



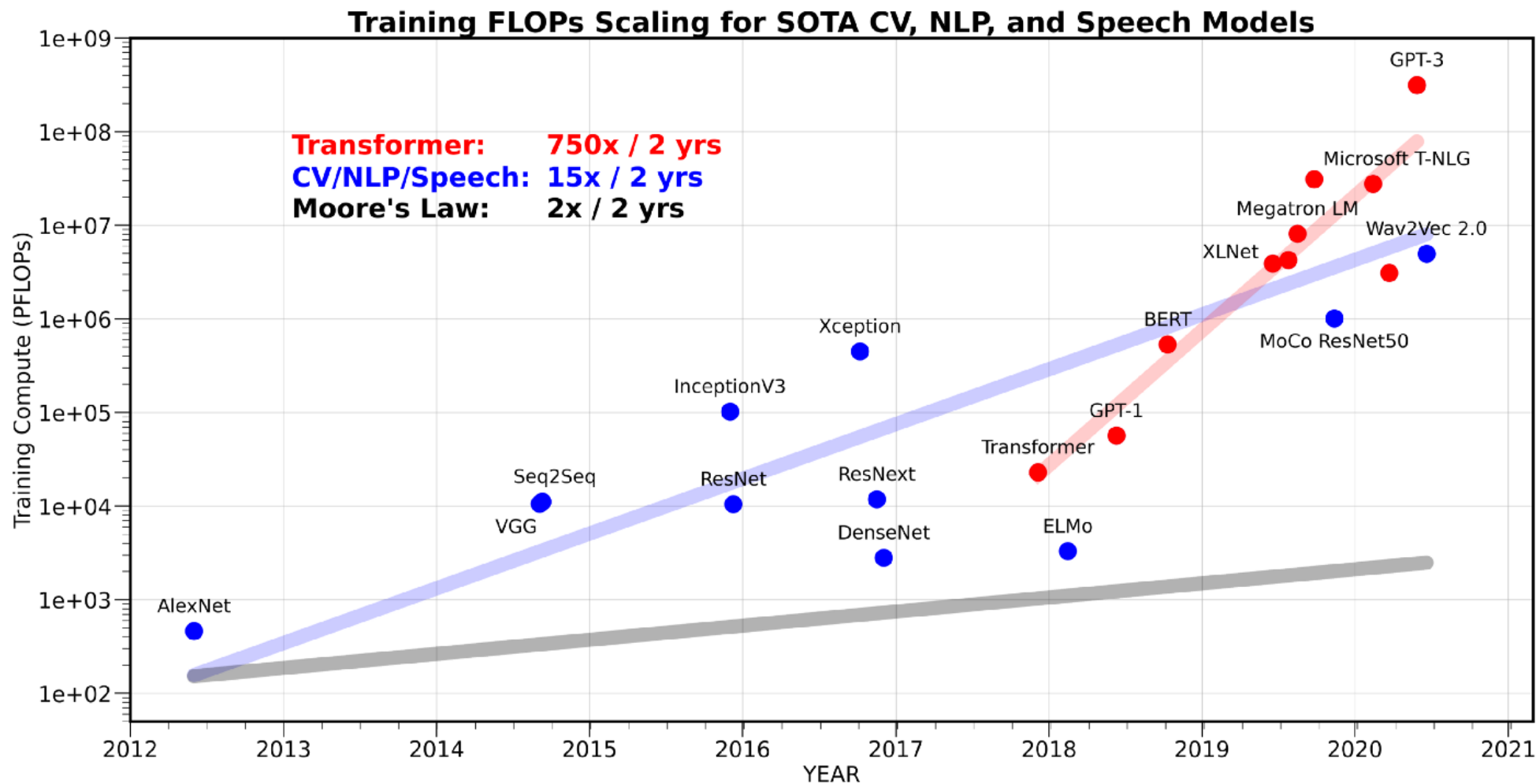
Natural Language Processing

第十二周 多模态学习

庞彦

yanpang@gzhu.edu.cn

