Foundation models

Summer School on GenAI 2024

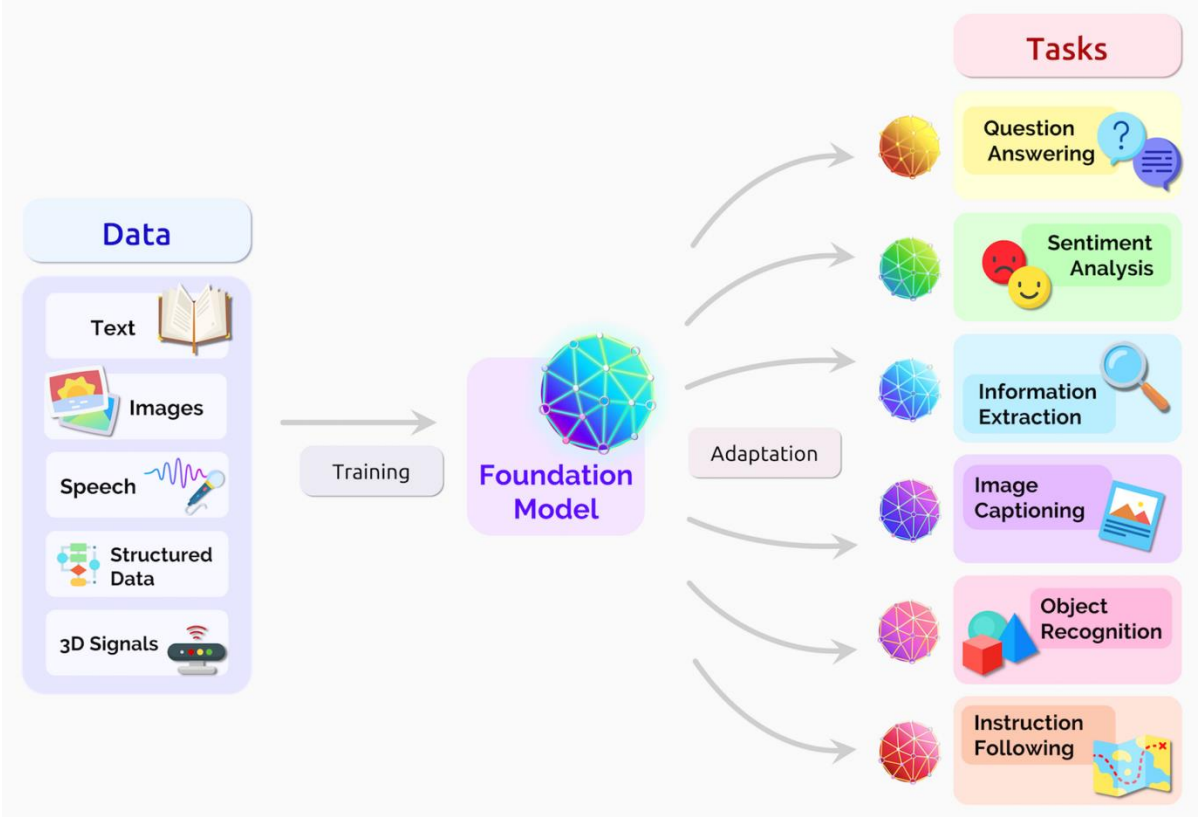
Outline

- Foundation models and Self-supervised learning
- Reconstruct from a corrupted (or partial) version
- Visual common sense tasks
- Contrastive Learning
- Feature Prediction
- Vision-language Foundation models

Outline

- Foundation models and Self-supervised learning
- Reconstruct from a corrupted (or partial) version
- Visual common sense tasks
- Contrastive Learning
- Feature Prediction
- Vision-language Foundation models

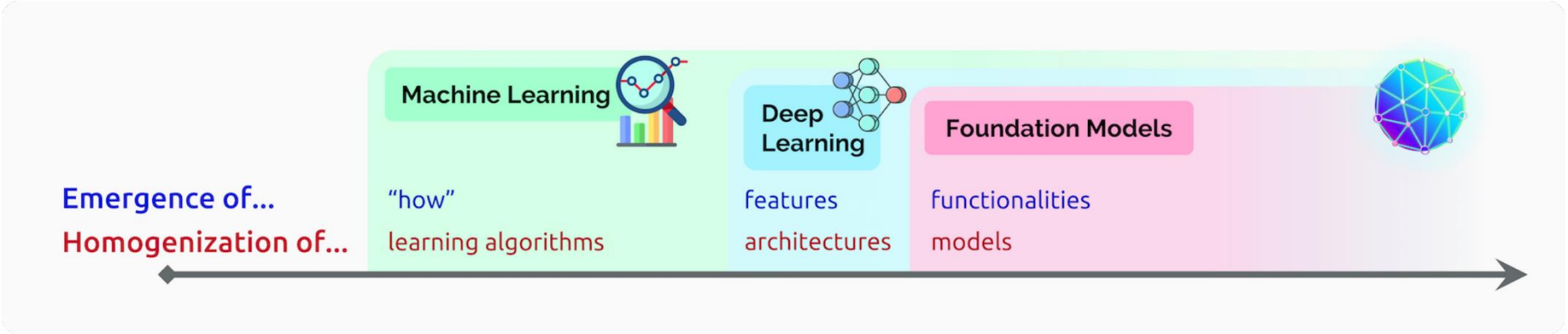
What is foundation models?



Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021)

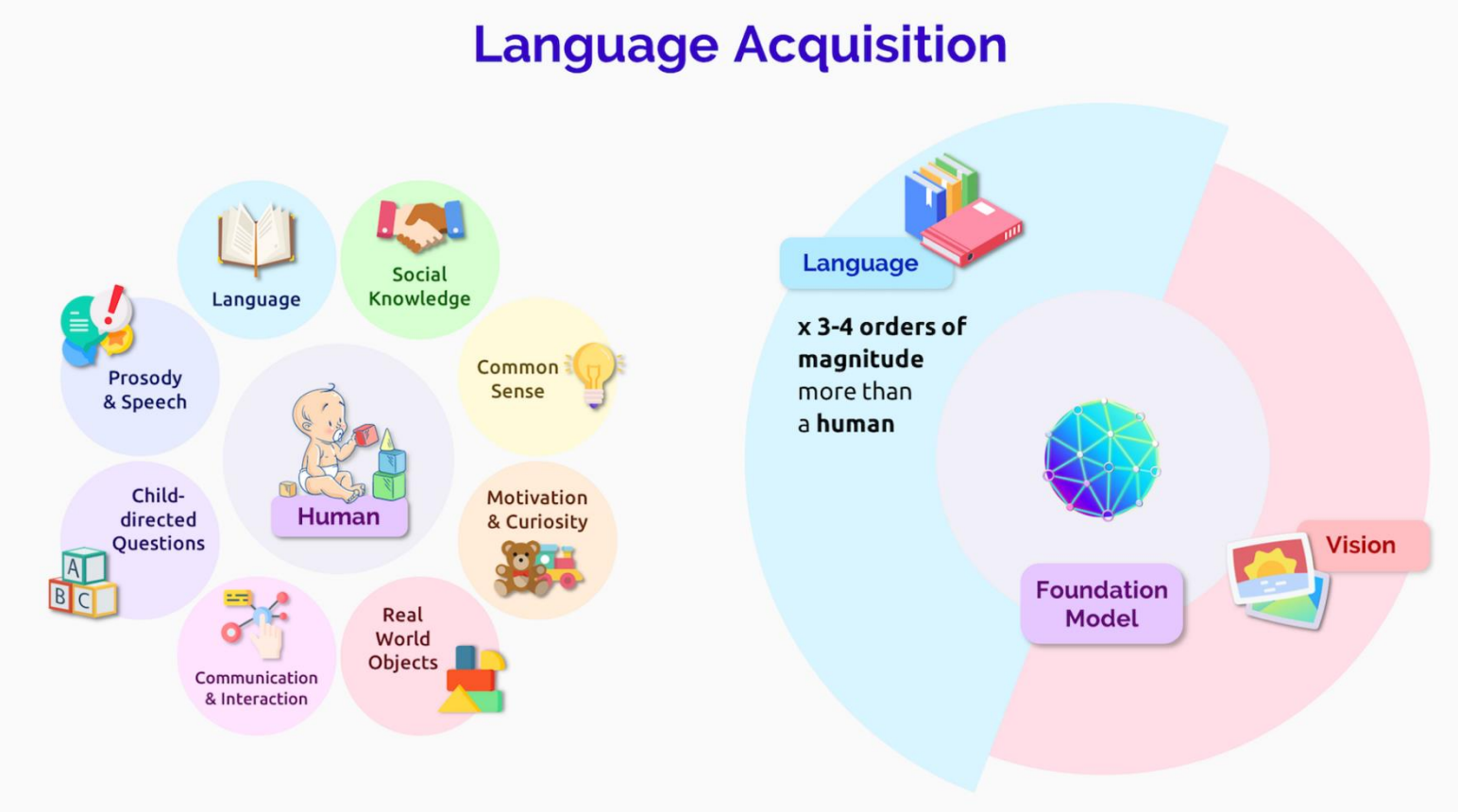
Why foundation models?

- Traditional programming: Input + Program = Output
- Machine learning: Input + Output = Program
- DL and FM



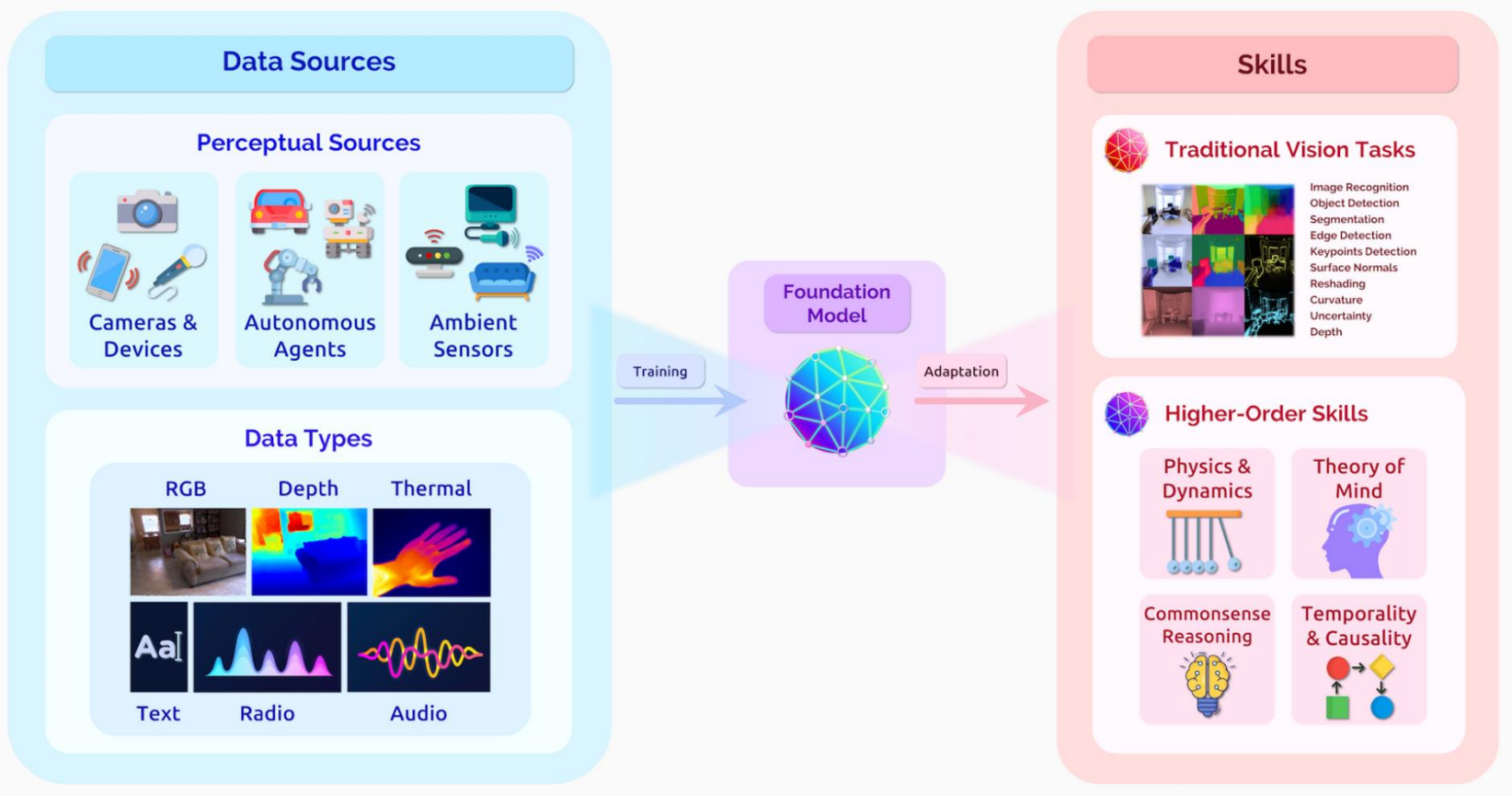
Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021)

Large Language Models



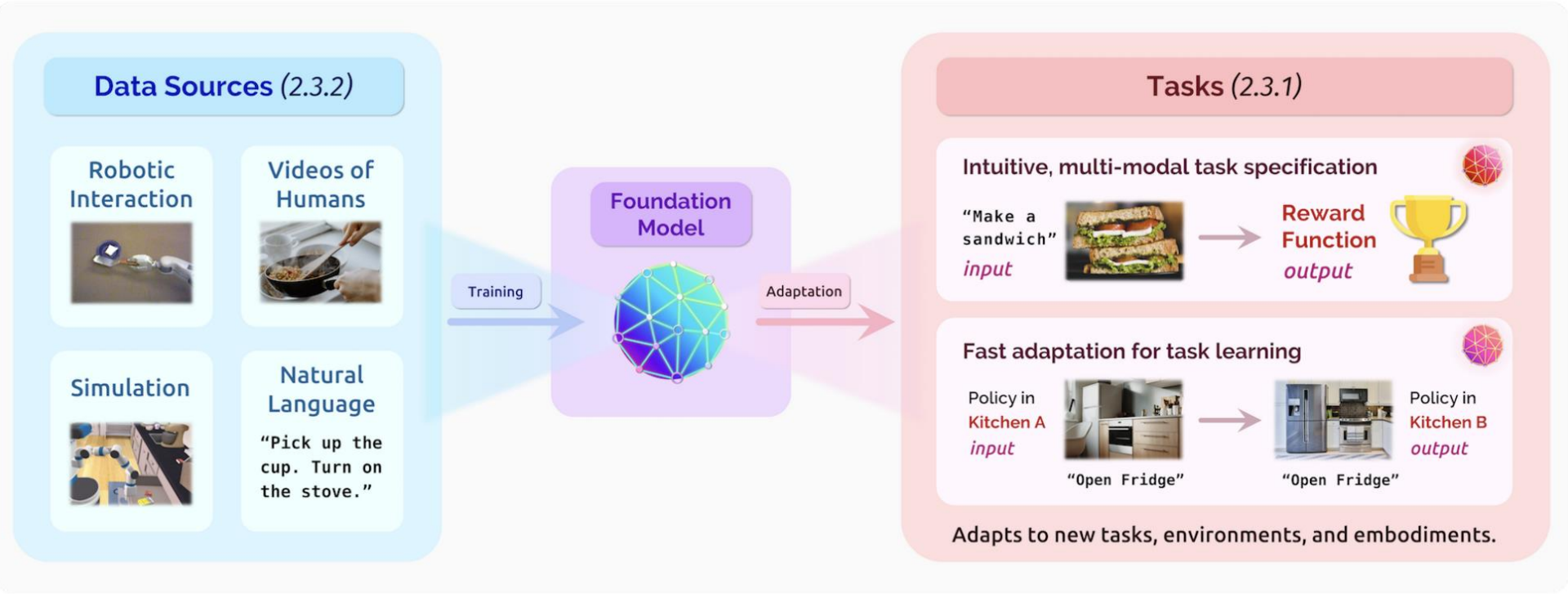
Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021)

Visual Foundation Models



Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021)

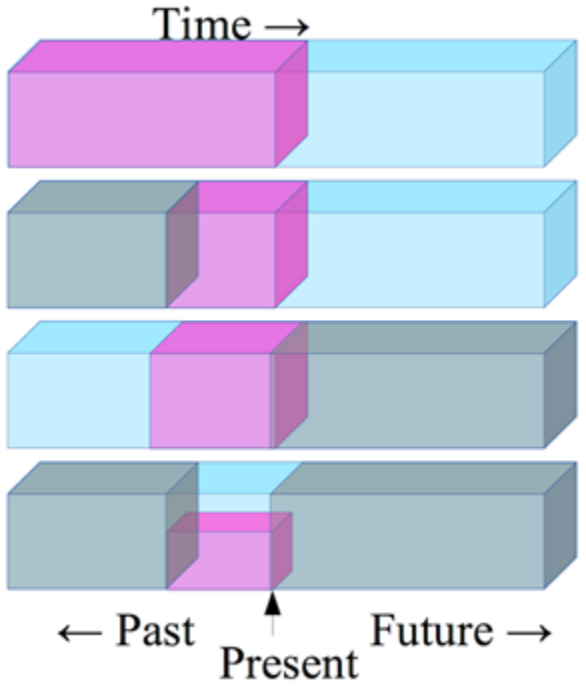
Foundation Models for Robotics



Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021)

What is Self-Supervised Learning?

- A type of **unsupervised learning** where data provides the supervision
  - Predict any part of the input from any other part.
  - Predict the future from the past.
  - Predict the future from the recent past.
  - Predict the past from the present.
  - Predict the top from the bottom.
  - Predict the occluded from the visible
  - Pretend there is a part of the input you don't know and predict that.



