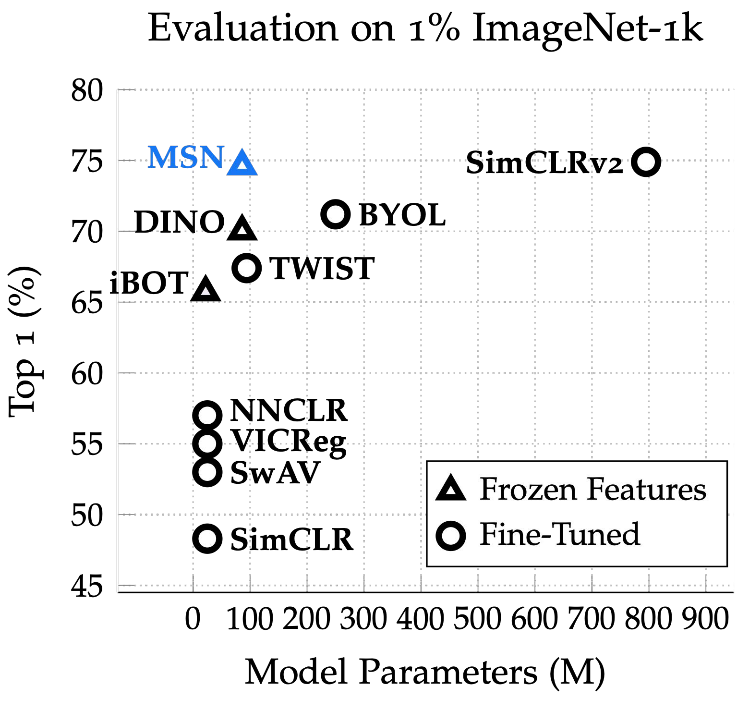
How much labeled data to use?