Foundation Models



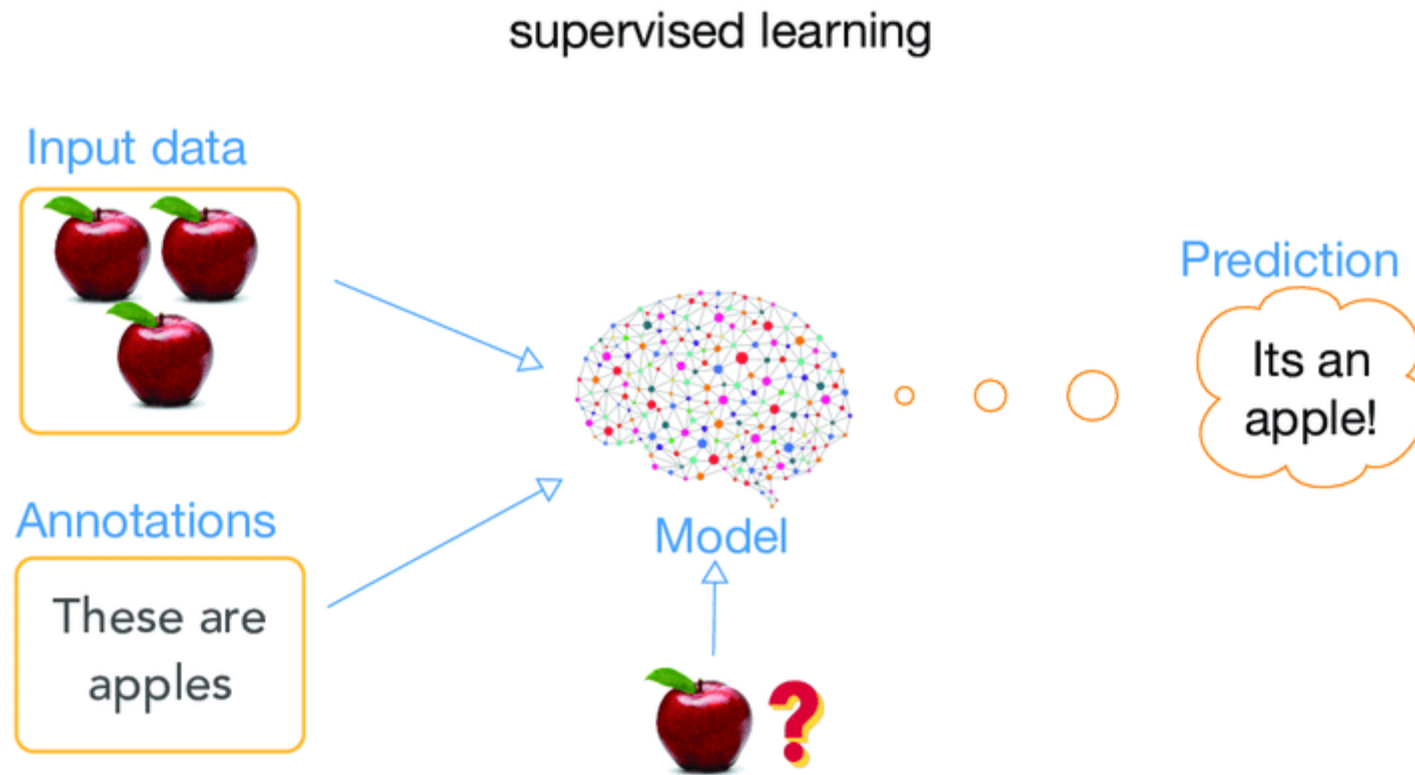


01 | Self-supervised Learning

自监督学习

Spring 2023