Motivation: LeCake

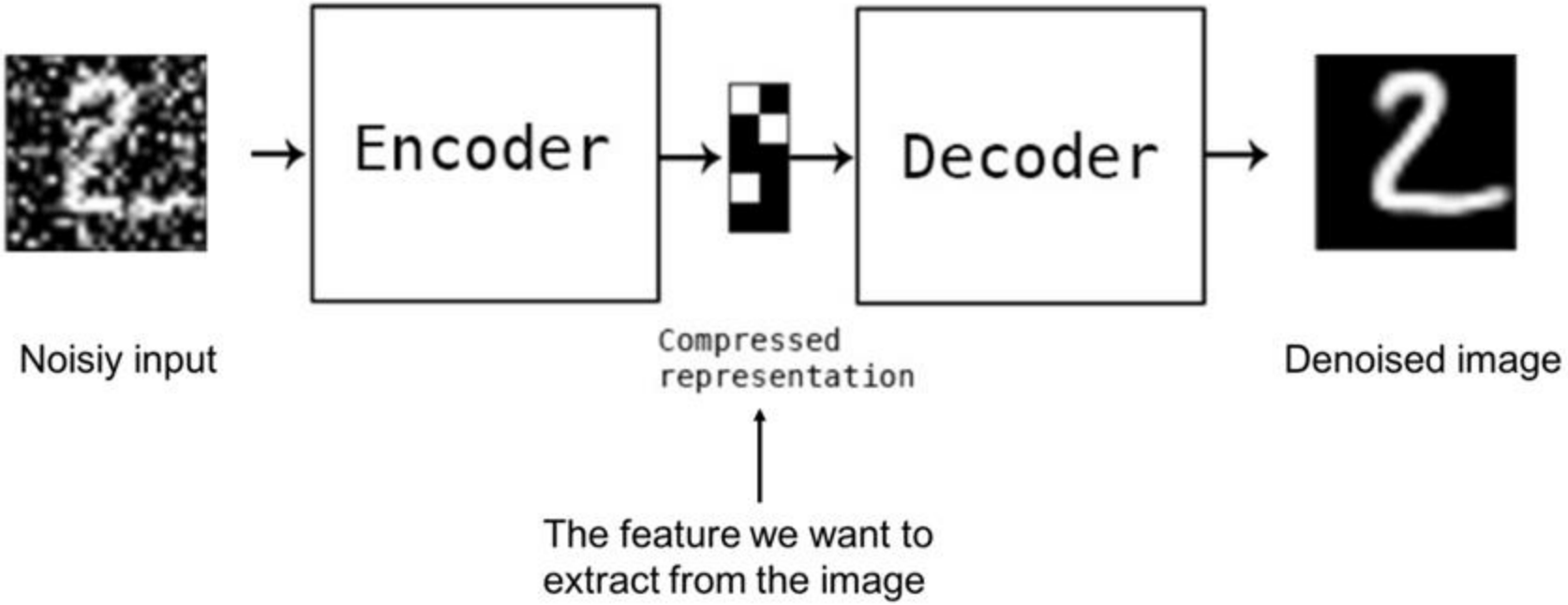
- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Yann LeCun's cake

Outline

- Foundation models and Self-supervised learning
- Reconstruct from a corrupted (or partial) version
- Visual common sense tasks
- Contrastive Learning
- Feature Prediction
- Vision-language Foundation models

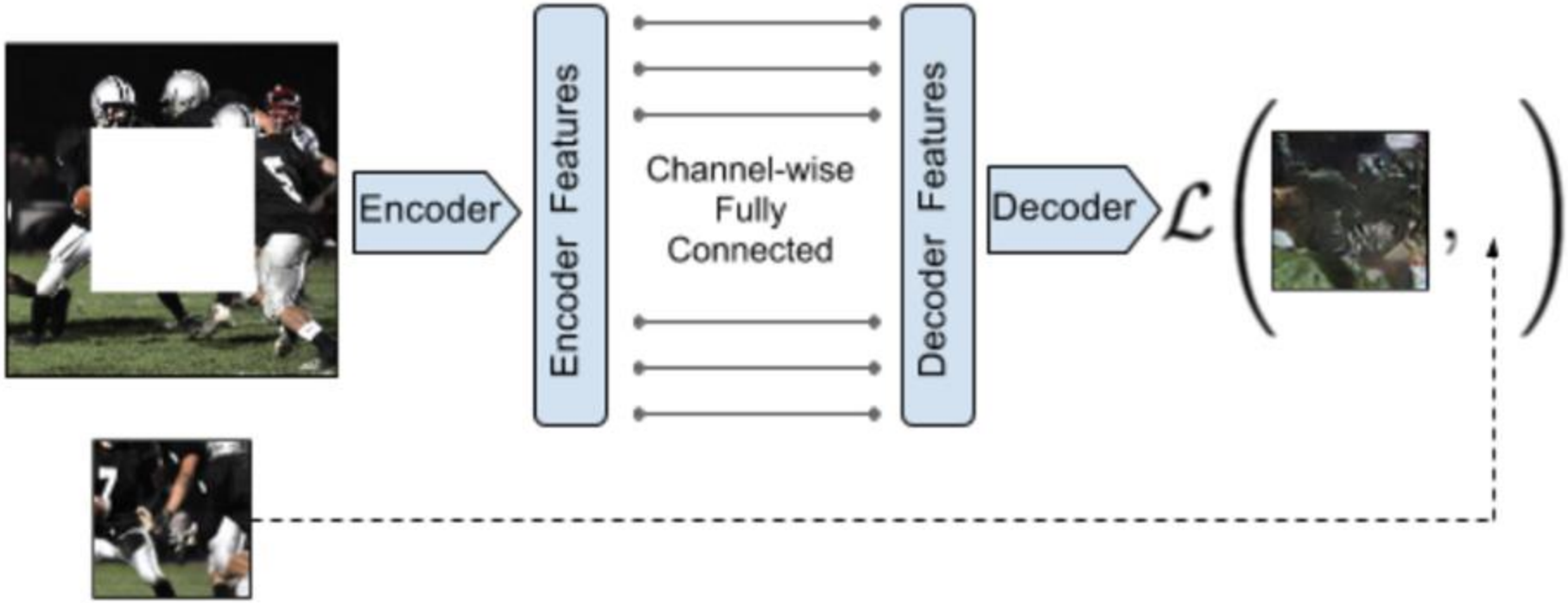
Denoising Autoencoder



Predict missing pieces

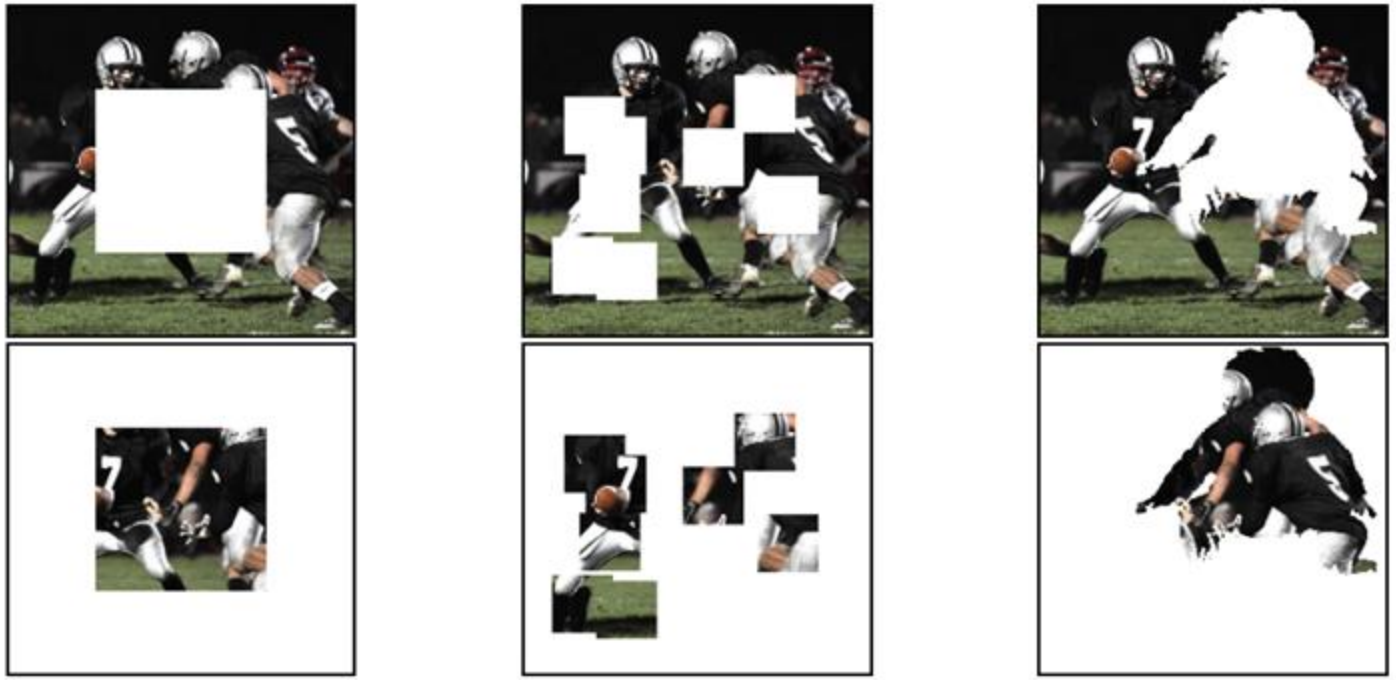
Pathak et al 2016

Context Encoders



Pathak et al 2016

Context Encoders



(a) Central region (b) Random block (c) Random region

Pathak et al 2016

Context Encoders

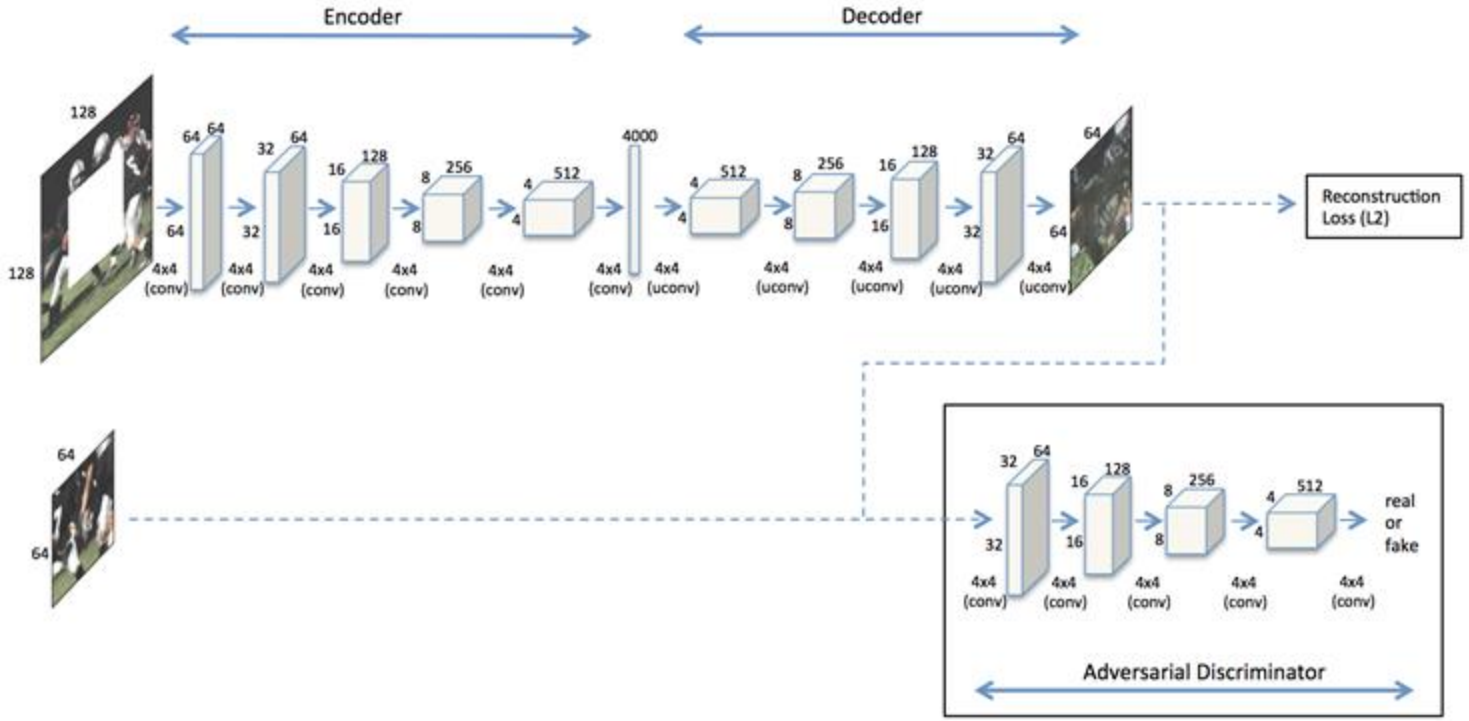
$$\mathcal{L}_{rec}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2$$

$$\mathcal{L}_{adv} = \max_D \quad \mathbb{E}_{x \in \mathcal{X}}[\log(D(x)) + \log(1 - D(F((1 - \hat{M}) \odot x)))]$$