Most important factor

Typically not measured in academic papers

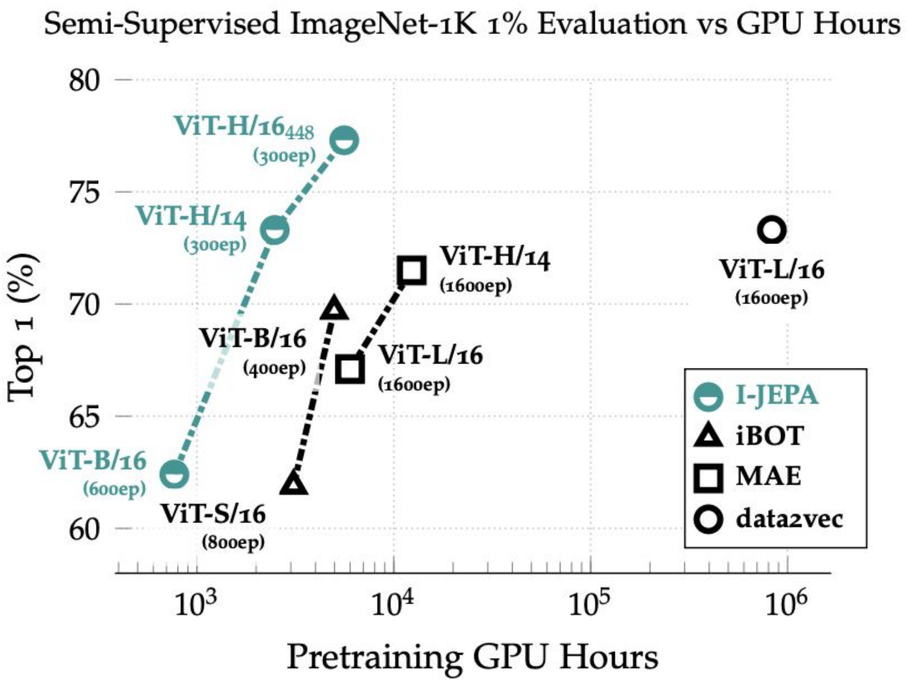
Label-efficient learning





MSN Masked Siamese Networks - Assran et al., 2022

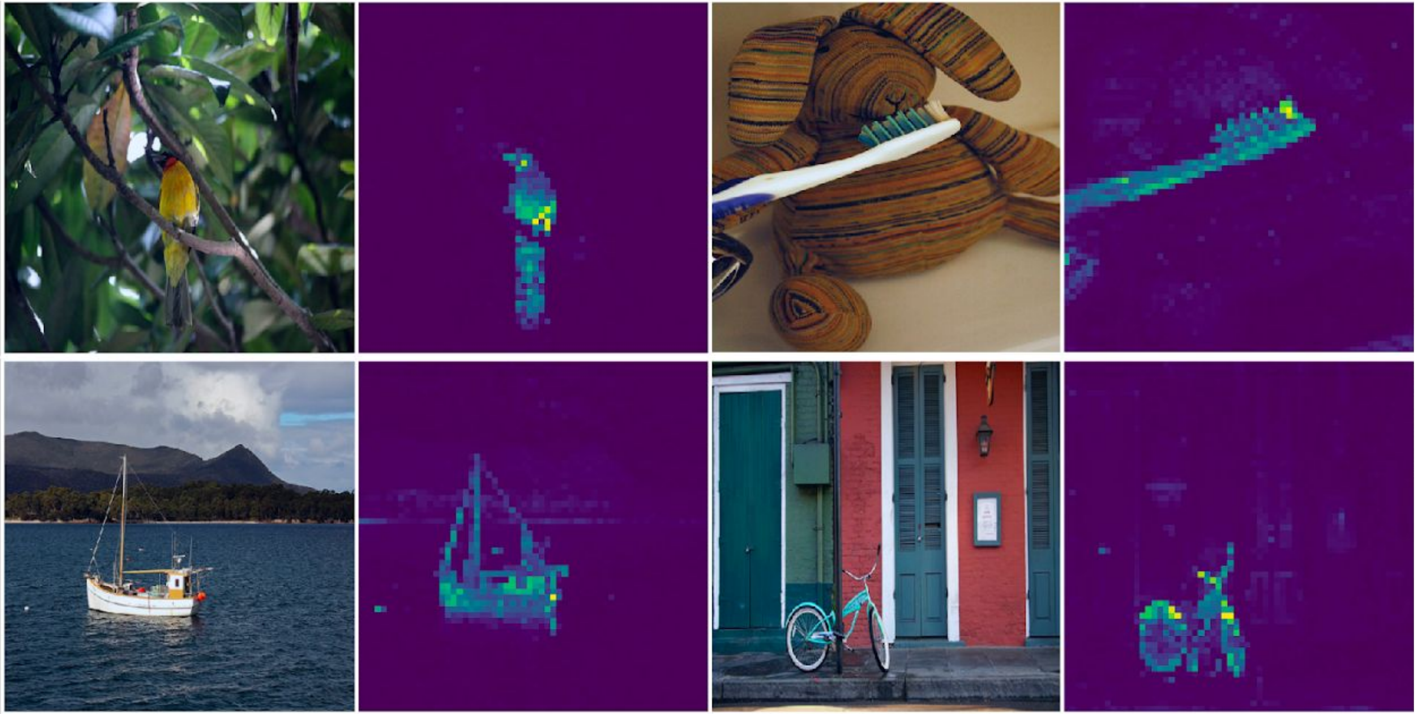
Pretraining time vs. Performance



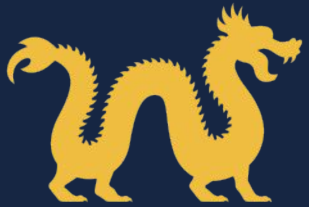
Label efficient **and** compute efficient

I-JEPA - Assran et al., 2023

Are the models useful without any labeled data?



DINO - Caron et al., 2022



dragon

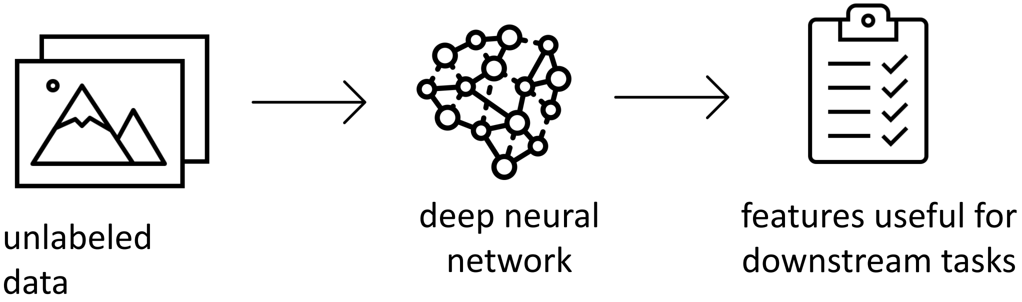
How to train your ~~self-supervised feature~~ extractor ?

Mathilde Caron

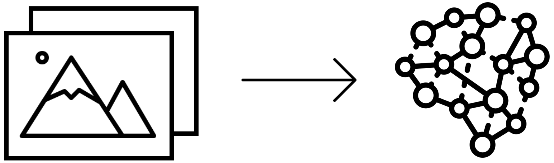
@Google Research

Practical use case

You have access to unlabeled data and you want to leverage these to learn a good feature space.



Option 1: Re-use opensourced models



You can directly download SSL models and use them to extract features on your data.

github.com/facebookresearch/dino

Pretrained models

You can choose to download only the weights of the pretrained backbone used for downstream tasks, or the full checkpoint which contains backbone and projection head weights for both student and teacher networks. We also provide the backbone in `onnx` format, as well as detailed arguments and training/evaluation logs. Note that `DeiT-S` and `ViT-S` names refer exactly to the same architecture.

arch	params	k-nn	linear	download					
ViT-S/16	21M	74.5%	77.0%	backbone only	full ckpt	onnx	args	logs	eval logs
ViT-S/8	21M	78.3%	79.7%	backbone only	full ckpt	onnx	args	logs	eval logs
ViT-B/16	85M	76.1%	78.2%	backbone only	full ckpt	onnx	args	logs	eval logs
ViT-B/8	85M	77.4%	80.1%	backbone only	full ckpt	onnx	args	logs	eval logs
ResNet-50	23M	67.5%	75.3%	backbone only	full ckpt	onnx	args	logs	eval logs