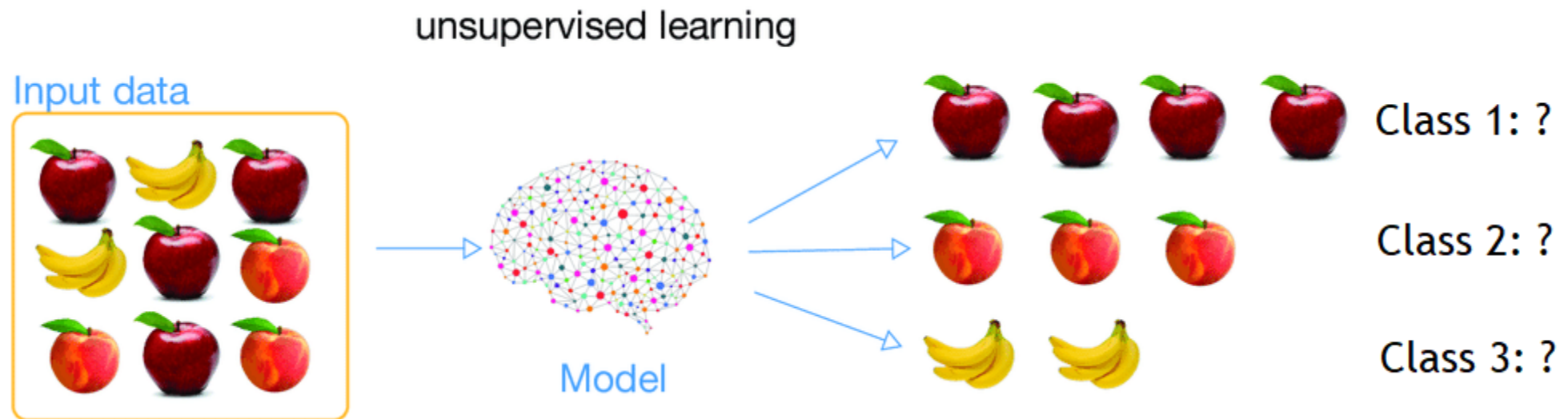
Supervised Learning



➤ **Supervised Learning**

Label: ✓

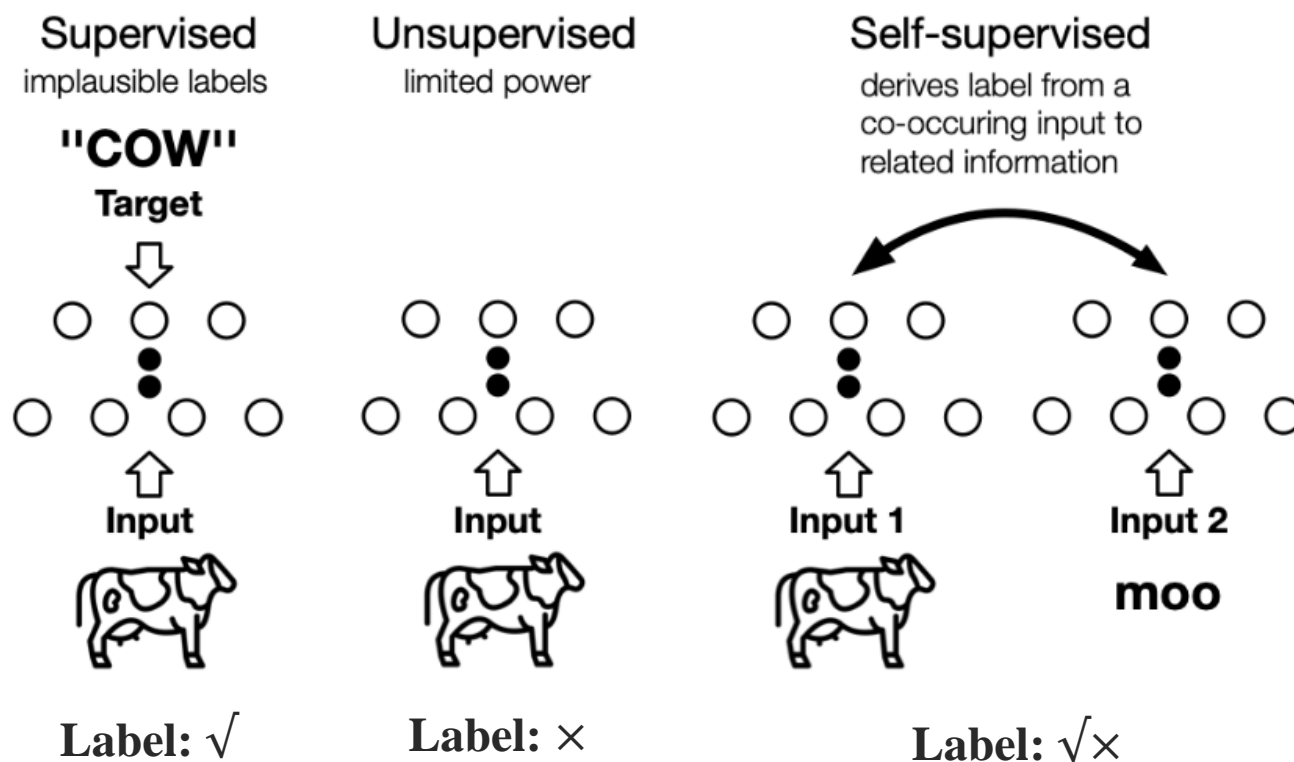