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv}$$

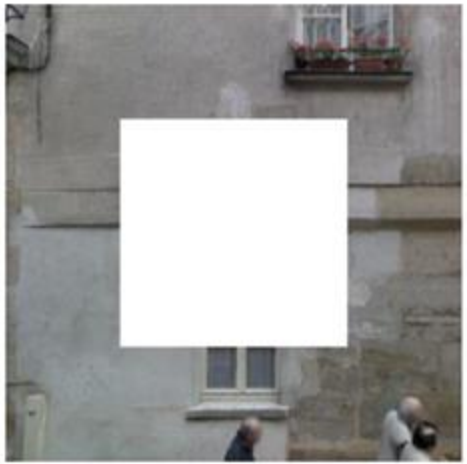
Pathak et al 2016

Context Encoders



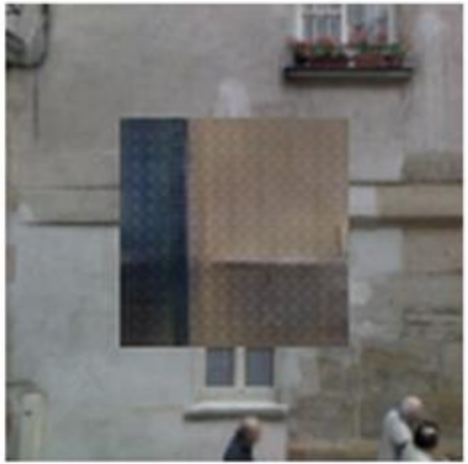
Pathak et al 2016

Context Encoders



Input Image

L2 Loss

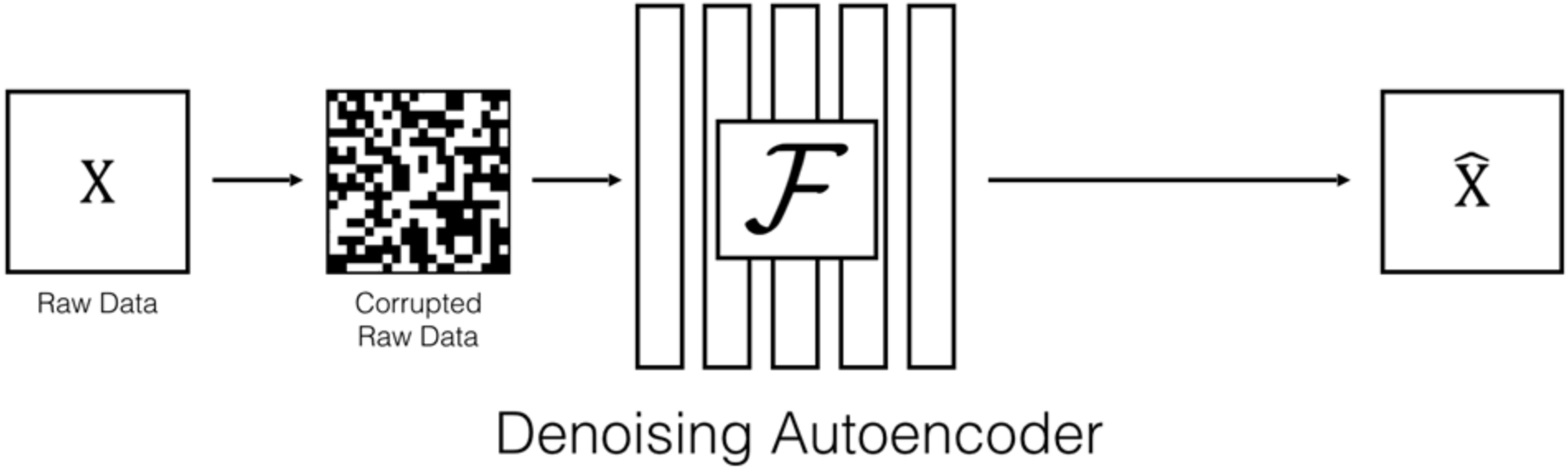


Adversarial Loss

Joint Loss

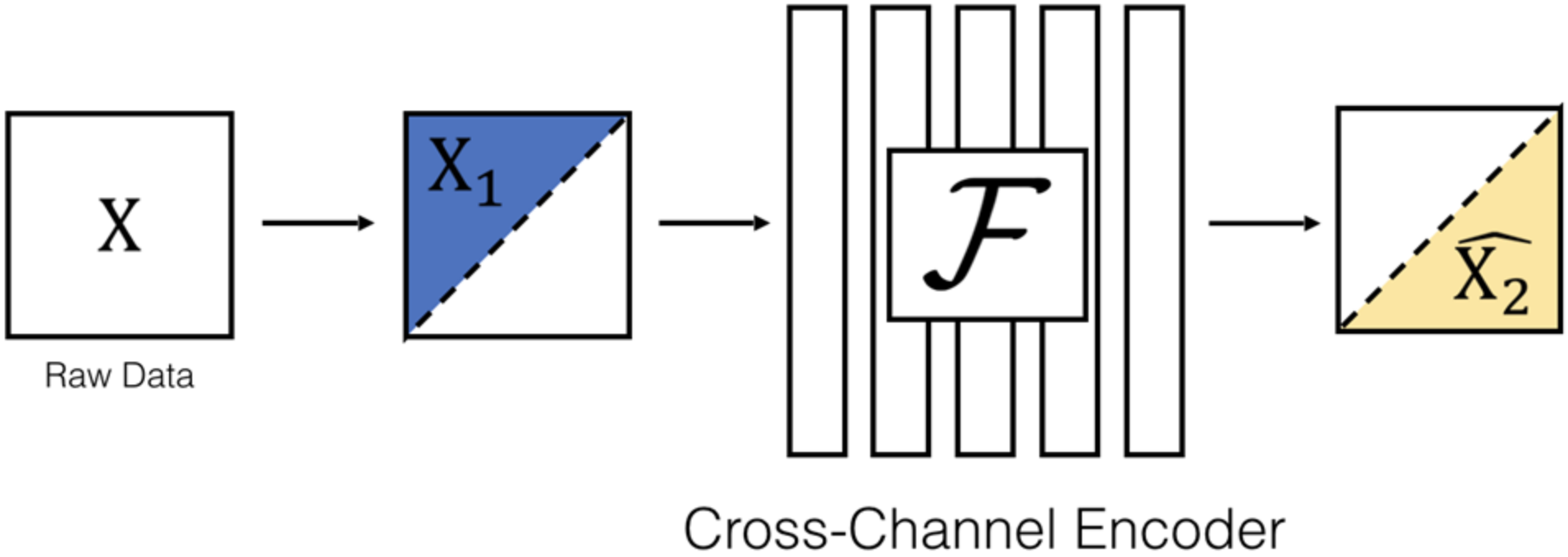
Pathak et al 2016

Predicting one view from another



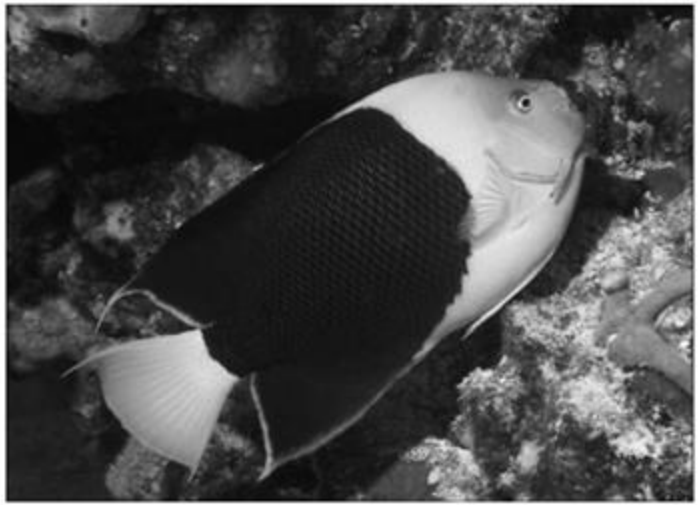
Slide: Richard Zhang

Predicting one view from another



Slide: Richard Zhang

Predicting one view from another



$\mathcal{F}$

Grayscale image: *L* channel

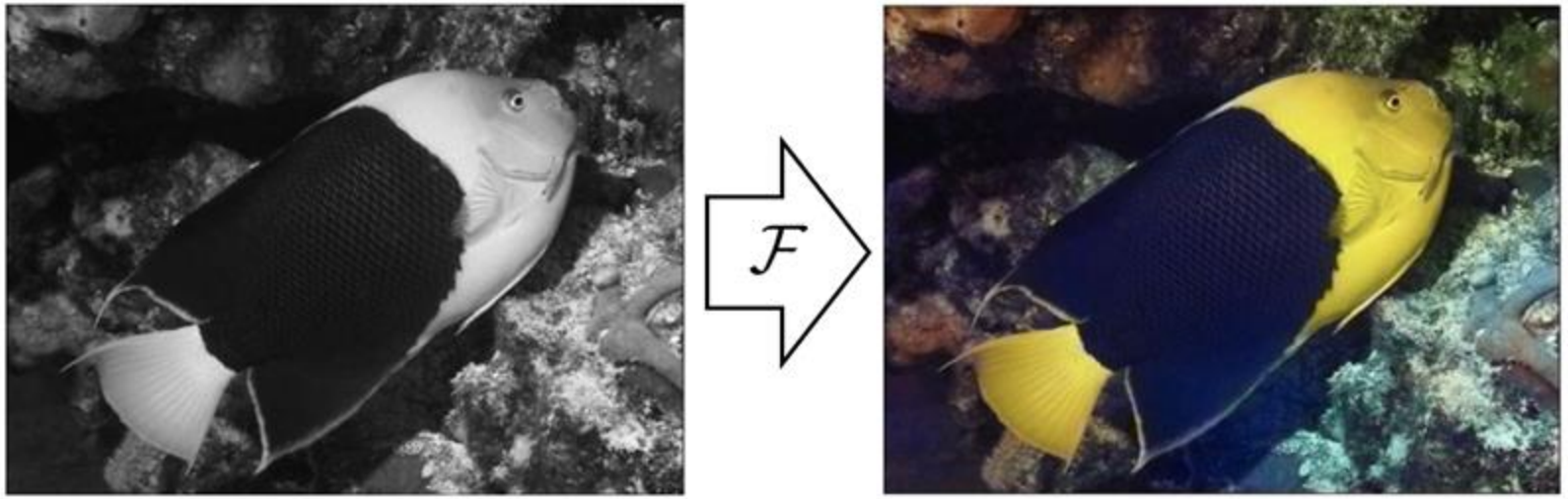
$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

Color information: *ab* channels

$\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$

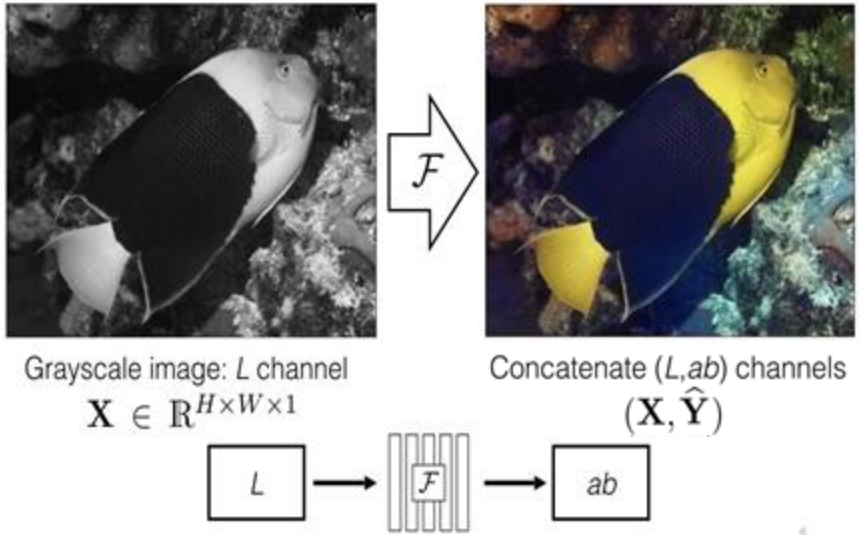
Slide: Richard Zhang

Predicting one view from another



Slide: Richard Zhang

Predicting one view from another

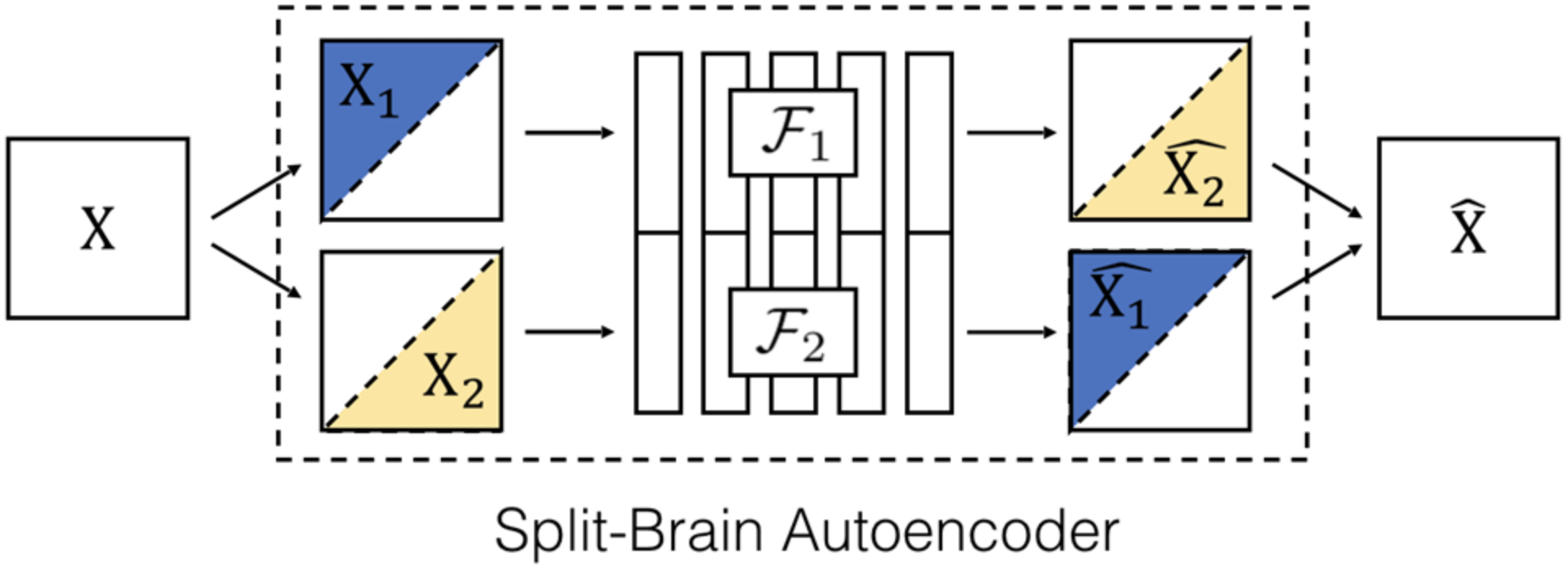


$$\mathrm{L}_2(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \hat{\mathbf{Y}}_{h,w}\|_2^2$$

$$\mathrm{L}(\hat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW} \sum_{h,w} \sum_{q} \mathbf{Z}_{h,w,q} \log(\hat{\mathbf{Z}}_{h,w,q})$$

Slide: Richard Zhang

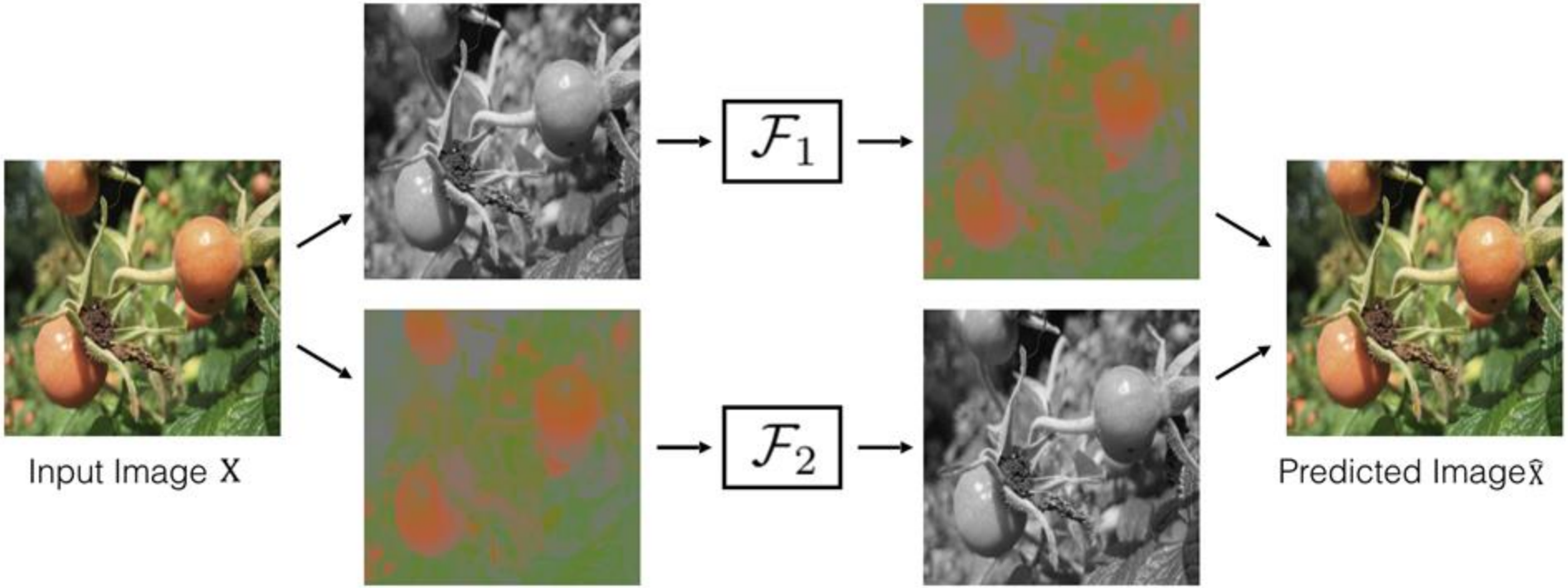
Predicting one view from another



Split-Brain Autoencoder

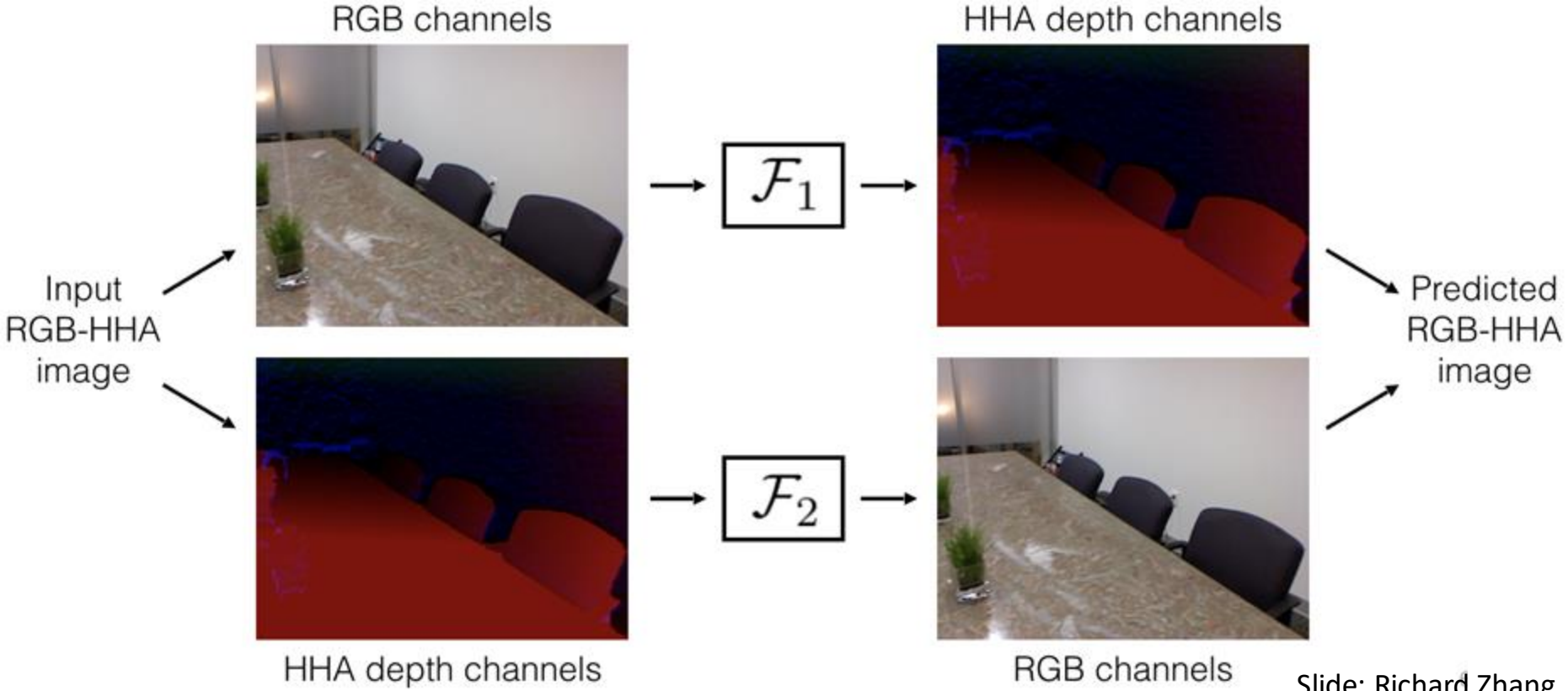
Slide: Richard Zhang

Predicting one view from another



Slide: Richard Zhang

Predicting one view from another



Slide: Richard Zhang

MAE

Masked Autoencoders Are Scalable Vision Learners

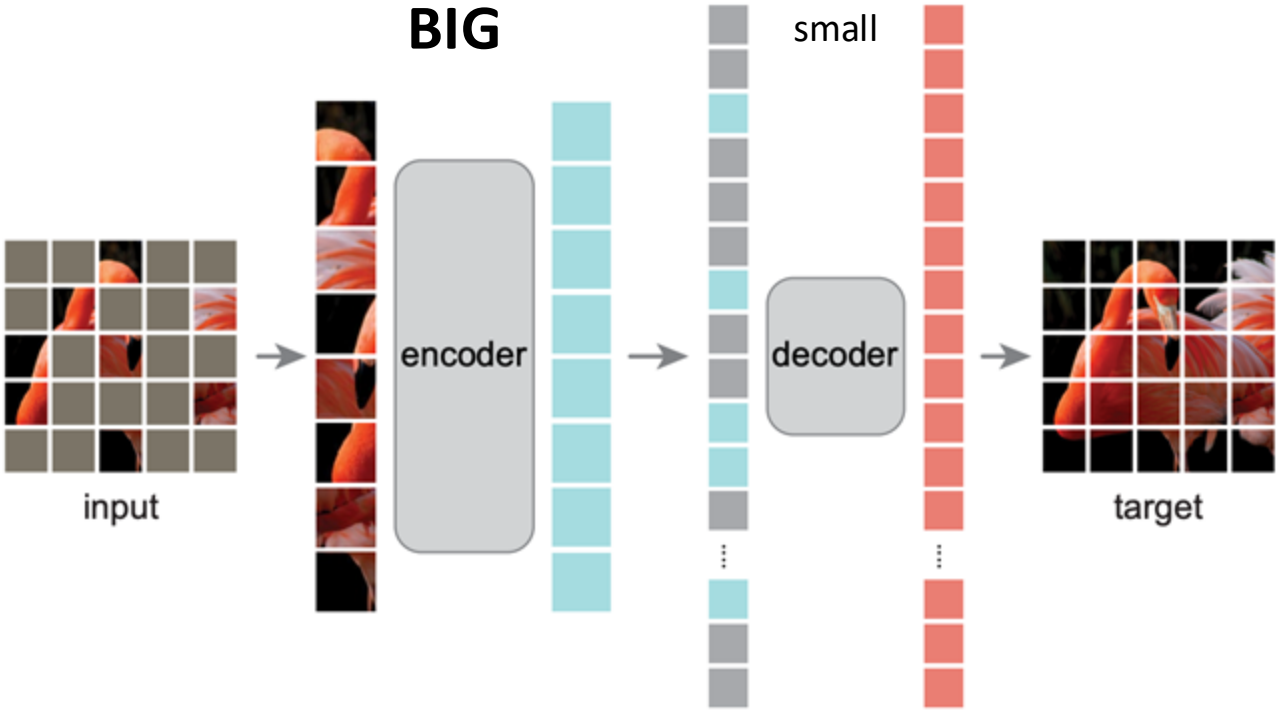
Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)