We also release XCiT models ([`arXiv`] [`code`]) trained with DINO:

arch	params	k-nn	linear	download				
xcit_small_12_p16	26M	76.0%	77.8%	backbone only	full ckpt	args	logs	eval
xcit_small_12_p8	26M	77.1%	79.2%	backbone only	full ckpt	args	logs	eval
xcit_medium_24_p16	84M	76.4%	78.8%	backbone only	full ckpt	args	logs	eval
xcit_medium_24_p8	84M	77.9%	80.3%	backbone only	full ckpt	args	logs	eval

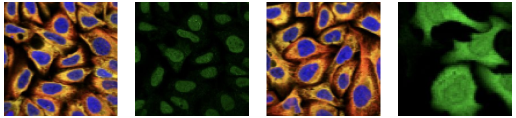
Pretrained models on PyTorch Hub

```
import torch
vits16 = torch.hub.load('facebookresearch/dino:main', 'dino_vits16')
vits8 = torch.hub.load('facebookresearch/dino:main', 'dino_vits8')
vitb16 = torch.hub.load('facebookresearch/dino:main', 'dino_vitb16')
vitb8 = torch.hub.load('facebookresearch/dino:main', 'dino_vitb8')
xcit_small_12_p16 = torch.hub.load('facebookresearch/dino:main', 'dino_xcit_small_12_p16')
xcit_small_12_p8 = torch.hub.load('facebookresearch/dino:main', 'dino_xcit_small_12_p8')
xcit_medium_24_p16 = torch.hub.load('facebookresearch/dino:main', 'dino_xcit_medium_24_p16')
xcit_medium_24_p8 = torch.hub.load('facebookresearch/dino:main', 'dino_xcit_medium_24_p8')
resnet50 = torch.hub.load('facebookresearch/dino:main', 'dino_resnet50')
```

However, there might be a domain gap…

- For example, opensourced SSL models is pre-trained on natural looking images:

- But, your data looks like this:

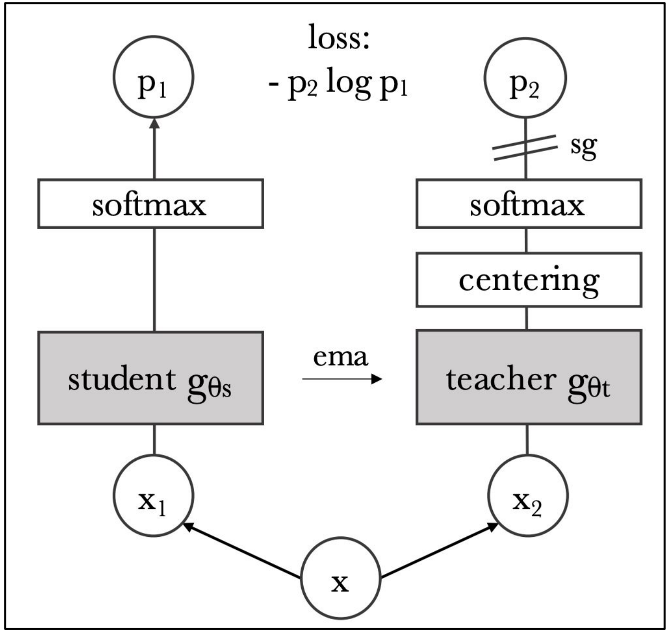


Solution: SSL training on your data

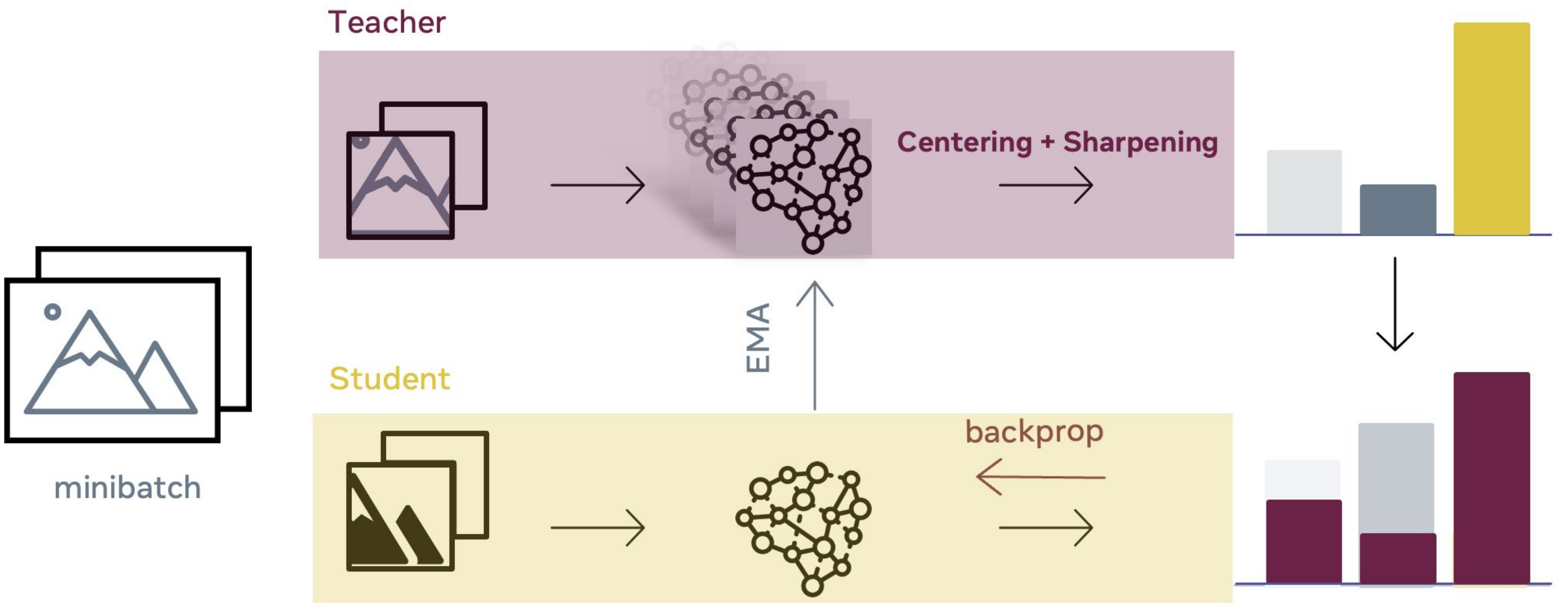
Option 2: Train SSL models on your data

- Most SSL algorithms look pretty simple to train :D !

DINO, Caron et al. ICCV 2021



The DINO training algorithm



Code snippet

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

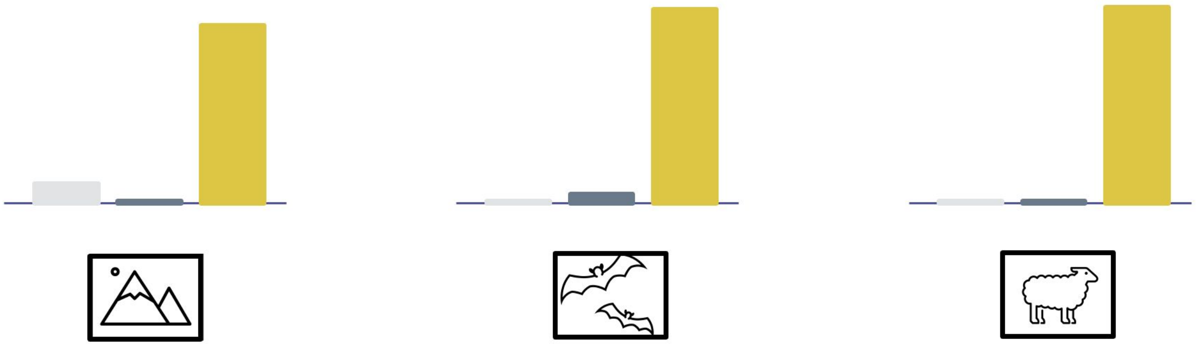
```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    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()
```

Important components for a successful SSL training

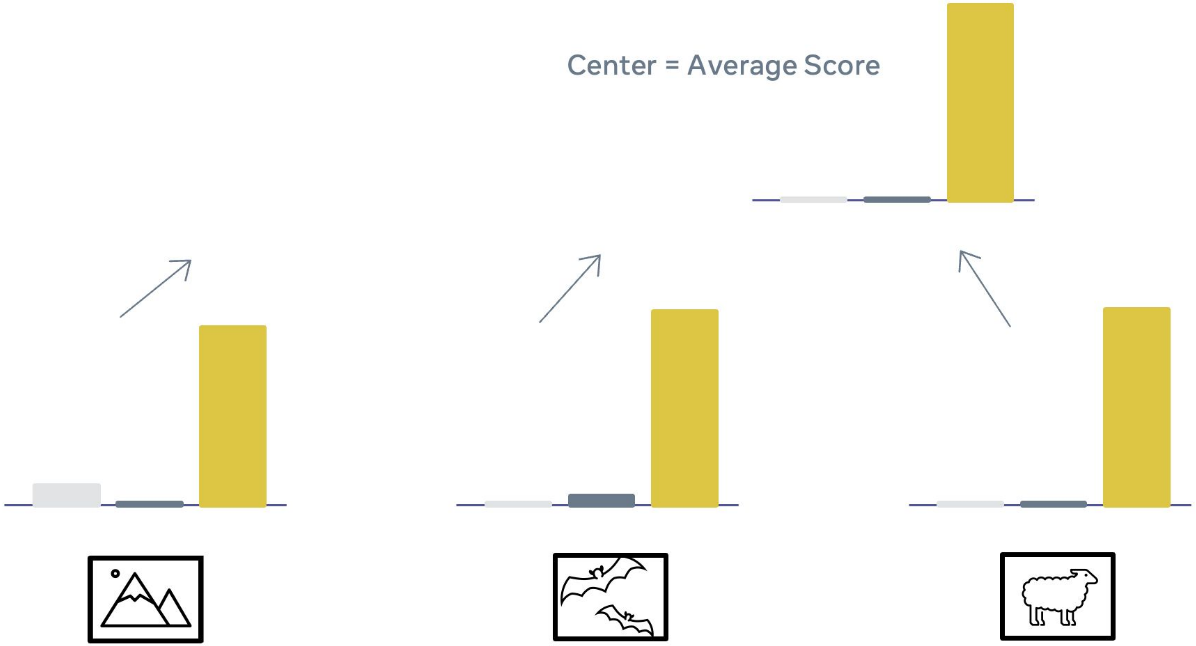
<u>Goal</u>: preventing the model from solving the task in a trivial way