Unsupervised Learning



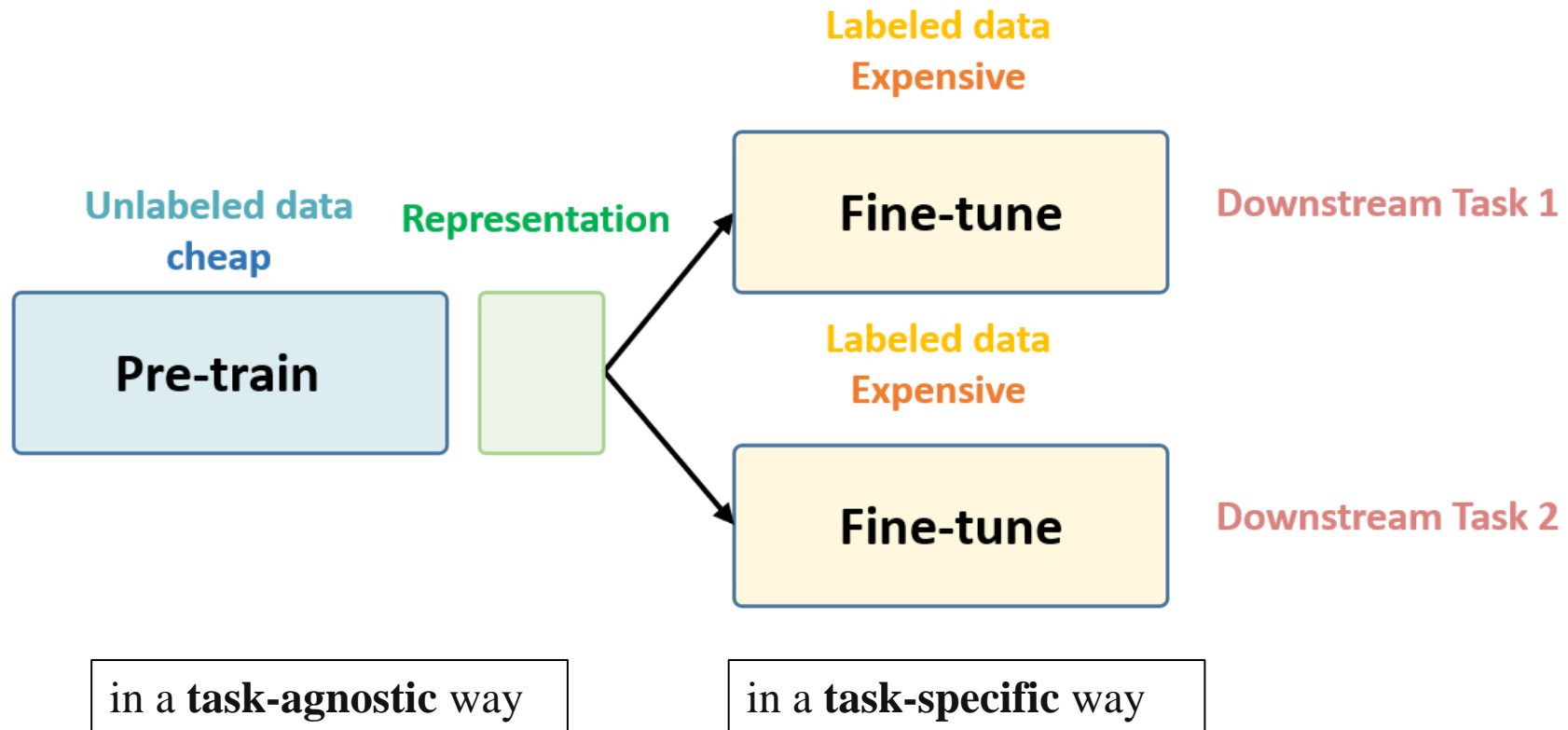
➤ **Unsupervised Learning** **Label: ×**