Nov, 2021

MAE



Architecture: Vision Transformer (ViT)

MAE on ImageNet validation images

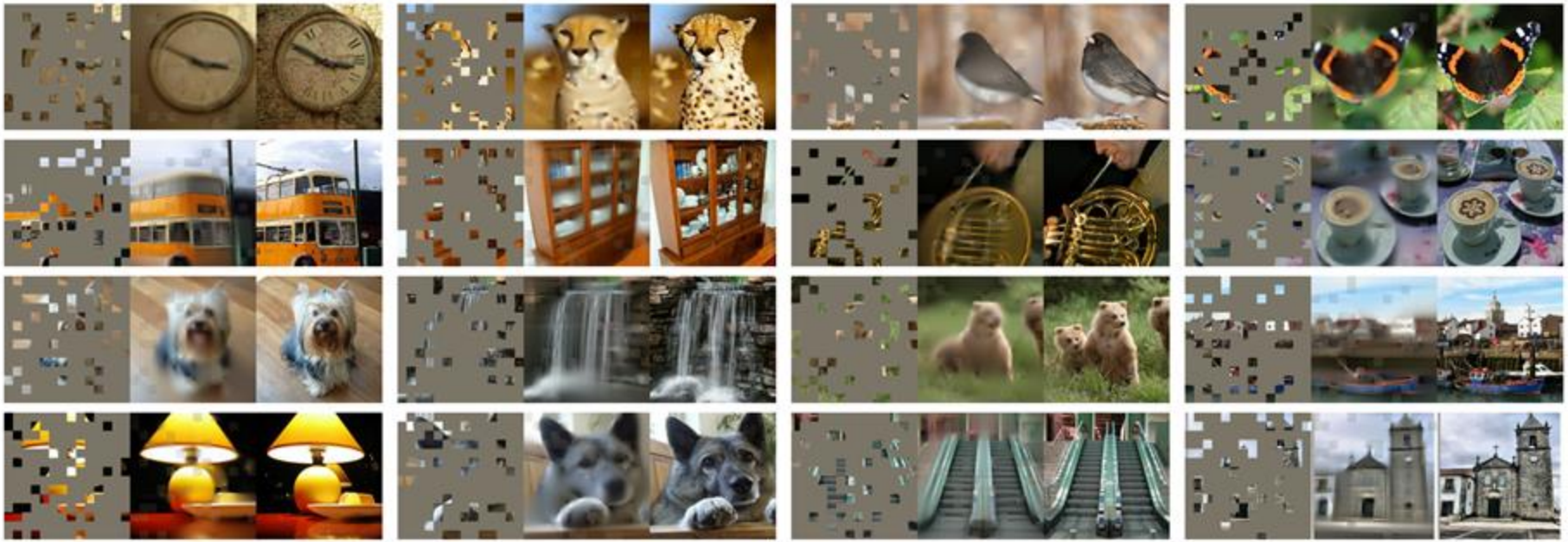


Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction† (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix.

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

MAE on CoCo validation images

Figure 3. Example results on COCO validation images, using an MAE trained on ImageNet (the same model weights as in Figure 2). Observe the reconstructions on the two right-most examples, which, although different from the ground truth, are semantically plausible.

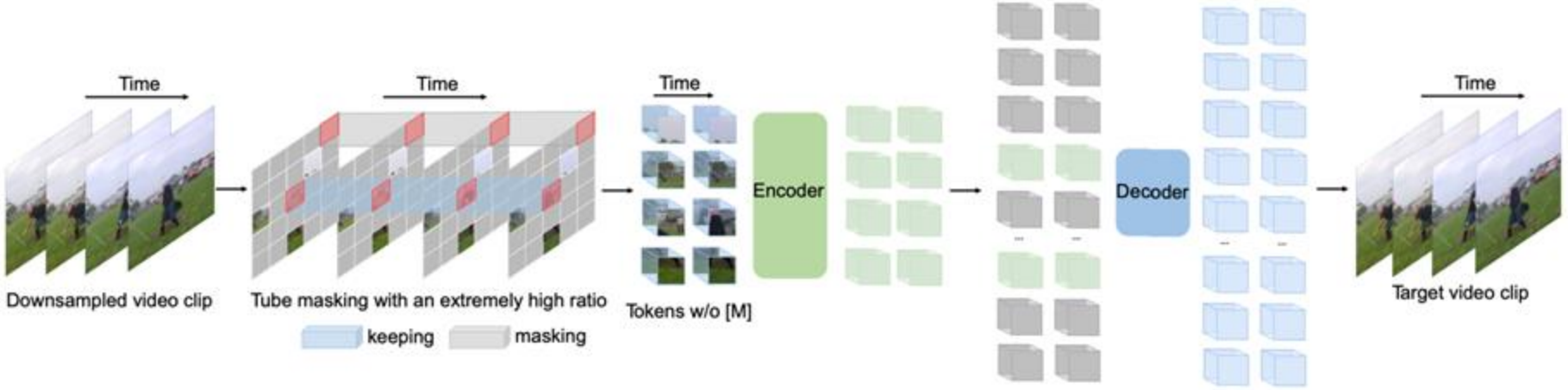
VideoMAE

VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training

Zhan Tong [1,2*] **Yibing Song** [2] **Jue Wang** [2] **Limin Wang** [1,3†]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Tencent AI Lab [3]Shanghai AI Lab
tongzhan@smail.nju.edu.cn {yibingsong.cv, arphid}@gmail.com lmwang@nju.edu.cn

[Oct, 2022]

VideoMAE Architecture



Experiments on Something-Something V2

Method	Backbone	Extra data	Ex. labels	Frames	GFLOPs	Param	Top-1	Top-5
TEINet$_{En}$ [40]	ResNet50$_{\times 2}$	ImageNet-1K	✓	8+16	99×10×3	50	66.5	N/A
TANet$_{En}$ [41]	ResNet50$_{\times 2}$		✓	8+16	99×2×3	51	66.0	90.1
TDN$_{En}$ [75]	ResNet101$_{\times 2}$		✓	8+16	198×1×3	88	69.6	92.2
SlowFast [23]	ResNet101	Kinetics-400	✓	8+32	106×1×3	53	63.1	87.6
MViTv1 [22]	MViTv1-B		✓	64	455×1×3	37	67.7	90.9
TimeSformer [6]	ViT-B	ImageNet-21K	✓	8	196×1×3	121	59.5	N/A
TimeSformer [6]	ViT-L		✓	64	5549×1×3	430	62.4	N/A
ViViT FE [3]	ViT-L	IN-21K+K400	✓	32	995×4×3	N/A	65.9	89.9
Motionformer [51]	ViT-B		✓	16	370×1×3	109	66.5	90.1
Motionformer [51]	ViT-L		✓	32	1185×1×3	382	68.1	91.2
Video Swin [39]	Swin-B		✓	32	321×1×3	88	69.6	92.7
VIMPAC [65]	ViT-L	HowTo100M+DALLE	✗	10	N/A×10×3	307	68.1	N/A
BEVT [77]	Swin-B	IN-1K+K400+DALLE	✗	32	321×1×3	88	70.6	N/A
MaskFeat↑$_{312}$ [80]	MViT-L	Kinetics-600	✓	40	2828×1×3	218	75.0	95.0
VideoMAE	ViT-B	Kinetics-400	✗	16	180×2×3	87	69.7	92.3
VideoMAE	ViT-L	Kinetics-400	✗	16	597×2×3	305	74.0	94.6
VideoMAE	ViT-S	*no external data*	✗	16	57×2×3	22	66.8	90.3
VideoMAE	ViT-B		✗	16	180×2×3	87	70.8	92.4
VideoMAE	ViT-L		✗	16	597×2×3	305	74.3	94.6
VideoMAE	ViT-L		✗	32	1436×1×3	305	**75.4**	**95.2**

Table 6: **Comparison with the state-of-the-art methods on Something-Something V2**. Our VideoMAE reconstructs normalized cube pixels and is pre-trained with a masking ratio of 90% for 2400 epochs. "Ex. labels ✗" means only *unlabelled* data is used during the pre-training phase. "N/A" indicates the numbers are not available for us.

Audio-MAE

Masked Autoencoders that Listen

Po-Yao Huang[1] Hu Xu[1] Juncheng Li[2] Alexei Baevski[1]
Michael Auli[1] Wojciech Galuba[1] Florian Metze[1] Christoph Feichtenhofer[1]

[1]Meta AI [2]Carnegie Mellon University

Audio-MAE: Architecture

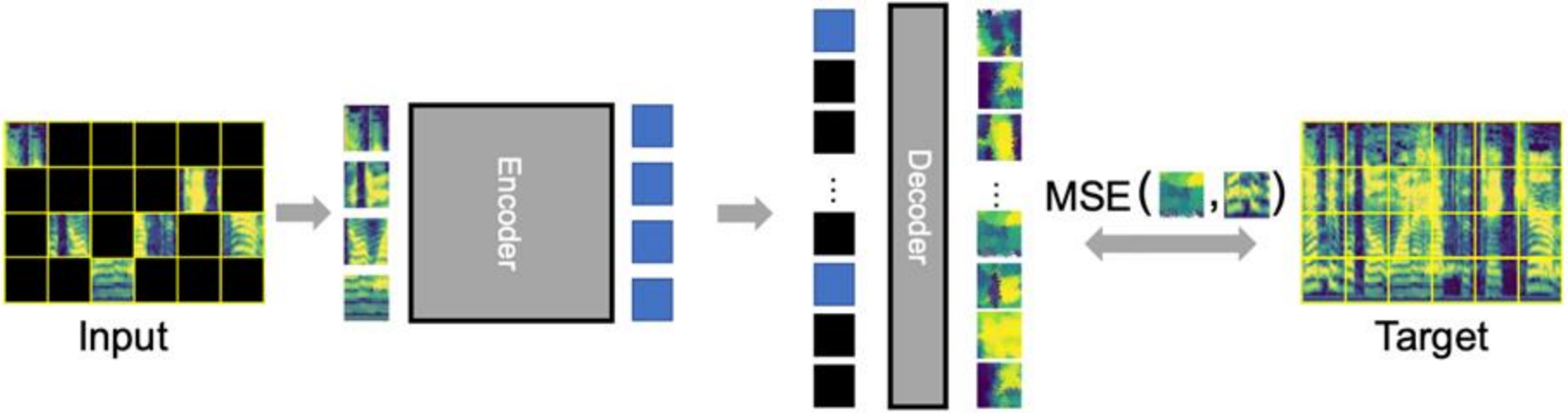


Figure 1: **Audio-MAE for audio self-supervised learning**. An audio recording is first transformed into a spectrogram and split into patches. We embed patches and mask out a large subset (80%). An encoder then operates on the visible (20%) patch embeddings. Finally, a decoder processes the order-restored embeddings and mask tokens to reconstruct the input. Audio-MAE is minimizing the mean square error (MSE) on the masked portion of the reconstruction and the input spectrogram.

Model	Backbone	PT-Data	AS-20K	AS-2M	ESC-50	SPC-2	SPC-1	SID
No pre-training								
ERANN [58]	CNN	-	-	45.0	89.2	-	-	-
PANN [59]	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±.11	46.8±.06	93.6±.11	**98.3**±.06	**97.6**±.06	94.1±.06
Audio-MAE (local)	ViT-B	AS	**37.0**±.11	**47.3**±.11	**94.1**±.10	**98.3**±.06	96.9±.00	**94.8**±.11
Out-of-domain supervised pre-training								
PSLA [30]	EffNet [60]	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†	98.0	-	-
PaSST [28]	DeiT-B	IN	-	47.1	96.8†	-	-	-

Table 2: **Comparison with other state-of-the-art models** on audio and speech classification tasks. Metrics are mAP for AS and accuracy (%) for ESC/SPC/SID. For pre-training (PT) dataset, AS:AudioSet, LS:LibriSpeech, and IN:ImageNet. †: Fine-tuning results with additional supervised training on AS-2M. We gray-out models pre-trained with external non-audio datasets (*e.g.*, ImageNet). Best single models in AS-2M are compared (no ensembles). *: linear evaluation results from [53].

MultiMAE

MultiMAE: Multi-modal Multi-task Masked Autoencoders

Roman Bachmann* David Mizrahi* Andrei Atanov Amir Zamir
Swiss Federal Institute of Technology Lausanne (EPFL)

https://multimae.epfl.ch

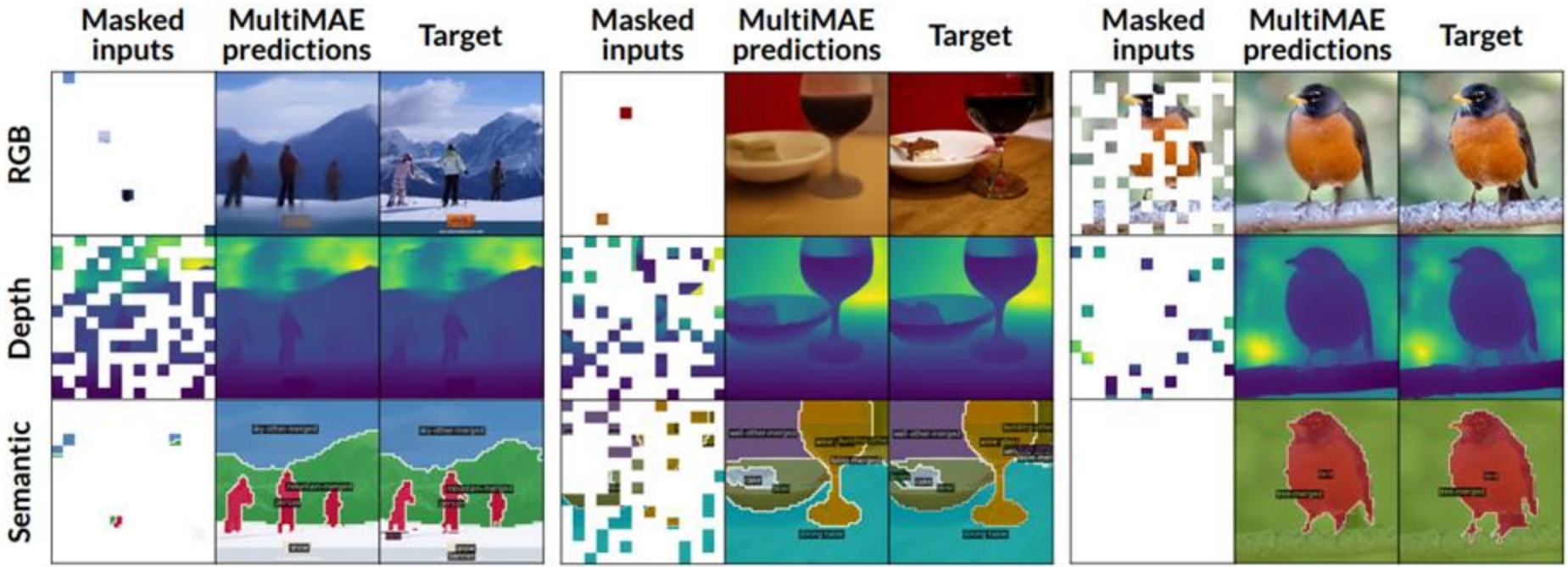


Figure 1. **MultiMAE pre-training objective.** We randomly select 1/6 of all 16×16 image patches from multiple modalities and learn to reconstruct the remaining 5/6 masked patches from them. The figure shows validation examples from ImageNet, where masked inputs (left), predictions (middle), and non-masked images (right) for RGB (top), depth (middle), and semantic segmentation (bottom) are provided. Since we do not compute a loss on non-masked patches, we overlay the input patches on the predictions.

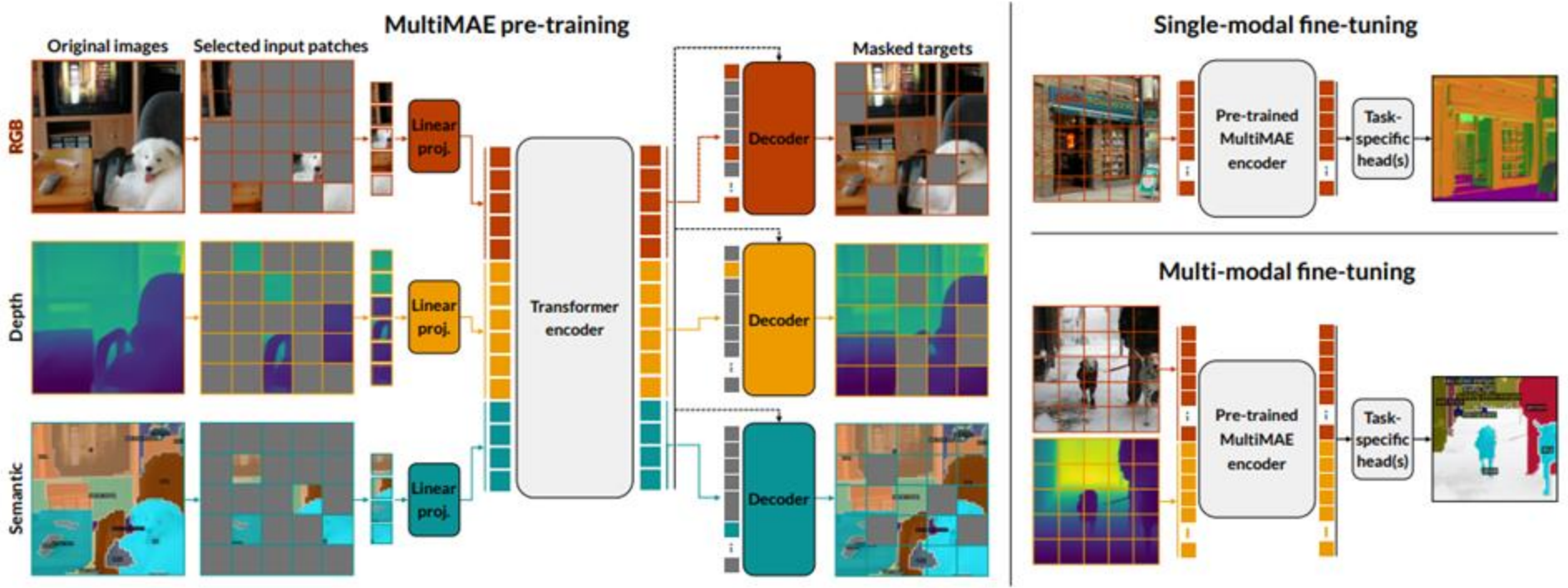


Figure 2. **(Left) MultiMAE pre-training**: A small subset of randomly sampled patches from multiple modalities (e.g., RGB, depth, and semantic segmentation) is linearly projected to tokens with a fixed dimension and encoded using a Transformer. Task-specific decoders reconstruct the masked-out patches by first performing a cross-attention step from queries to the encoded tokens, followed by a shallow Transformer. The queries consist of mask tokens (in gray), with the task-specific encoded tokens added at their respective positions. **(Right) Fine-tuning**: By pre-training on multiple modalities, MultiMAE lends itself to fine-tuning on single-modal and multi-modal downstream tasks. No masking is performed at transfer time.