How the model finds trivial ways to solve the SSL	… and how to prevent it.
Collapse all the representations to a constant output.	Centering+sharpening or Sinkhorn-Knopp normalizations

Collapse to constant output

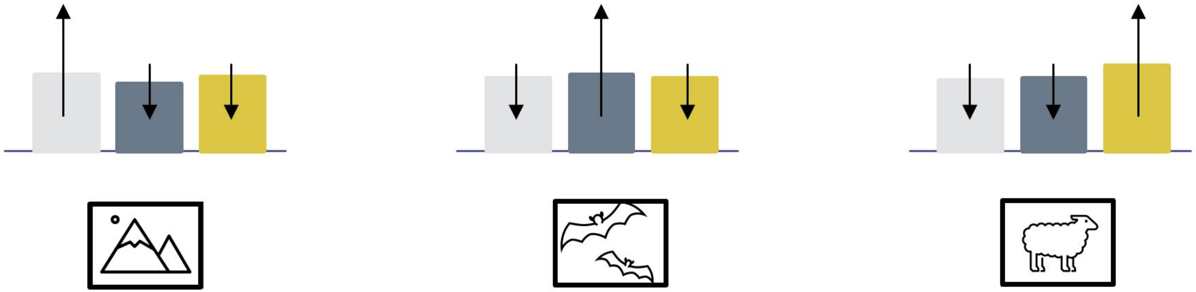


Centering

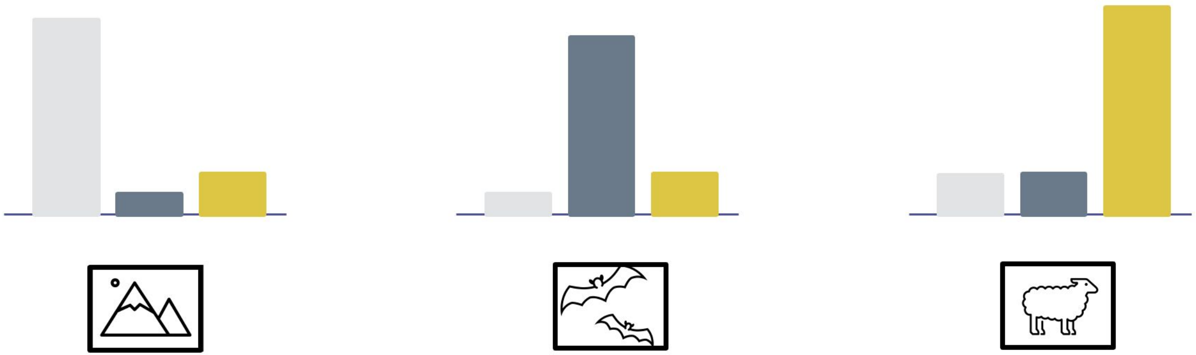


Centering

Centering alone -> it still collapses



Centering + sharpening



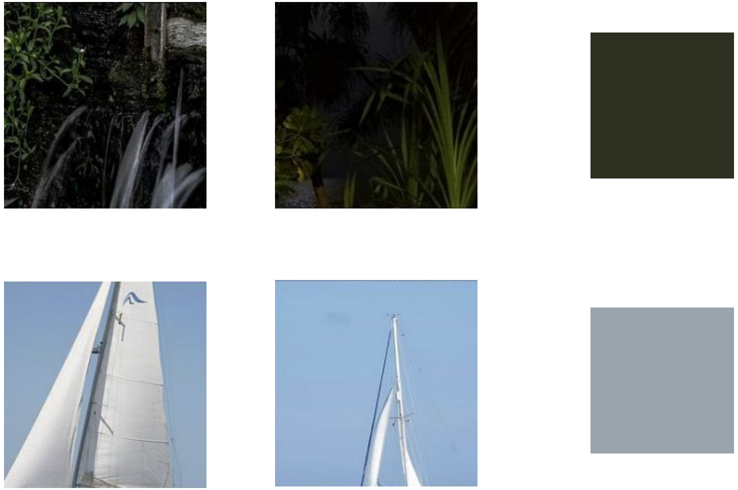
Important components for a successful SSL training

Goal: preventing the model from solving the task in a trivial way

How the model finds trivial ways to solve the SSL	… and how to prevent it.
Collapse all the representations to a constant output.	Centering+sharpening or Sinkhorn-Knopp normalizations
Find similar images based on color statistics	Data augmentation

Data augmentation to prevent solving the task with low-level cues

Two crops from the same image: The model just need to encode color information to predict one from the other.