Semi-supervised Learning

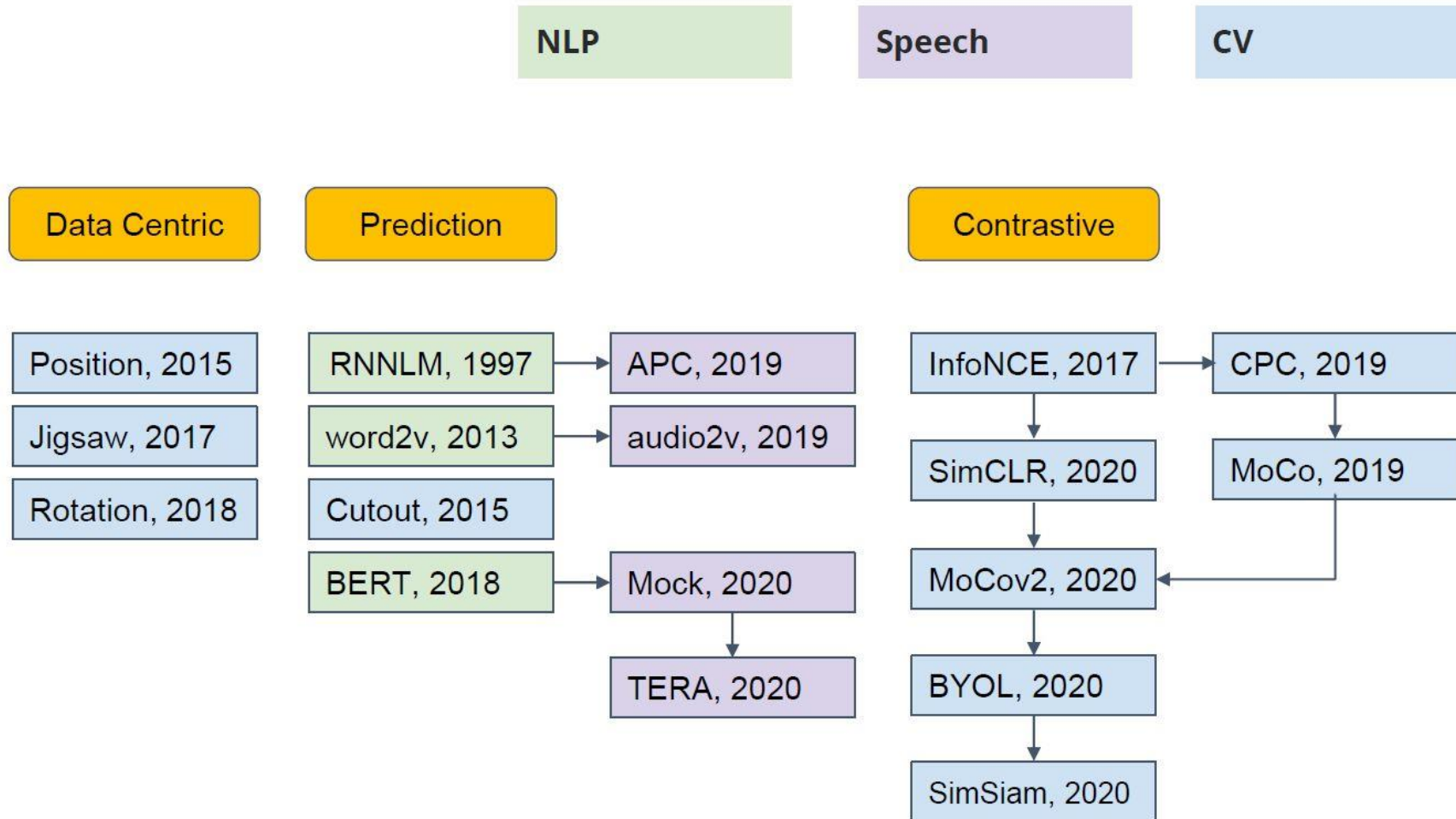
Semi-supervised learning is a learning problem that involves a **small** number of **labeled** examples and a **large** number of **unlabeled** examples



Self-Supervised Learning



Self-Supervised Learning



Self-Supervised Learning

Generative / Predictive



Loss measured in the output space
Examples: Colorization, Auto-Encoders

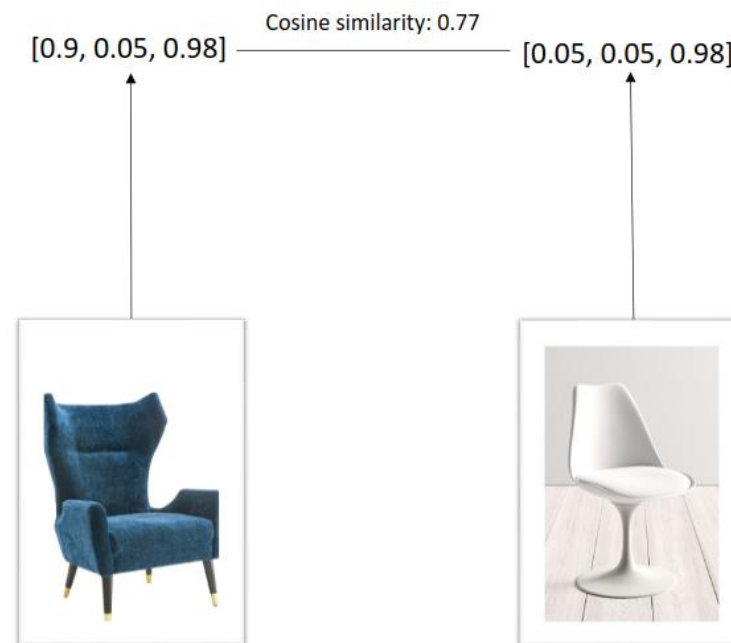
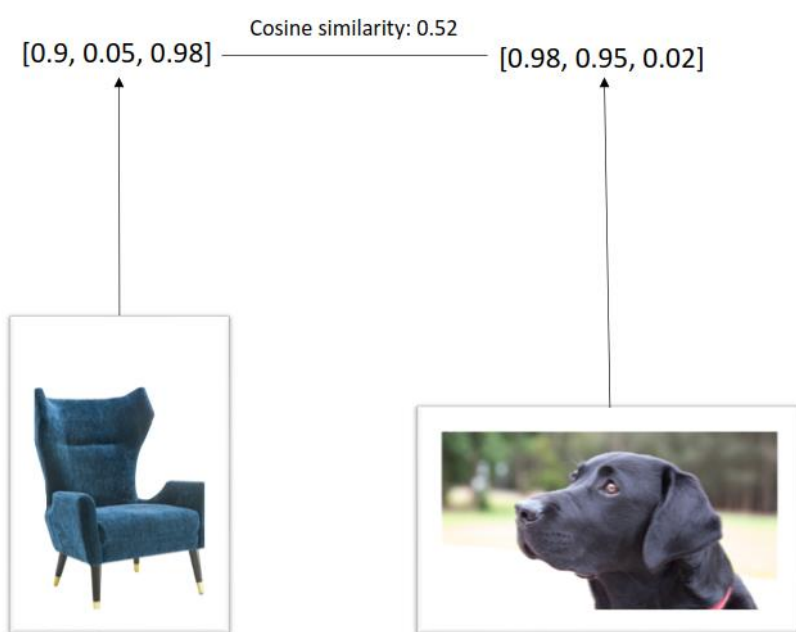
Contrastive