MultiMAE Experiments

Method	IN-1K (C)	ADE20K (S)	Hypersim (S)	NYUv2 (S)	NYUv2 (D)
Supervised [81]	81.8	45.8	33.9	50.1	80.7
DINO [12]	83.1	44.6	32.5	47.9	81.3
MoCo-v3 [17]	82.8	43.7	31.7	46.6	80.9
MAE [35]	**83.3**	**46.2**	<u>36.5</u>	<u>50.8</u>	<u>85.1</u>
MultiMAE	**83.3**	**46.2**	**37.0**	**52.0**	**86.4**

Table 1. **Fine-tuning with RGB-only.** We report the top-1 accuracy (↑) on ImageNet-1K (IN-1K) [23] classification (C), mIoU (↑) on ADE20K [102] , Hypersim [68] , and NYUv2 [73] semantic segmentation (S), as well as δ_1 accuracy (↑) on NYUv2 depth (D). Text in **bold** and <u>underline</u> indicates the first and second-best results, respectively. All methods are pre-trained on ImageNet-1K (with pseudo labels for MultiMAE).

MultiMAE Experiments

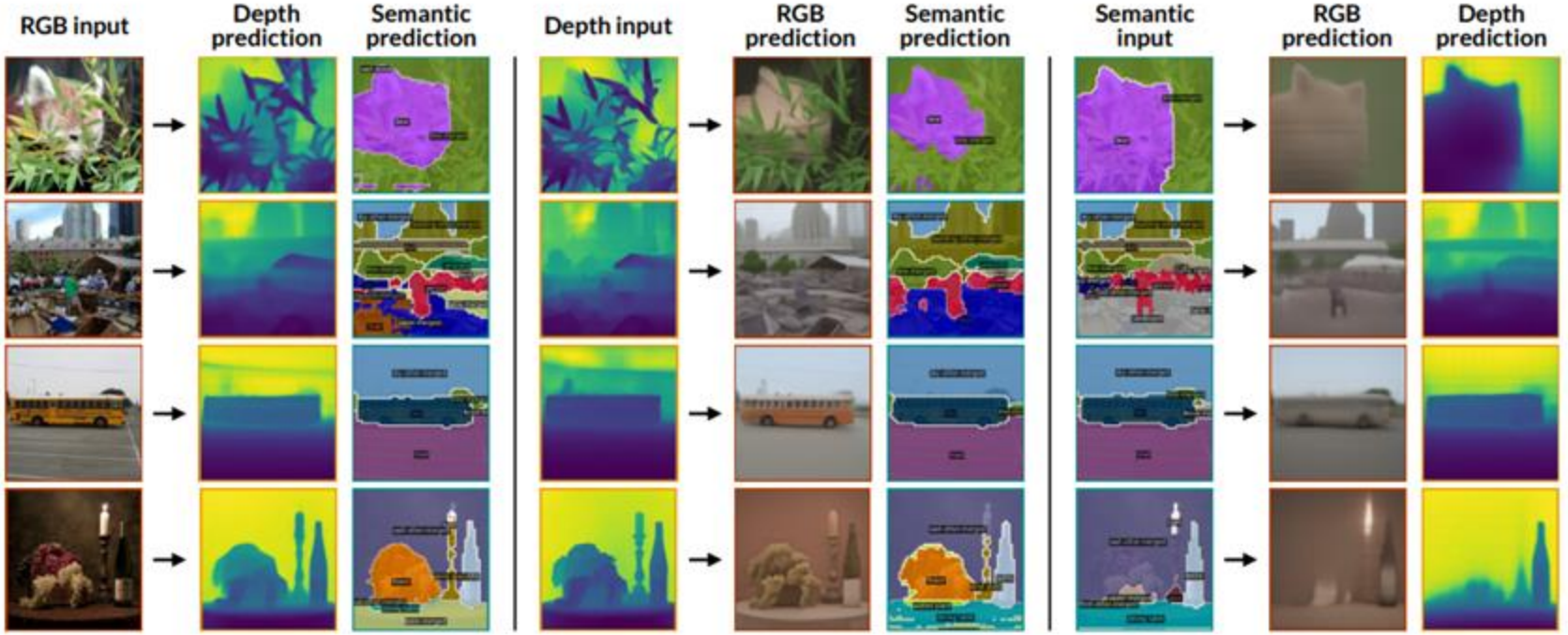


Figure 4. **Single-modal predictions.** We visualize MultiMAE cross-modal predictions on ImageNet-1K validation images. Only a single, full modality is used as input. The predictions remain plausible despite the absence of input patches from other modalities.

M3AE: MultiModal MAE

Multimodal Masked Autoencoders Learn Transferable Representations

Xinyang Geng[1 *] **Hao Liu**[1 2 * †] **Lisa Lee**[2]
Dale Schuurmans[2] **Sergey Levine**[1] **Pieter Abbeel**[1]
[1]UC Berkeley [2]Google Research, Brain Team
[*] Equal contribution. [†] Project lead.
{young.geng, hao.liu}@berkeley.edu

[Oct 2022]

M3AE: Architecture

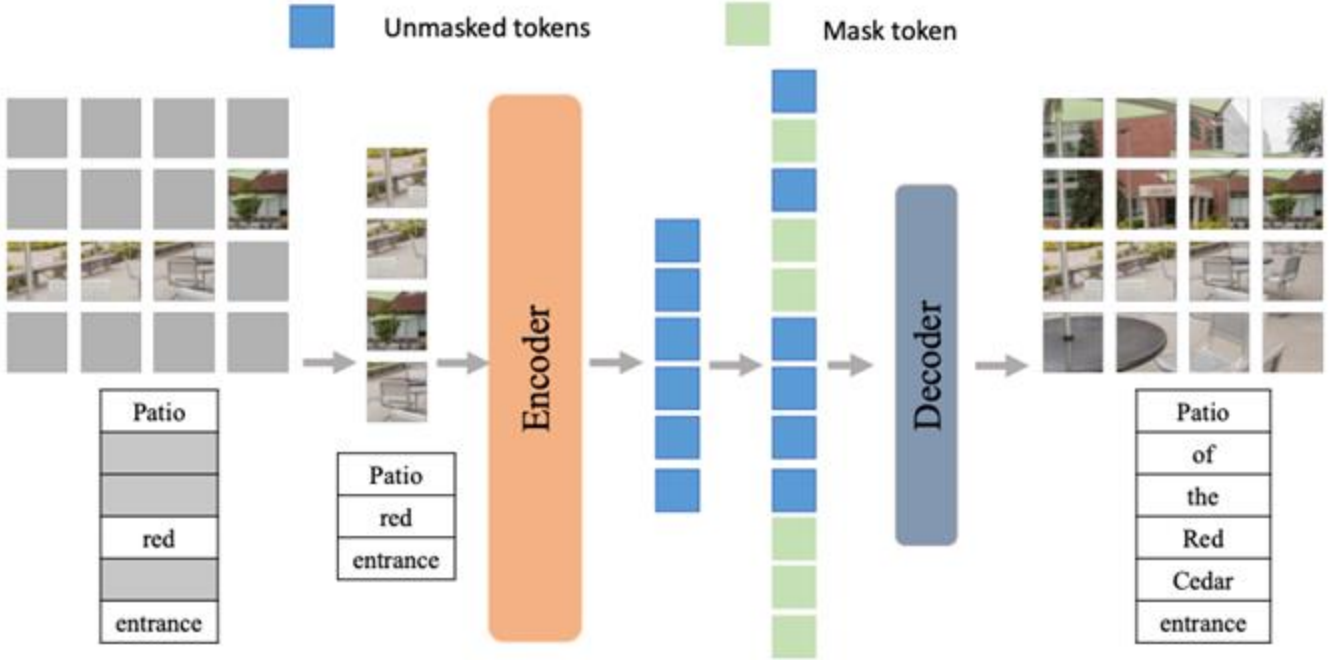


Figure 1: Multimodal masked autoencoder (M3AE) consists of an encoder that maps language tokens and image patches to a shared representation space, and a decoder that reconstructs the original image and language from the representation.

Comparison with MAE

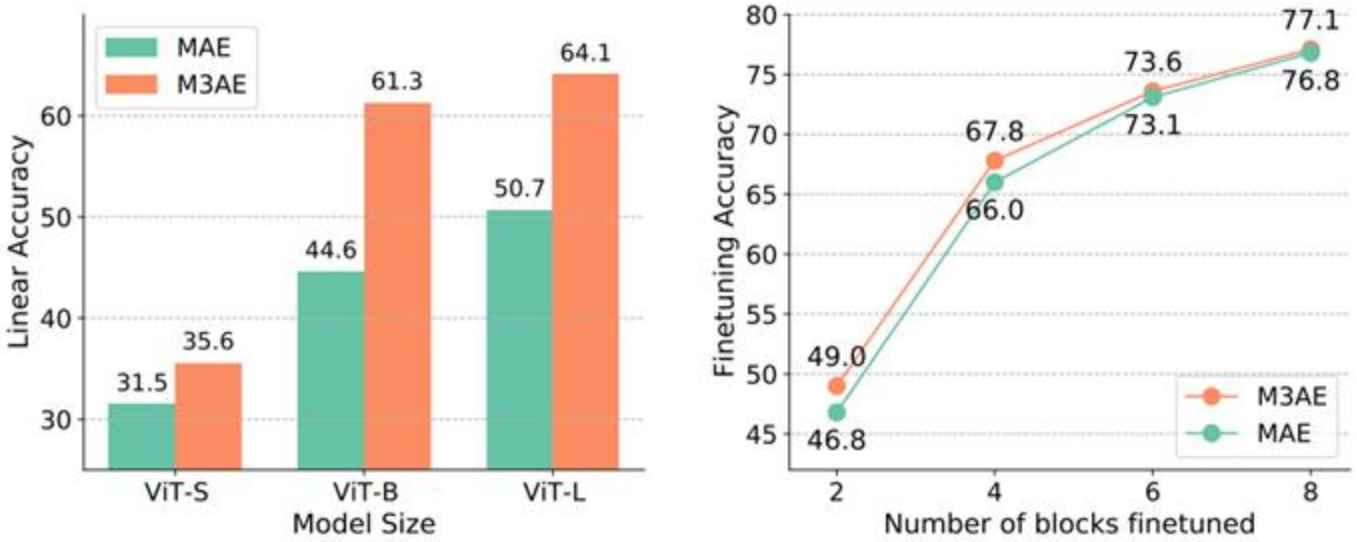


Figure 4: Left: Comparing the linear classification accuracy ViT model variants of different capacities (ViT-S/B/L). All models are pre-trained for 50 epochs. M3AE scales well with model size, outperforming MAE in every setting. **Right**: Comparing finetuning different number of blocks for ViT-L. All models are pre-trained for 50 epochs.

Outline

- Foundation models and Self-supervised learning
- Reconstruct from a corrupted (or partial) version
- Visual common sense tasks
- Contrastive Learning
- Feature Prediction
- Vision-language Foundation models

Relative Position of Image Patches

Unsupervised Visual Representation Learning by Context Prediction

Carl Doersch[1,2] Abhinav Gupta[1] Alexei A. Efros[2]

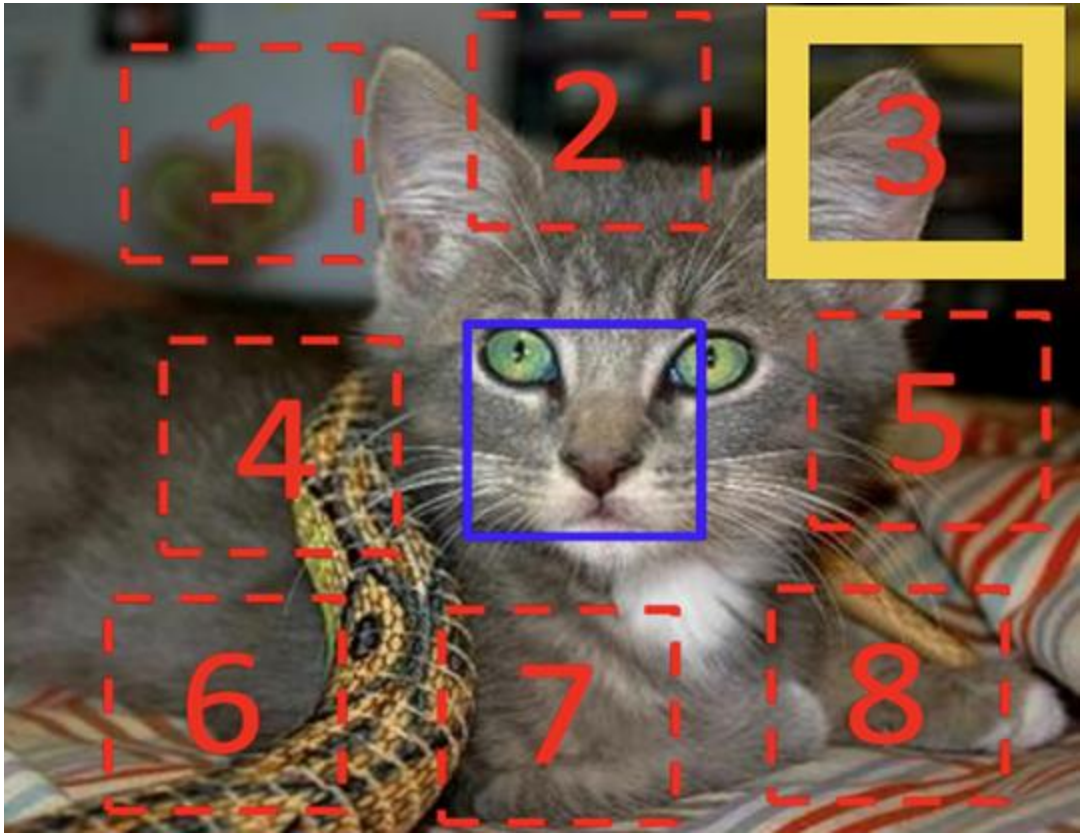
[1] School of Computer Science
Carnegie Mellon University

[2] Dept. of Electrical Engineering and Computer Science
University of California, Berkeley

Task: Predict the relative position of the second patch with respect to the first

Slide: Zisserman

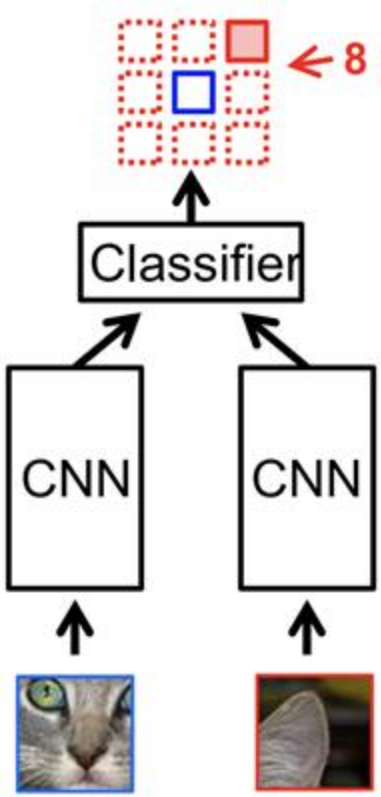
Relative Position of Image Patches

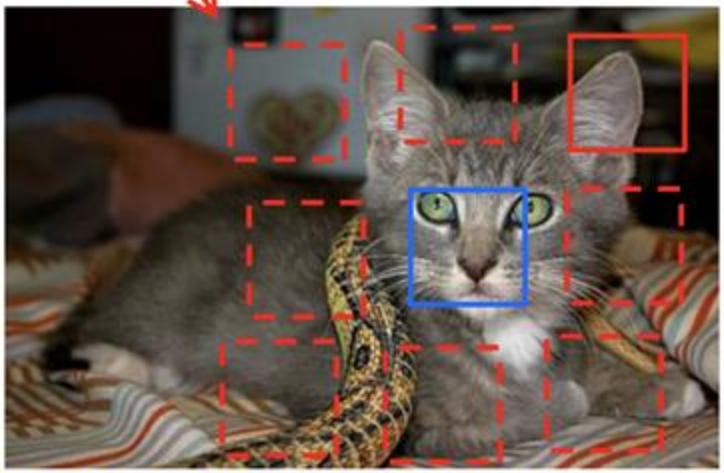


Doersch, Gupta, Efros

Slide: Zisserman

Relative Position of Image Patches

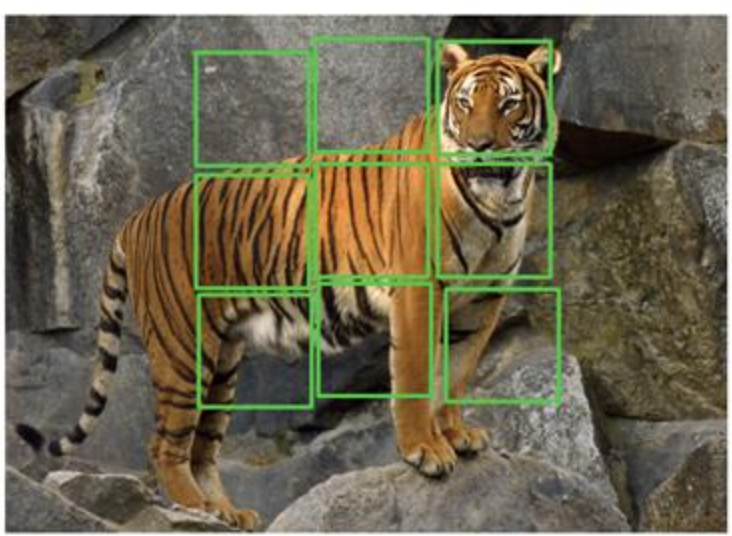




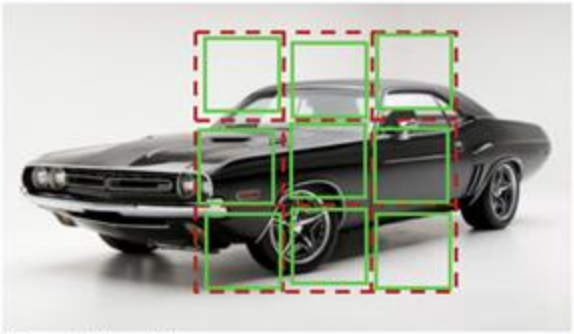
Randomly Sample Patch
Sample Second Patch

Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

Solving Jigsaw Puzzles



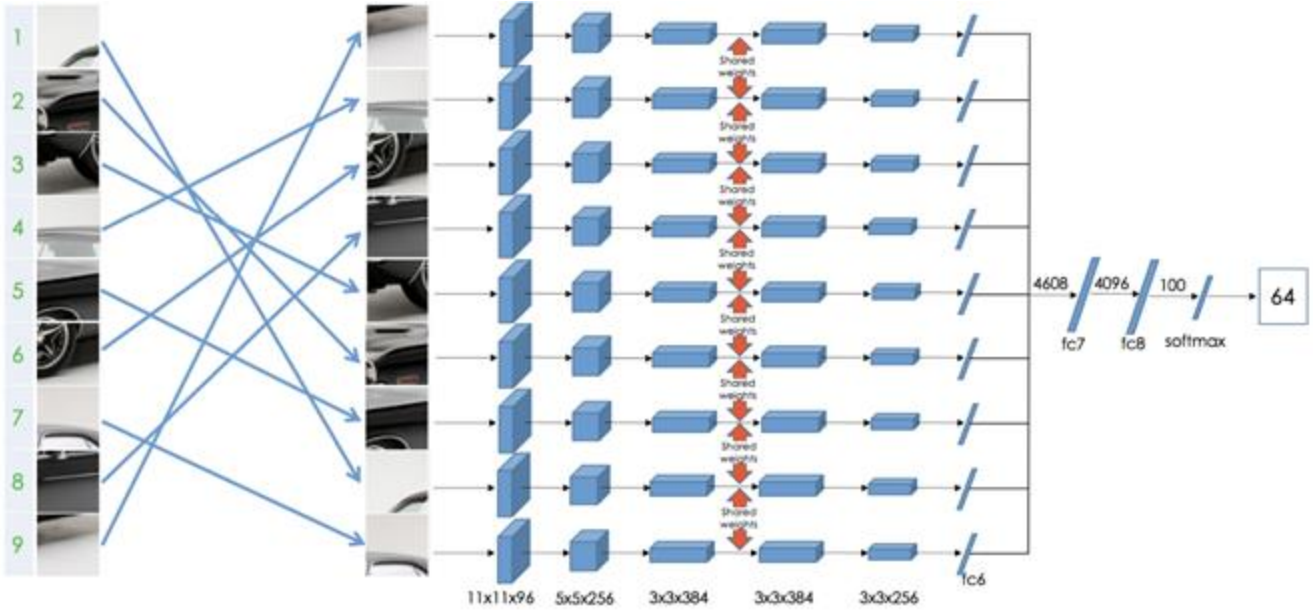
Solving Jigsaw Puzzles



Permutation Set

index	permutation
64	9,4,6,8,3,2,5,1,7

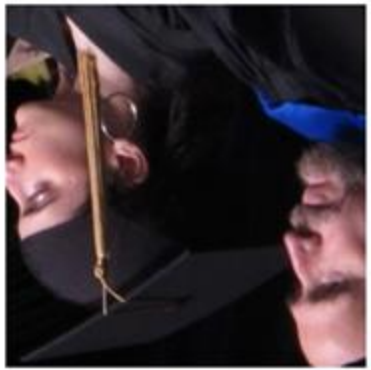
Reorder patches according to the selected permutation



Rotation

90° rotation

270° rotation



180° rotation

0° rotation

270° rotation

Rotation

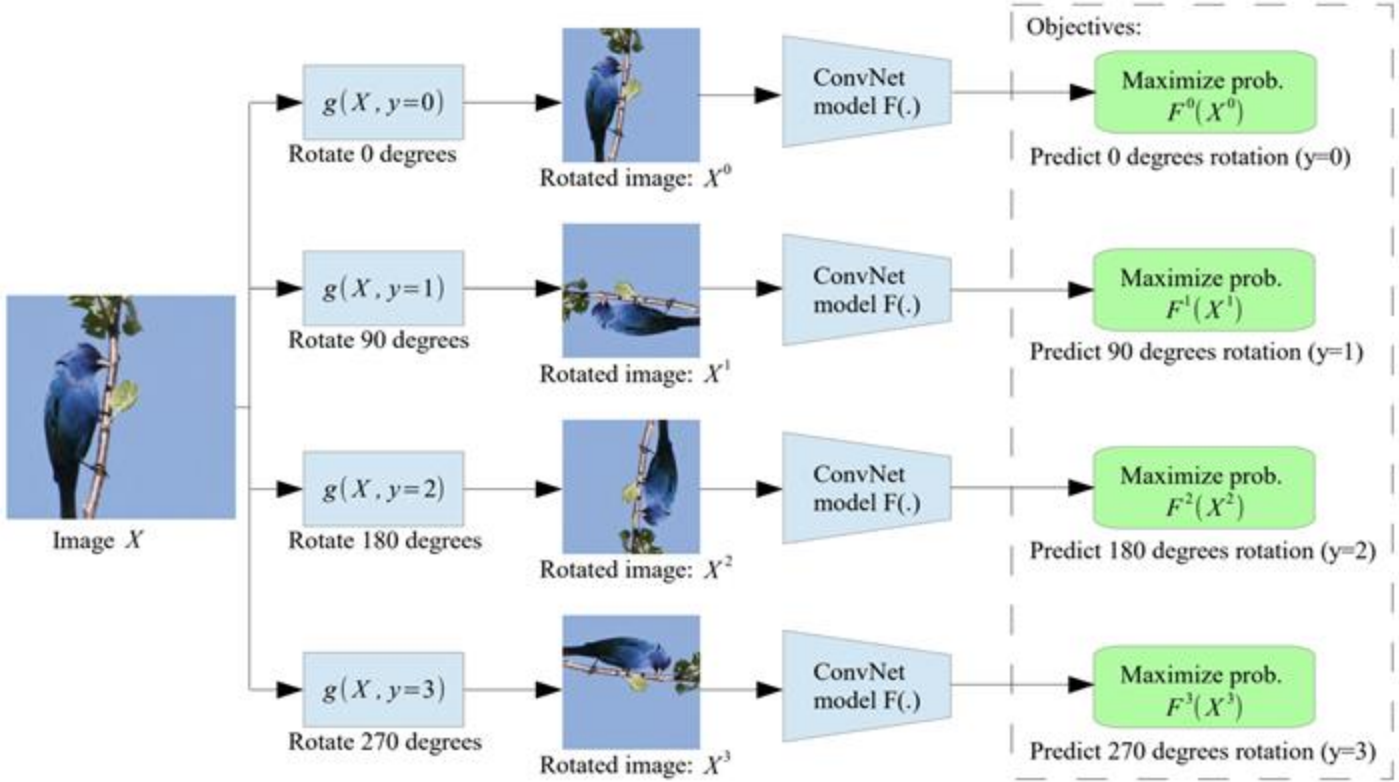


Image X
$g(X, y=0)$
Rotate 0 degrees
Rotated image: X^0
$g(X, y=1)$
Rotate 90 degrees
Rotated image: X^1
$g(X, y=2)$
Rotate 180 degrees
Rotated image: X^2
$g(X, y=3)$
Rotate 270 degrees
Rotated image: X^3
ConvNet model F(.)
ConvNet model F(.)
ConvNet model F(.)
ConvNet model F(.)
Objectives:
Maximize prob. $F^0(X^0)$
Predict 0 degrees rotation (y=0)
Maximize prob. $F^1(X^1)$
Predict 90 degrees rotation (y=1)
Maximize prob. $F^2(X^2)$
Predict 180 degrees rotation (y=2)
Maximize prob. $F^3(X^3)$
Predict 270 degrees rotation (y=3)

Rotation

	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5	29.7
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4	
Context (Doersch et al., 2015)	55.1	65.3	51.1	
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9	35.6
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9	34.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2	37.6
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4	
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7	36.0
ColorProxy (Larsson et al., 2017)		65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4	36.6
(Ours) RotNet	**70.87**	**72.97**	**54.4**	**39.1**

Outline

- Foundation models and Self-supervised learning
- Reconstruct from a corrupted (or partial) version
- Visual common sense tasks
- Contrastive Learning
- Feature Prediction
- Vision-language Foundation models

Momentum Contrast (MoCo)

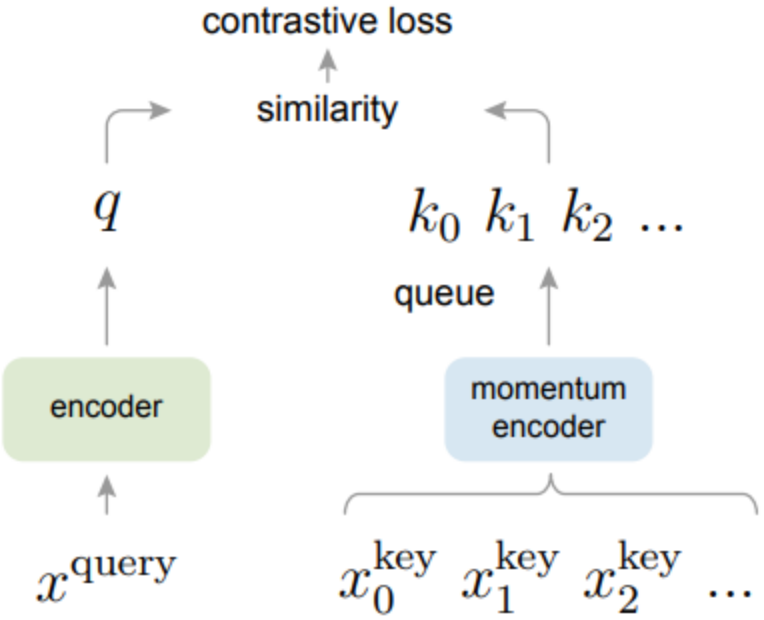
Momentum Contrast for Unsupervised Visual Representation Learning

Kaiming He Haoqi Fan Yuxin Wu Saining Xie Ross Girshick

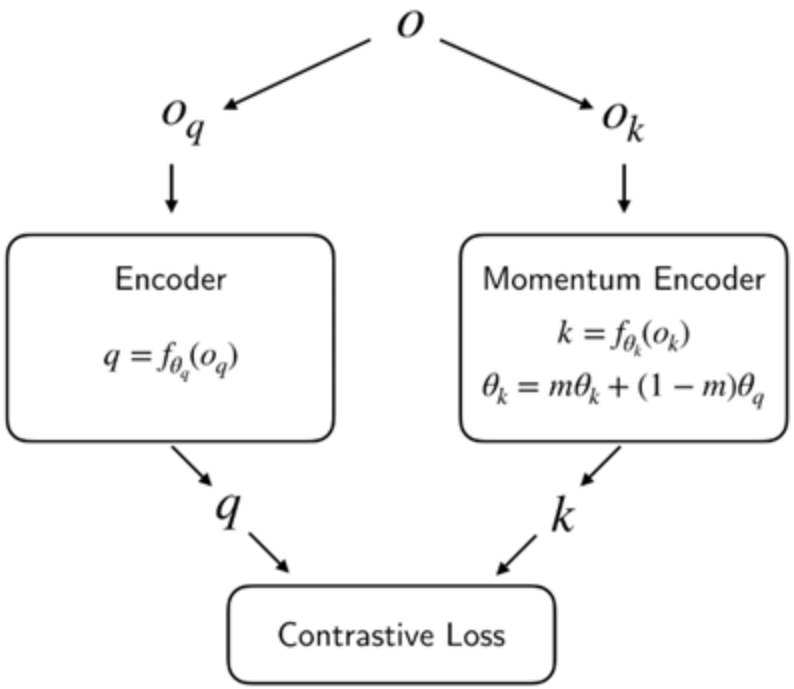
Facebook AI Research (FAIR)

Nov 2019

Momentum Contrast (MoCo)



Momentum Contrast (MoCo)



$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

Momentum Contrast (MoCo)

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

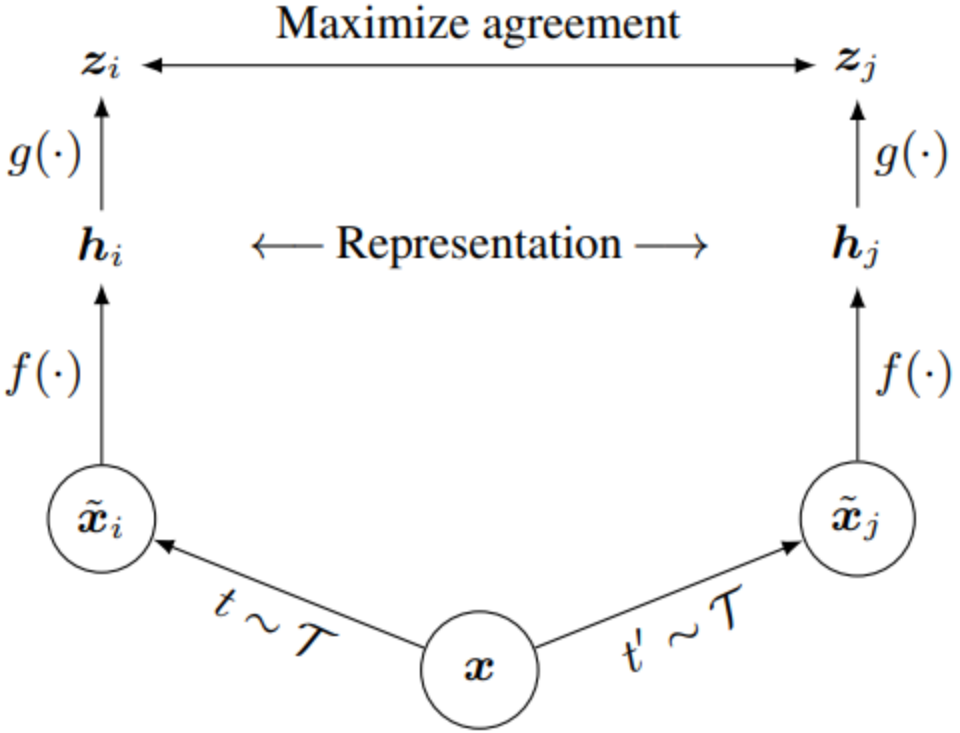
bmm: **batch matrix multiplication**; mm: **matrix multiplication**; cat: **concatenation**.