Data augmentation to prevent solving the task with low-level cues

```
ssl_data_augmentation = transforms.Compose([
            transforms.RandomResizedCrop(224),
            transforms.RandomHorizontalFlip(p=0.5),
            transforms.RandomApply(
            [transforms.ColorJitter(brightness=0.4, contrast=0.4, saturation=0.2, hue=0.1)], p=0.8),
            transforms.RandomGrayscale(p=0.2),
            utils.GaussianBlur(0.1),
            utils.Solarization(0.2),
            normalize,
        ])
```

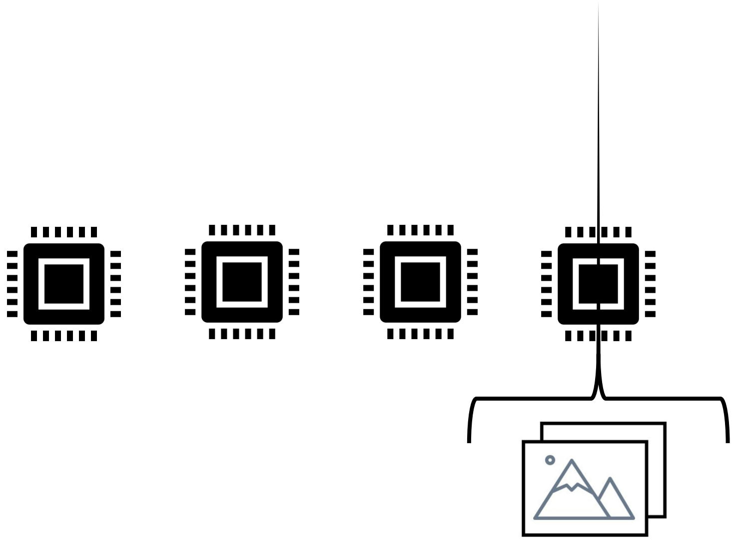
https://github.com/facebookresearch/dino/blob/main/main_dino.py

Important components for a successful SSL training

<u>Goal</u>: preventing the model from solving the task in a trivial way

How the model finds trivial ways to solve the SSL	… and how to prevent it.
Collapse all the representations to a constant output.	Centering+sharpening or Sinkhorn-Knopp normalizations
Find similar images based on color statistics	Data augmentation
Find similar images based on who is located on which hosts/machines	Batch synchronisation

Importance of batch normalization

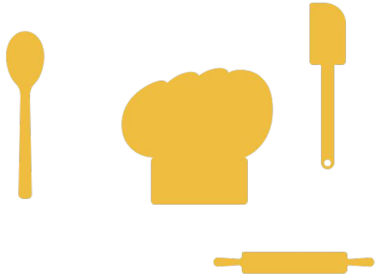


Images located on the same device are closer together because they share the same batch statistics.

A few more recipes of best practices

Mark & Randall

A Cookbook of Self-Supervised Learning



Meta AI

Research

The self-supervised learning cookbook

April 25, 2023

To contribute
send us email

marksibrahim@meta.com
randallbalestriero@gmail.com

Randall Balestriero[*], Mark Ibrahim[*], *Vlad Sobal*[*], *Ari Morcos*[*], *Shashank Shekhar*[*], *Tom Goldstein*[†], *Florian Bordes*[*‡], *Adrien Bardes*[*], *Gregoire Mialon*[*], *Yuandong Tian*[*], *Avi Schwarzschild*[†], *Andrew Gordon Wilson*[**], *Jonas Geiping*[†], *Quentin Garrido*[*§], *Pierre Fernandez*[**], *Amir Bar*[*], *Hamed Pirsiavash*[+], *Yann LeCun*[*] and *Micah Goldblum*[**]

\+ Special thanks to Ishan[*] and Mathilde[***]

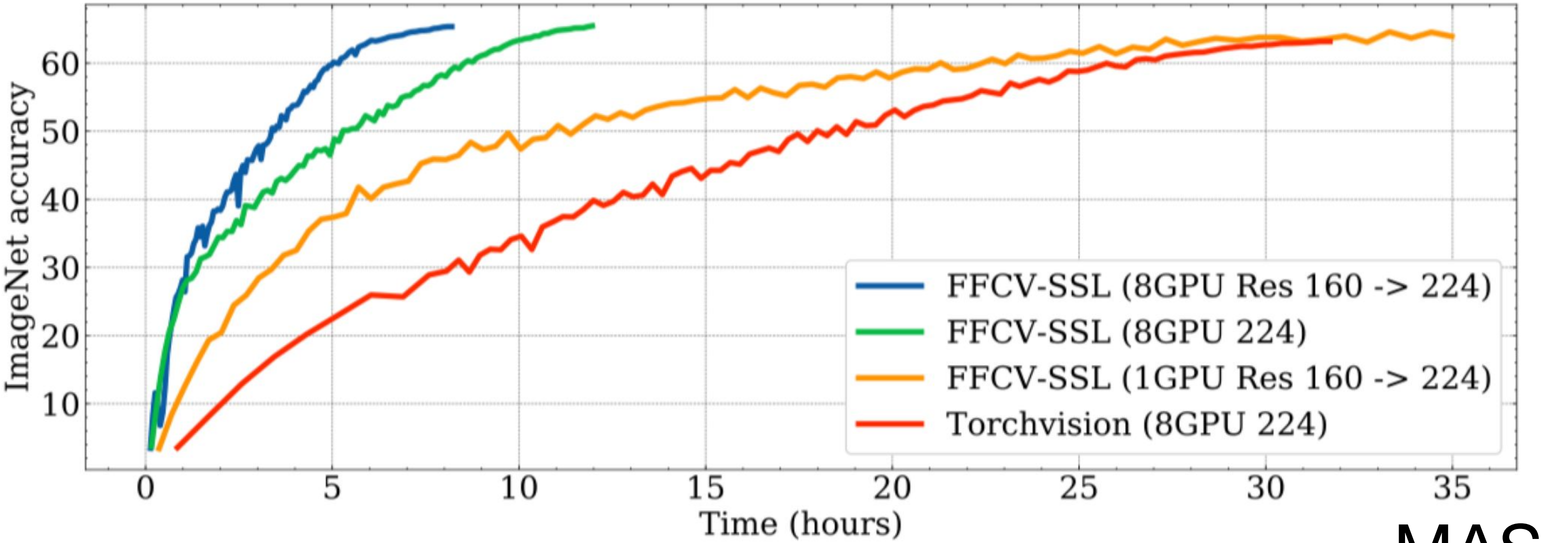
[*]Meta AI, FAIR
[**]New York University
[†]University of Maryland
[+]University of California, Davis
[‡]Universite de Montreal, Mila
[§]Univ Gustave Eiffel, CNRS, LIGM
[⋆]Univ. Rennes, Inria, CNRS, IRISA
[italic]Equal contributions, randomized ordering