Loss measured in the representation space
Examples: TCN, CPC, Deep-InfoMax

Contrastive Learning

对比学习的目的是从图像中提取特征，同时努力将相似的图片（也称为正对）放在一起，而将不同的图片（也称为负对）放在很远的地方。





02

Prompt in NLP & CV

NLP和CV中的Prompt

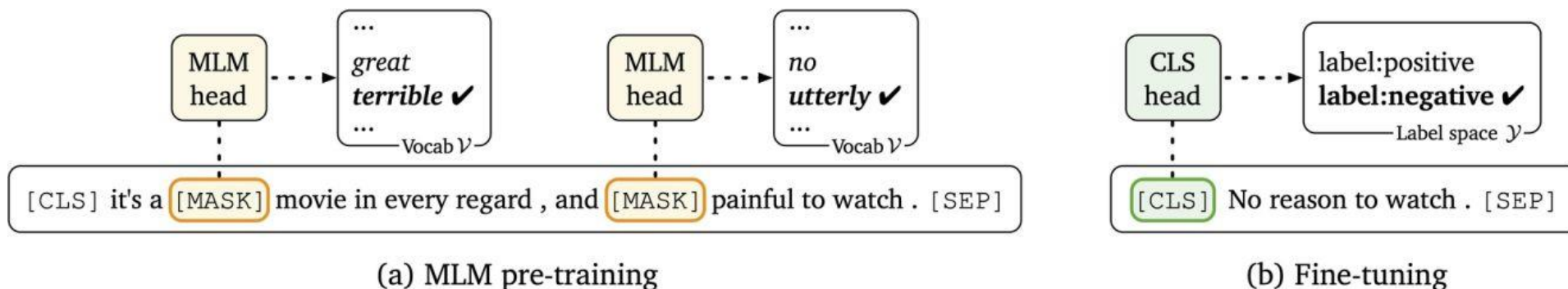
Spring 2023

Prompt in NLP

Randomly masked A quick [MASK] fox jumps over the [MASK] dog

↓

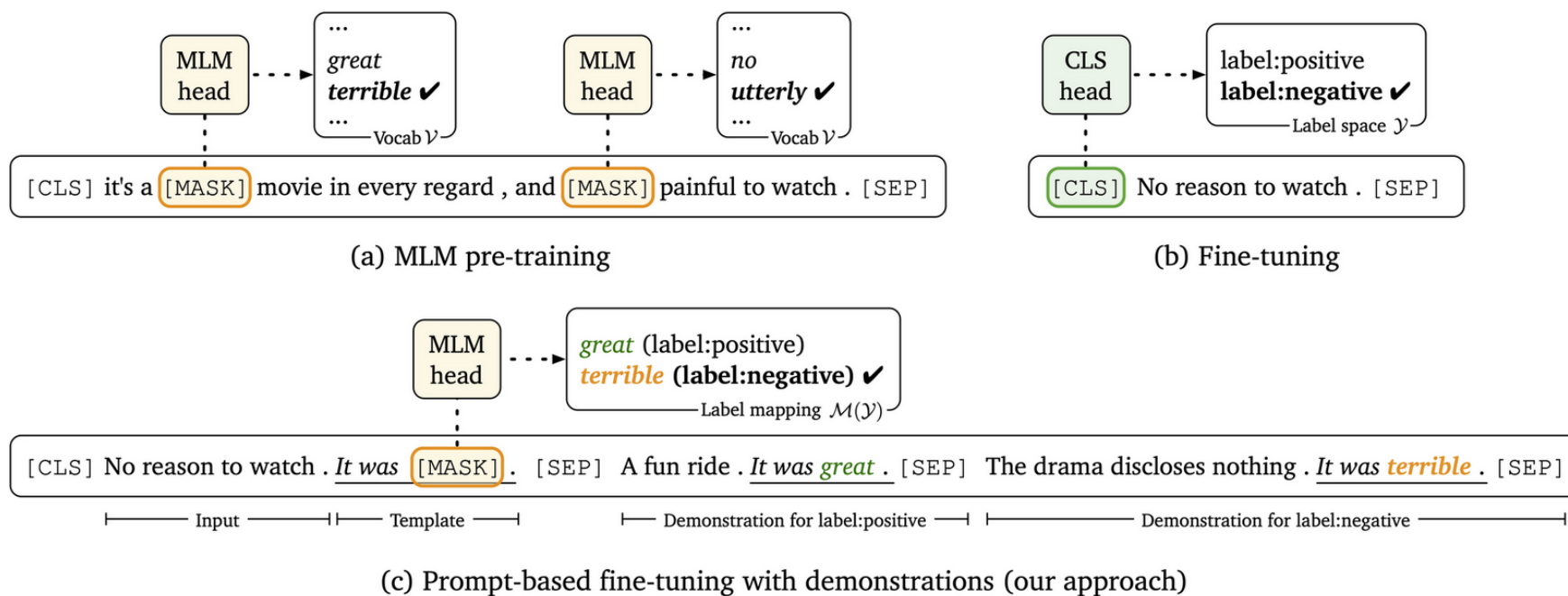
Predict A quick brown fox jumps over the lazy dog