SimCLR

A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen [1] Simon Kornblith [1] Mohammad Norouzi [1] Geoffrey Hinton [1]

SimCLR

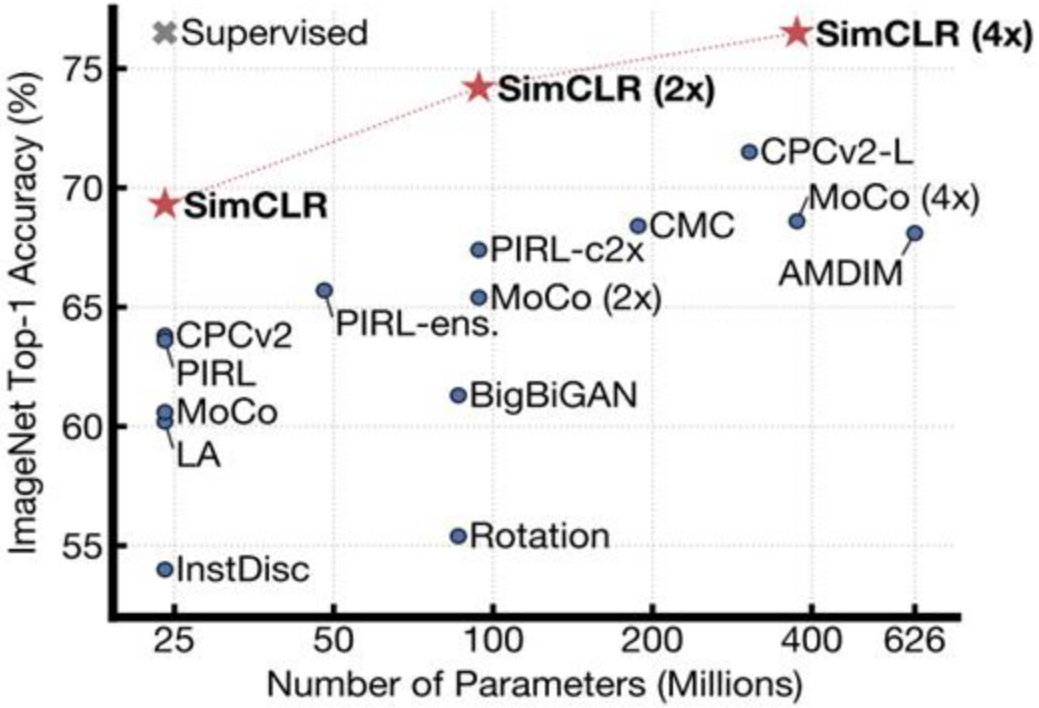

Maximize agreement
z_i
z_j
$g(\cdot)$
$g(\cdot)$
h_i
Representation
h_j
$f(\cdot)$
$f(\cdot)$
$\tilde{x}_i$
$\tilde{x}_j$
$t \sim \mathcal{T}$
x
$t' \sim \mathcal{T}$

SimCLR

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, temperature τ, structure of $f, g, \mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}, t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\tau \|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\ell(i, j) = -\log \frac{\exp(s_{i,j})}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k})}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network f

SimCLR


Supervised
SimCLR (4x)
SimCLR (2x)
CPCv2-L
MoCo (4x)
SimCLR
CMC
PIRL-c2x
AMDIM
MoCo (2x)
PIRL-ens.
CPCv2
PIRL
BigBiGAN
MoCo
LA
Rotation
InstDisc
ImageNet Top-1 Accuracy (%)
75
70
65
60
55
25
50
100
200
400
626
Number of Parameters (Millions)

Outline

- Foundation models and Self-supervised learning
- Reconstruct from a corrupted (or partial) version
- Visual common sense tasks
- Contrastive Learning
- Feature Prediction
- Vision-language Foundation models

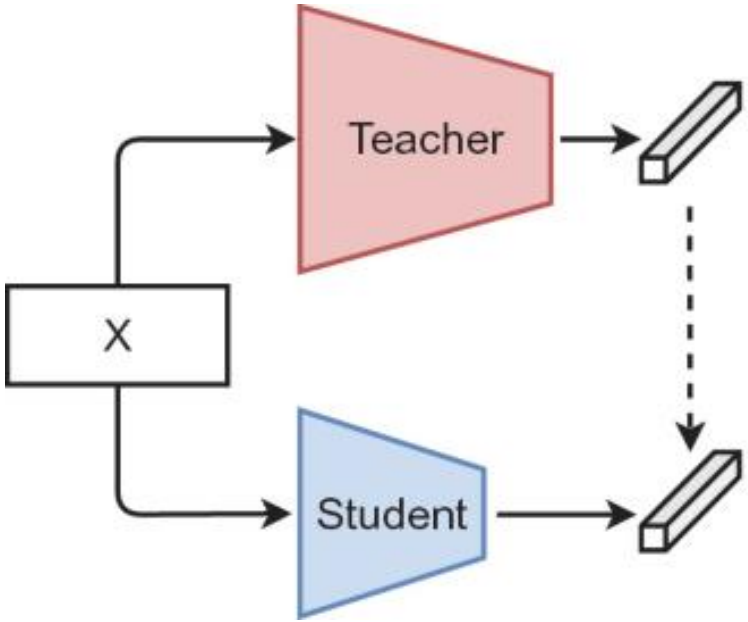
DINO

Emerging Properties in Self-Supervised Vision Transformers

Mathilde Caron[1,2] Hugo Touvron[1,3] Ishan Misra[1] Hervé Jegou[1]
Julien Mairal[2] Piotr Bojanowski[1] Armand Joulin[1]

[1] Facebook AI Research [2] Inria* [3] Sorbonne University

DINO



Consider **knowledge distillation**

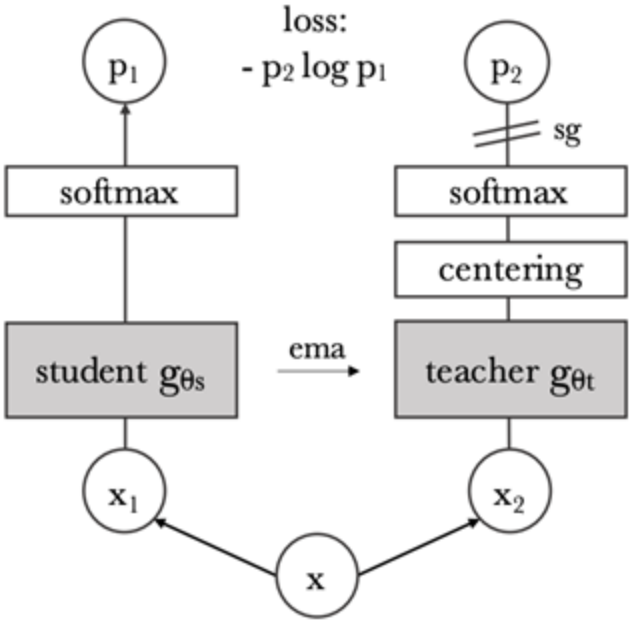
- Student network g_{θ_t} tries to match a teacher network g_{θ_s}
- Minimize the cross entropy of the distributions

$$\min_{\theta_s} H(P_t(x), P_s(x)) \quad P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s)}$$

$$\text{where } H(a, b) = -a \log b$$

DINO

Self supervised learning as knowledge distillation



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

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

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

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

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

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

DINO

Apply centering to avoid collapse - use EMA so things work across different batch sizes

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

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

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

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

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

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

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

DINO

Supervised

DINO

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

Threshold attention map get mask

Compare similarity to ground truth mask

Linear and k-NN classification on ImageNet.

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

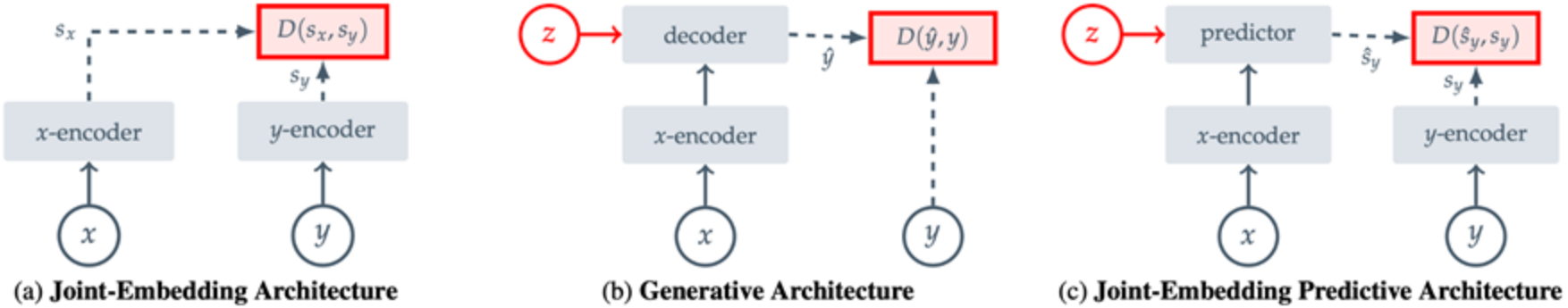
I-JEPA

Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture

Mahmoud Assran[1,2,3*] **Quentin Duval**[1] **Ishan Misra**[1] **Piotr Bojanowski**[1]
Pascal Vincent[1] **Michael Rabbat**[1,3] **Yann LeCun**[1,4] **Nicolas Ballas**[1]

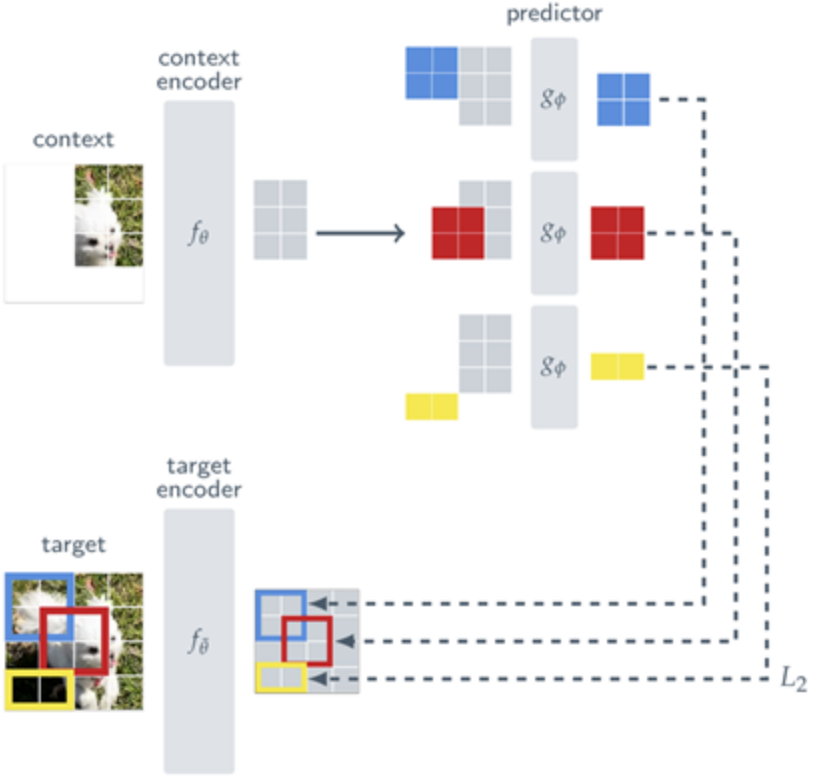
[1]Meta AI (FAIR) [2]McGill University [3] Mila, Quebec AI Institute [4]New York University

I-JEPA



(a) **Joint-Embedding Architecture** (b) **Generative Architecture** (c) **Joint-Embedding Predictive Architecture**

I-JEPA



Context Encoder, Target Encoder and Predictor are ViTs

Predictor

- Transformer encoder
- Concat context tokens
- Have masked tokens for prediction patches

$$\hat{\boldsymbol{s}}_y(i) = \{\hat{\boldsymbol{s}}_{y_j}\}_{j\in B_i} = g_\phi(\boldsymbol{s}_x, \{\boldsymbol{m}_j\}_{j\in B_i})$$

$$\frac{1}{M}\sum_{i=1}^{M} D\left(\hat{\boldsymbol{s}}_y(i), \boldsymbol{s}_y(i)\right) = \frac{1}{M}\sum_{i=1}^{M}\sum_{j\in B_i} \|\hat{\boldsymbol{s}}_{y_j} - \boldsymbol{s}_{y_j}\|_2^2.$$

I-JEPA

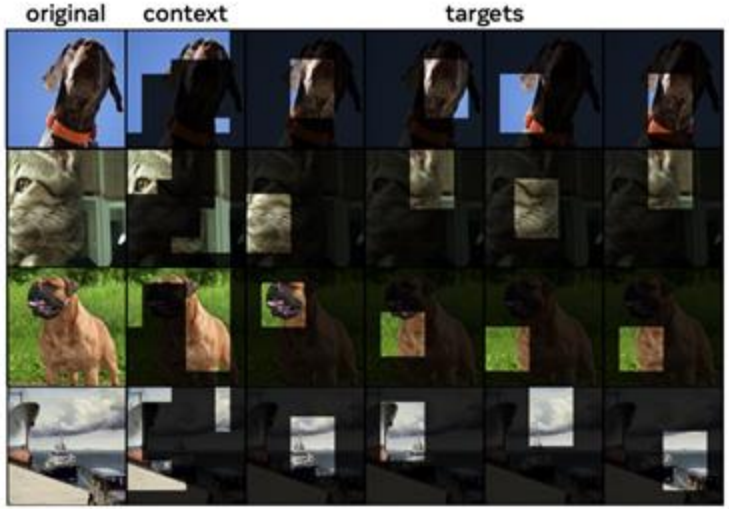
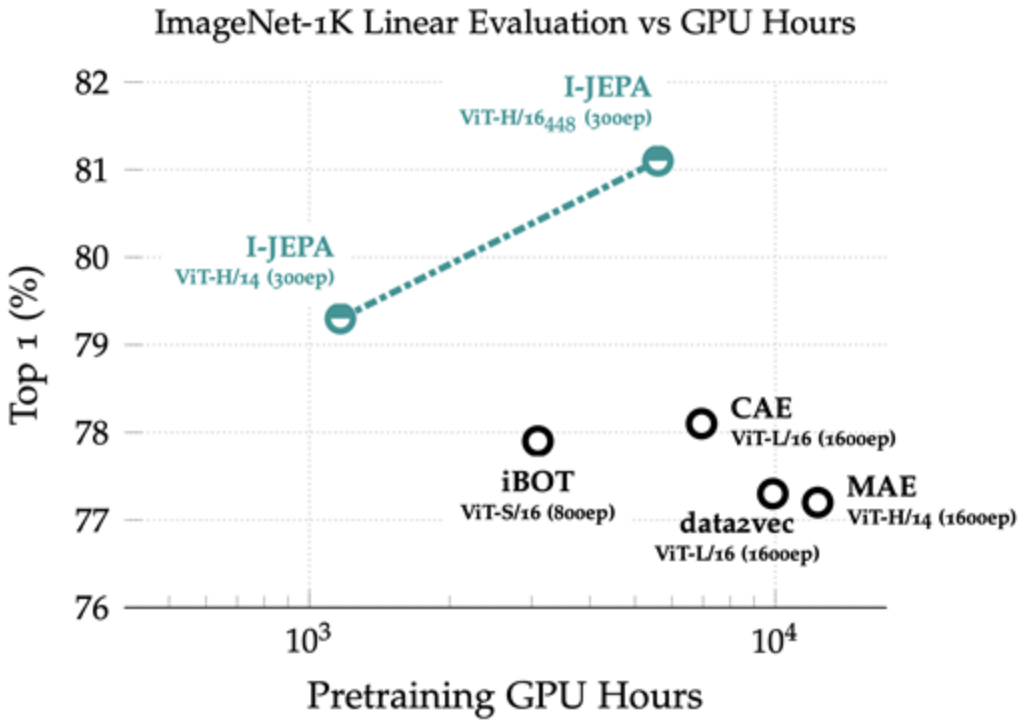


Figure 4. Examples of our context and target-masking strategy. Given an image, we randomly sample 4 target blocks with scale in the range $(0.15, 0.2)$ and aspect ratio in the range $(0.75, 1.5)$. Next, we randomly sample a context block with scale in the range $(0.85, 1.0)$ and remove any overlapping target blocks. Under this strategy, the target-blocks are relatively semantic, and the context-block is informative, yet sparse (efficient to process).

Context and Target Selection

I-JEPA

Method	Arch.	Epochs	Top-1
Methods without view data augmentations			
data2vec [8]	ViT-L/16	1600	77.3
MAE [36]	ViT-B/16	1600	68.0
	ViT-L/16	1600	76.0
	ViT-H/14	1600	77.2
CAE [22]	ViT-B/16	1600	70.4
	ViT-L/16	1600	78.1
I-JEPA	ViT-B/16	600	72.9
	ViT-L/16	600	77.5
	ViT-H/14	300	79.3
	ViT-H/16_{448}	300	**81.1**
Methods using extra view data augmentations			
SimCLR v2 [21]	RN152 (2×)	800	79.1
DINO [18]	ViT-B/8	300	80.1
iBOT [79]	ViT-L/16	250	**81.0**



Freeze context encoder and predictor

Train a RCDM (representation conditioned diffusion model to visualize predictions

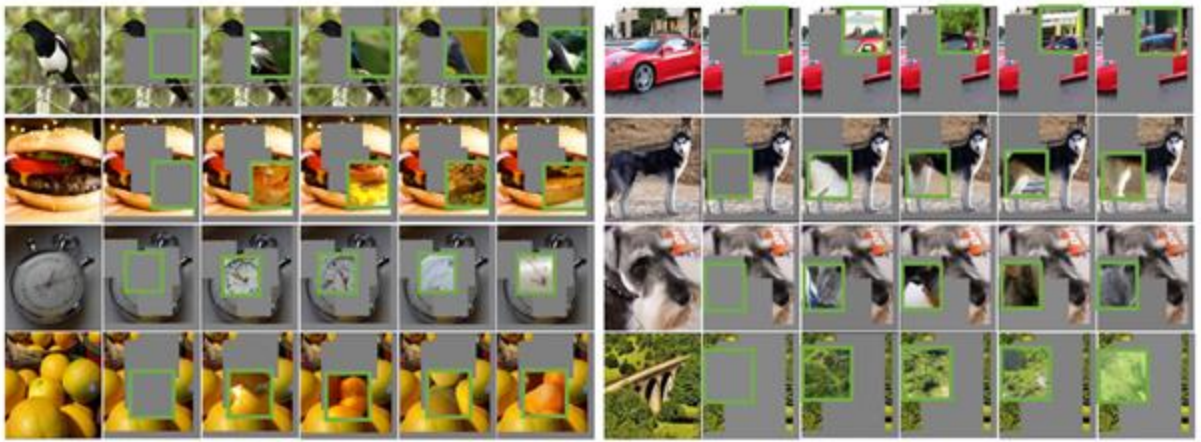


Figure 6. **Visualization of I-JEPA predictor representations**. For each image: first column contains the original image; second column contains the context image, which is processed by a pretrained I-JEPA ViT-H/14 encoder. Green bounding boxes in subsequent columns contain samples from a generative model decoding the output of the pretrained I-JEPA predictor, which is conditioned on positional mask tokens corresponding to the location of the green bounding box. Qualities that are common across samples represent information that

Outline

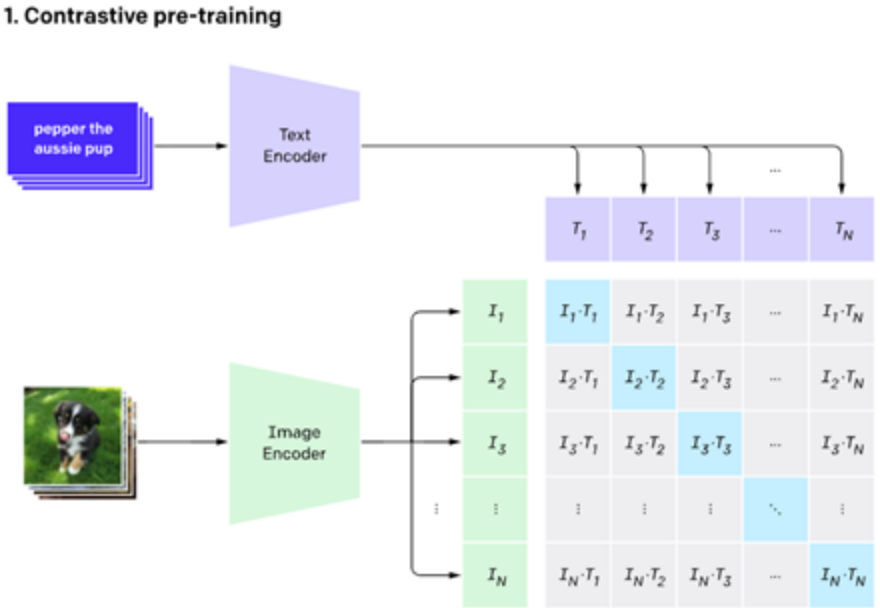
- Foundation models and Self-supervised learning
- Reconstruct from a corrupted (or partial) version
- Visual common sense tasks
- Contrastive Learning
- Feature Prediction
- Vision-language Foundation models

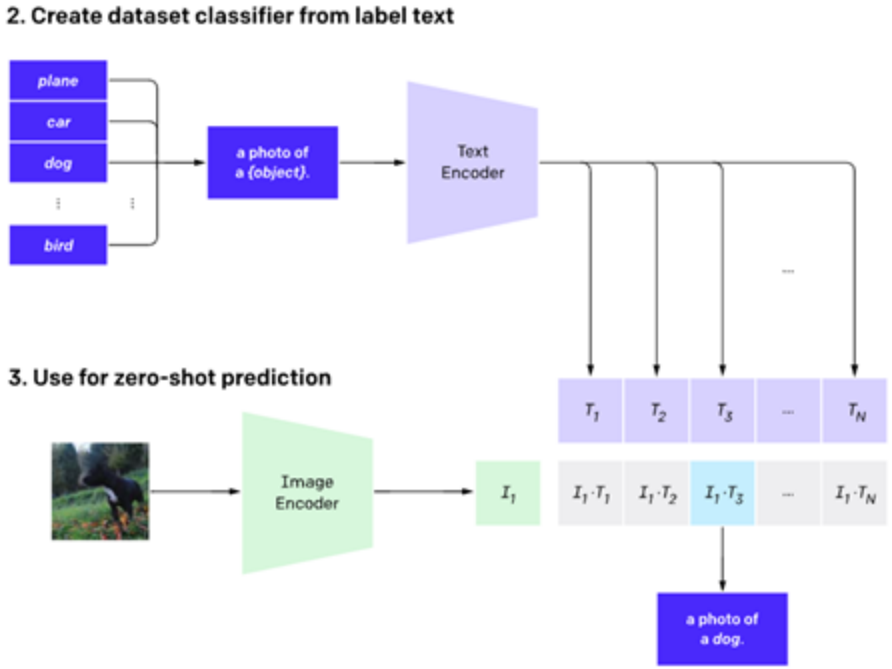
CLIP

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford [*1] Jong Wook Kim [*1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1]
Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

CLIP


1. Contrastive pre-training
pepper the aussie pup
Text Encoder
T_1 T_2 T_3 ... T_N
Image Encoder
I_1 I_2 I_3 I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$


2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
T_1 T_2 T_3 ... T_N
3. Use for zero-shot prediction
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
a photo of a dog.