[***]Google Research

arXiv > cs > arXiv:2304.12210

https://arxiv.org/abs/2304.12210

Speeding up your training



MASKING

FFCV-SSL Public

Fast Forward Computer Vision for Self-Supervised Learning

Speeding up your training

PyTorch

TORCH.TENSOR.BFLOAT16

Tensor.bfloat16(*memory_format=torch.preserve_format*) → Tensor

`self.bfloat16()` is equivalent to `self.to(torch.bfloat16)`. See `to()`.

```
# Enables autocasting for the forward
with autocast():
    output = model(input)
    loss = loss_fn(output, target)
```

solo-learn

Distributed Training Gotchas

#1 Sync your batchnorm

```
model = torch.nn.SyncBatchNorm.convert_sync_batchnorm(model)
```

Distributed Training Gotchas

#2 Gather & Reduce

Algorithm 1:

```python
class GatherLayer(torch.autograd.Function):
    """
    Gather tensors from all process and support backward propagation
    for the gradients across processes.
    """

    @staticmethod
    def forward(ctx, x):
        output = [torch.zeros_like(x) for _ in range(dist.get_world_size())]
        dist.all_gather(output, x)
        return tuple(output)

    @staticmethod
    def backward(ctx, *grads):
        all_gradients = torch.stack(grads)
        dist.all_reduce(all_gradients)
        return all_gradients[dist.get_rank()]
```

```python
# Forward
torch.distributed.all_gather(output, x)

# Backward
torch.distributed.all_reduce(all_gradients)
```

Other considerations

CNNs or ViTs?

Project size?