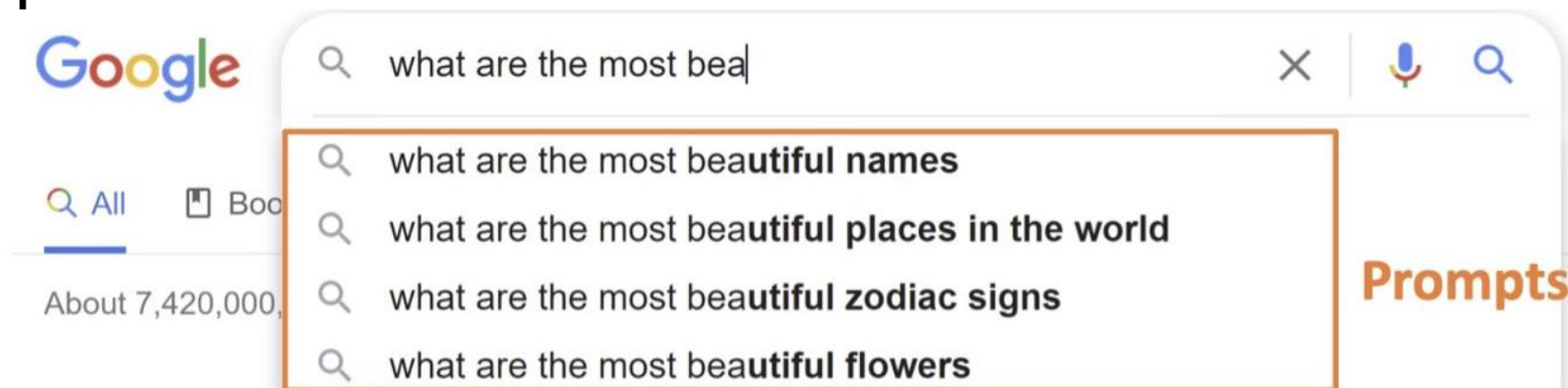
Prompt in NLP

A **prompt** is a piece of text **inserted** in the input examples, so that the original task can be formulated as a (**masked**) language modeling problem.

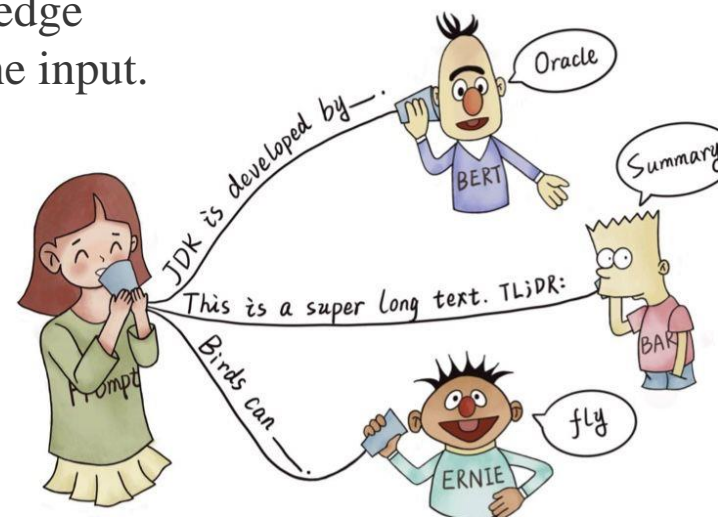
Why Prompts?