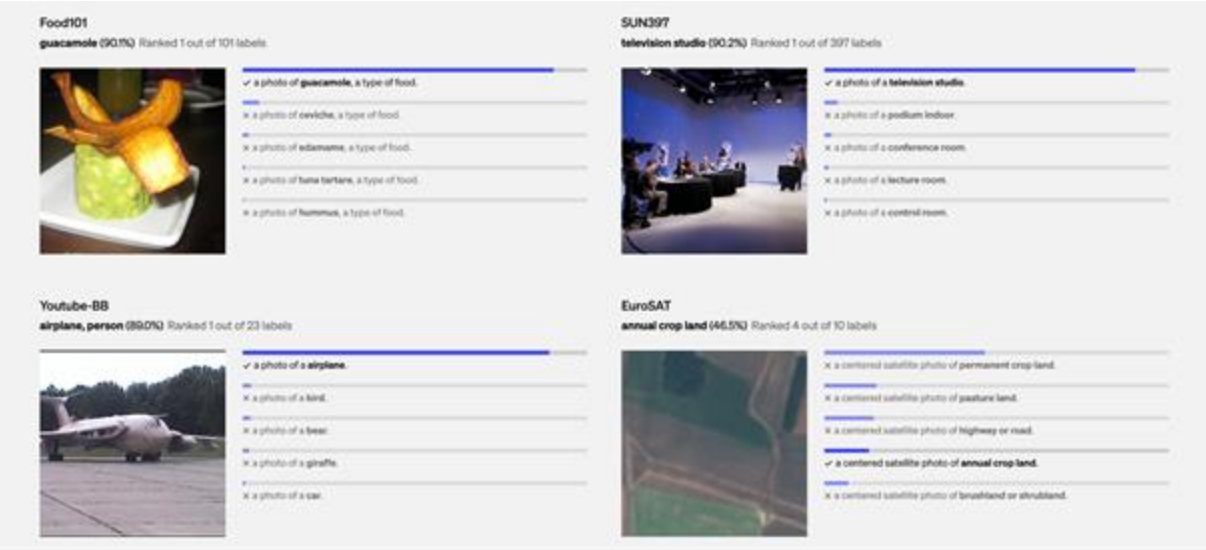
CLIP

Dataset

- Existing text annotated image dataset at the time were relatively small
- YFCC100M
  - Text metadata quality is low, some captions are automatically generated file names like “20160716 113957.JPG”
- Constructed dataset of 400M image-text pairs
- Images searched with one of 500K generated queries

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

CLIP



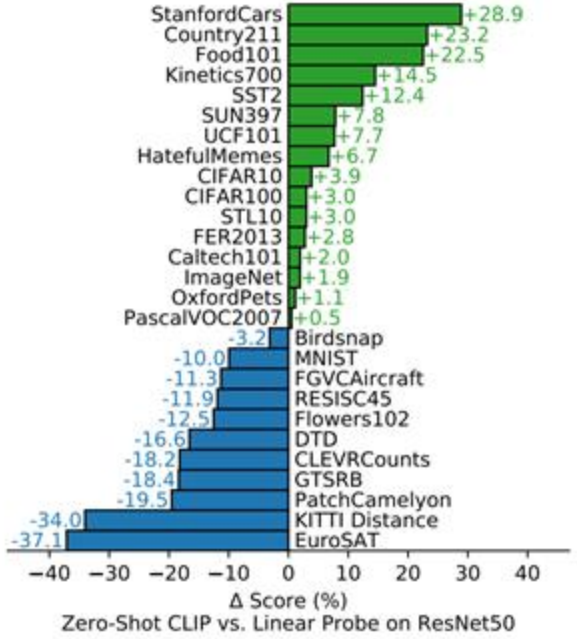
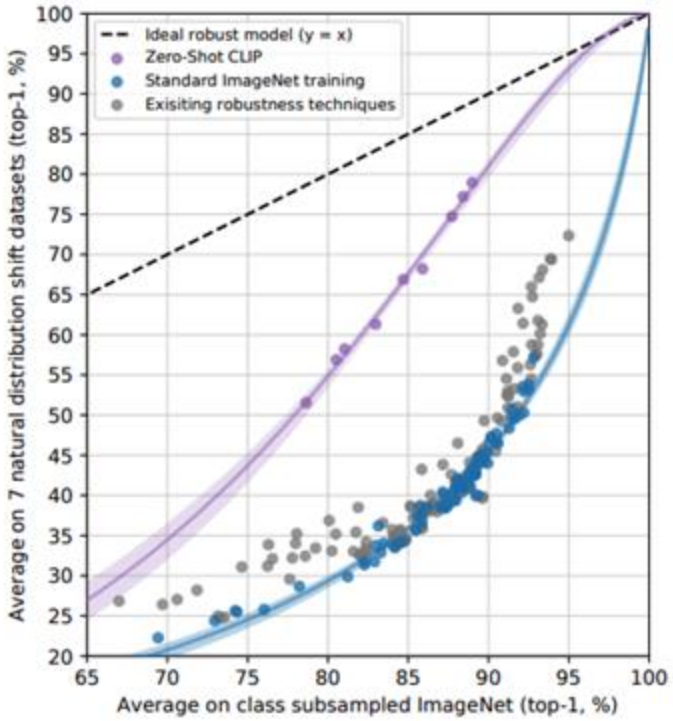


Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

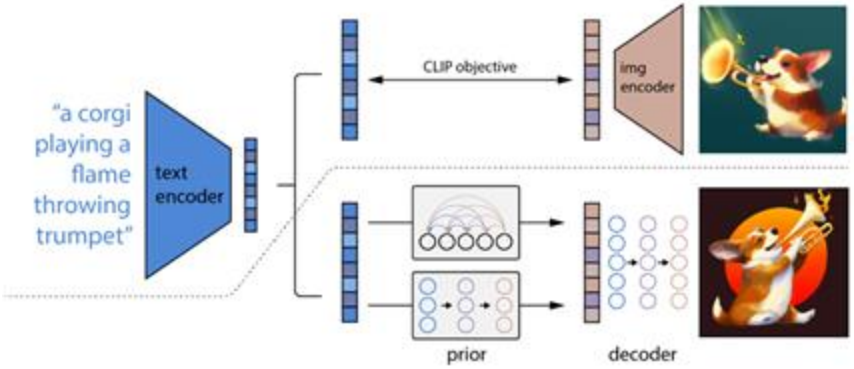
CLIP


Ideal robust model (y = x)
Zero-Shot CLIP
Standard ImageNet training
Exisiting robustness techniques
Average on 7 natural distribution shift datasets (top-1, %)
Average on class subsampled ImageNet (top-1, %)
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
65
70
75
80
85
90
95
100

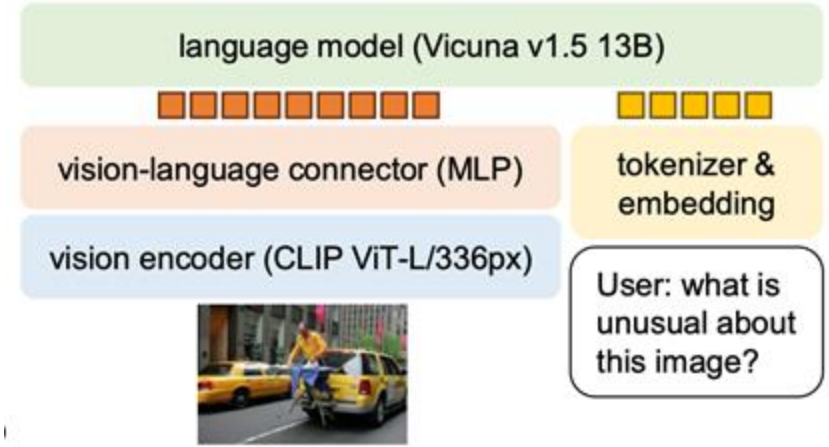

Dataset Examples
ImageNet ResNet101
Zero-Shot CLIP
Δ Score
ImageNet
76.2
76.2
0%
ImageNetV2
64.3
70.1
+5.8%
ImageNet-R
37.7
88.9
+51.2%
ObjectNet
32.6
72.3
+39.7%
ImageNet Sketch
25.2
60.2
+35.0%
ImageNet-A
2.7
77.1
+74.4%

CLIP

CLIP learns features useful for other model



unCLIP



LLaVA

FLIP

Scaling Language-Image Pre-training via Masking

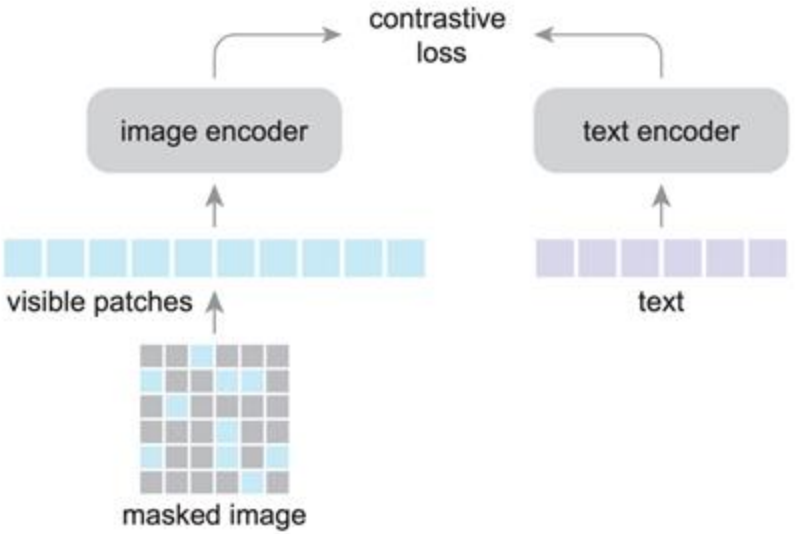
Yanghao Li* Haoqi Fan* Ronghang Hu* Christoph Feichtenhofer† Kaiming He†

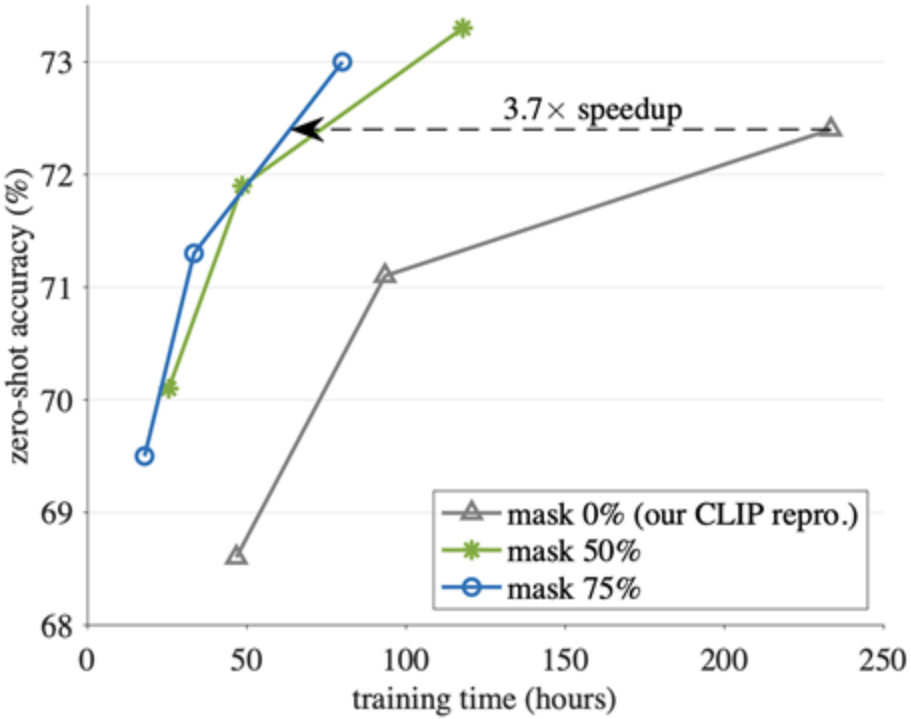
*equal technical contribution, †equal advising

Meta AI, FAIR

https://github.com/facebookresearch/flip

FLIP

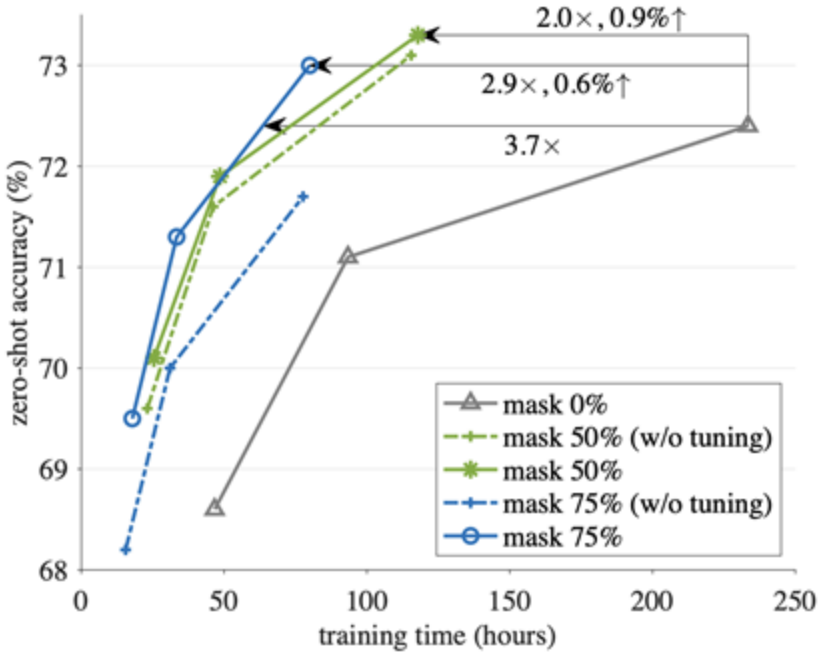

contrastive loss
image encoder
text encoder
visible patches
text
masked image


3.7× speedup
zero-shot accuracy (%)
training time (hours)
mask 0% (our CLIP repro.)
mask 50%
mask 75%
73
72
71
70
69
68
0
50
100
150
200
250

FLIP

	mask 50%	mask 75%
baseline	69.6	68.2
+ tuning	**70.1**	**69.5**

(e) **Unmasked tuning**. The distribution shift by masking is reduced by a short tuning.



FLIP

case	data	epochs	B/16	L/16	L/14	H/14
CLIP [52]	WIT-400M	32	68.6	-	75.3	-
OpenCLIP [36]	LAION-400M	32	67.1	-	72.8	-
CLIP, our repro.	LAION-400M	32	68.2	72.4	73.1	-
FLIP	LAION-400M	32	68.0	74.3	74.6	75.5

Table 2. **Zero-shot accuracy on ImageNet-1K classification**, compared with various CLIP baselines. The image size is 224. The entries noted by grey are pre-trained on a different dataset. Our models use a 64k batch, 50% masking ratio, and unmasked tuning.

case	data	epochs	model	zero-shot	linear probe	fine-tune
CLIP [52]	WIT-400M	32	L/14	75.3	83.9†	-
CLIP [52], our transfer	WIT-400M	32	L/14	75.3	83.0	87.4
OpenCLIP [36]	LAION-400M	32	L/14	72.8	82.1	86.2
CLIP, our repro.	LAION-400M	32	L/16	72.4	82.6	86.3
FLIP	LAION-400M	32	L/16	74.3	83.6	86.9

Table 3. **Linear probing and fine-tuning accuracy on ImageNet-1K classification**, compared with various CLIP baselines. The entries noted by grey are pre-trained on a different dataset. The image size is 224. †: CLIP in [52] optimizes with L-BFGS; we use SGD instead.

Thank you!