SSL for unbalanced data

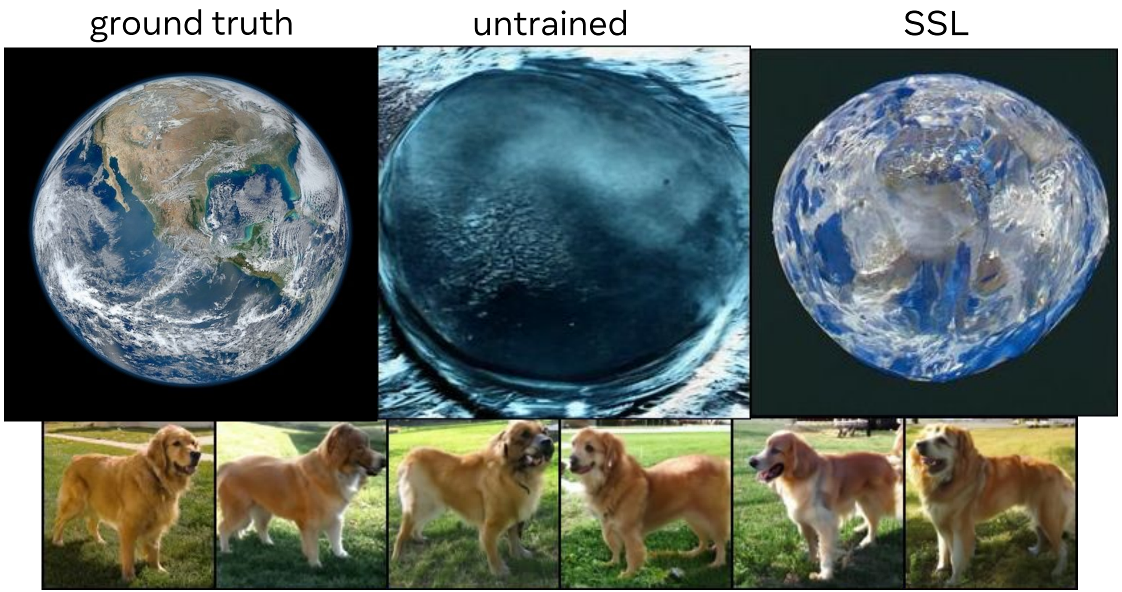
Standard hyperparameters

Extending SSL to other modalities

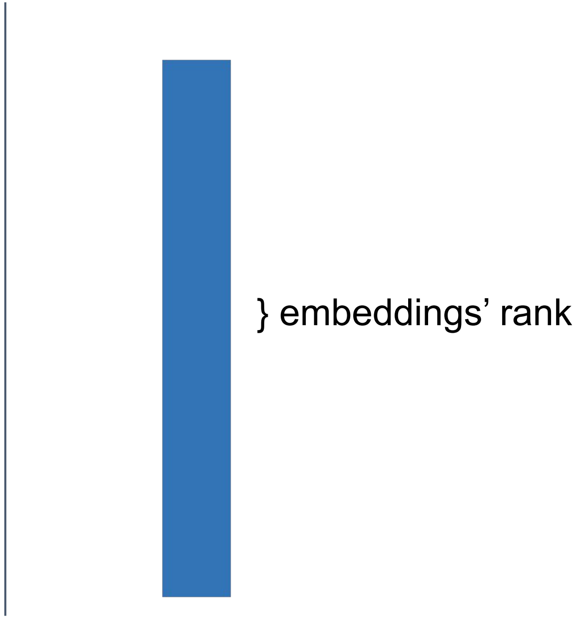
…

Evaluation without labels

RCDM



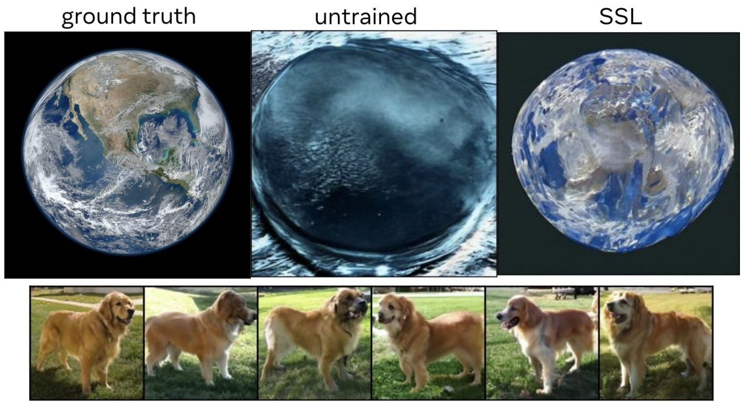
RankMe



RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank. Garrido et al. 2022
High Fidelity Visualization of What Your Self-Supervised Representation Knows About. Bordes et al. 2022

Evaluation without labels

RCDM



Florian Bordes

https://www.linkedin.com/in/florianbordes/
florian.bordes@umontreal.ca

RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank. Garrido et al. 2022
High Fidelity Visualization of What Your Self-Supervised Representation Knows About. Bordes et al. 2022

Evaluation without labels

Quentin Garrido

Poster

RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank

Quentin Garrido · Randall Balestriero · Laurent Najman · Yann LeCun

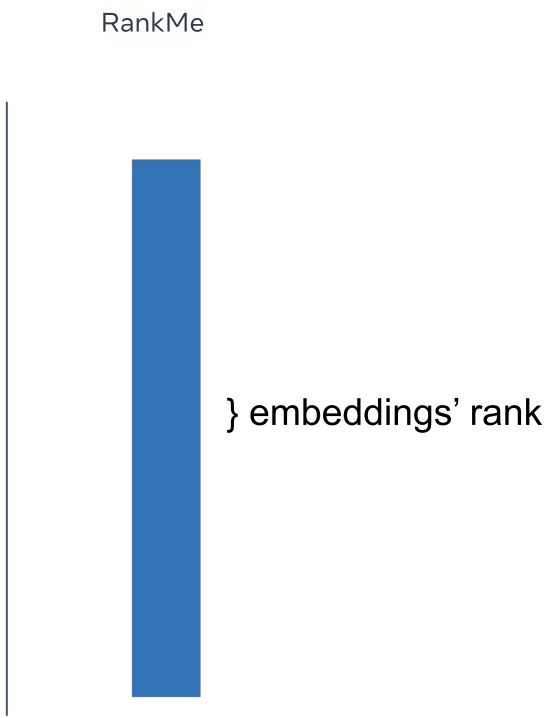
Exhibit Hall 1 #609

[Abstract]

[Poster]

Thu 27 Jul 7:30 p.m. EDT — 9 p.m. EDT (Bookmark)

Oral presentation: Oral B5 Self/Semi-Supervised Learning and Interpretability / Observing Aspects of NN
Wed 26 Jul 10 p.m. EDT — 11:30 p.m. EDT (Bookmark)



RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank. Garrido et al. 2022
High Fidelity Visualization of What Your Self-Supervised Representation Knows About. Bordes et al. 2022

A Cookbook of Self-Supervised Learning

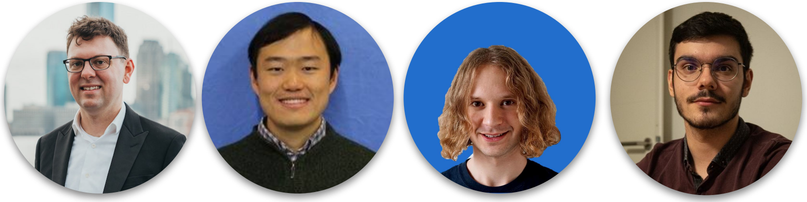
Meta AI

Research

The self-supervised learning cookbook

April 25, 2023

@ICML in person



Andrew Yuandong Vlad Quentin

Ishan Ari Mark Tom Randall

arXiv > cs > arXiv:2304.12210

https://arxiv.org/abs/2304.12210