Prompt in NLP

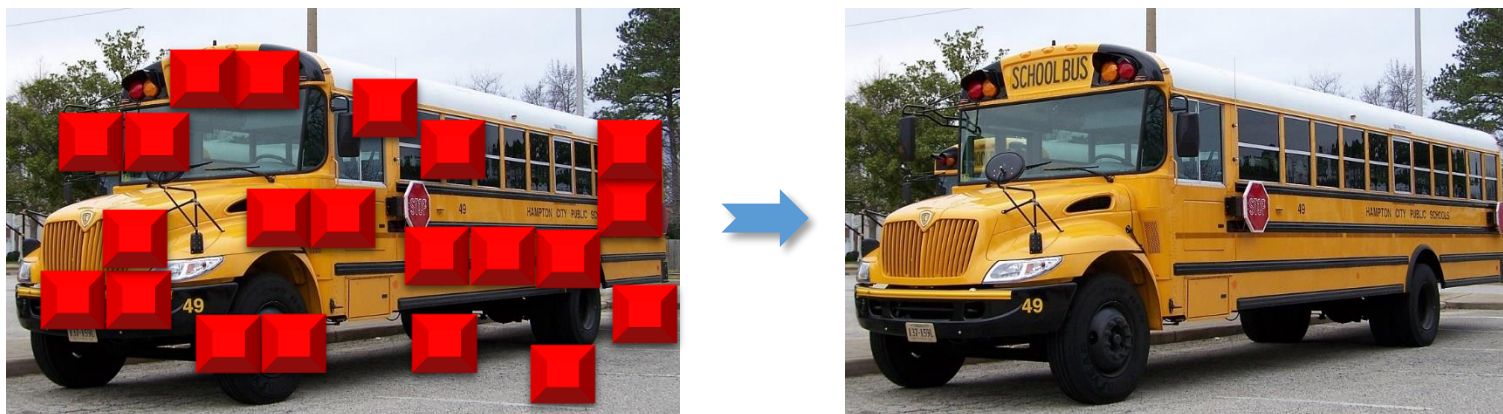


Prompt is the technique of making better use of the knowledge from the pre-trained model by adding additional texts to the input.



Prompt in CV

However, despite significant interest in this idea following the success of BERT, progress of autoencoding methods in vision lags behind NLP.



What makes masked autoencoding different between vision and language?

Masked AutoEncoders

What makes masked autoencoding different between vision and language?

1. Until recently, architectures were different.
2. Information density is different between language and vision.
3. The autoencoder's decoder, which maps the latent representation back to the input, plays a different role between reconstructing text and images

Masked AutoEncoders

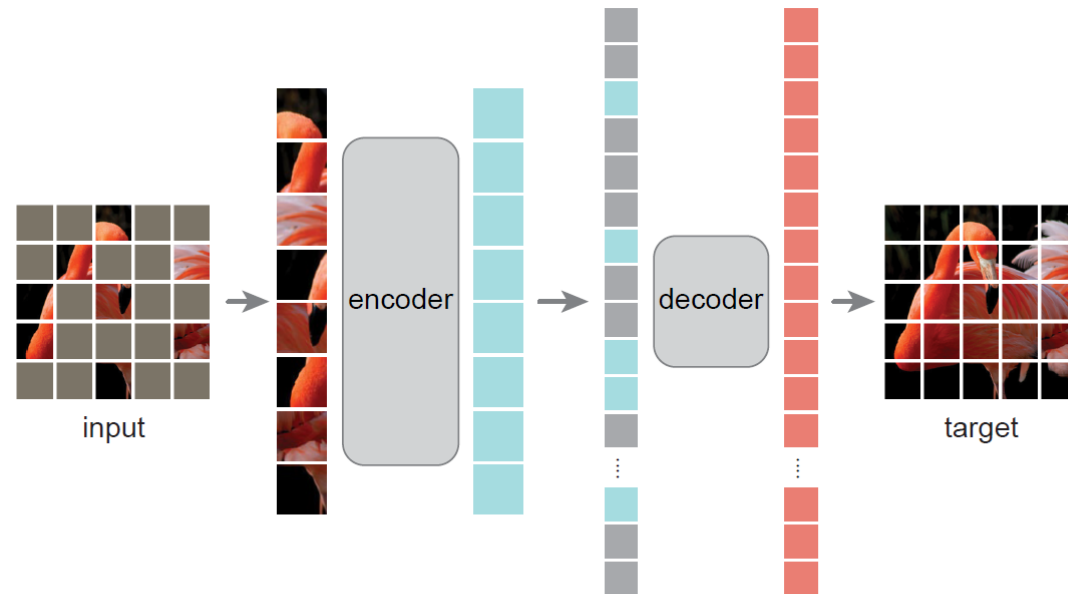
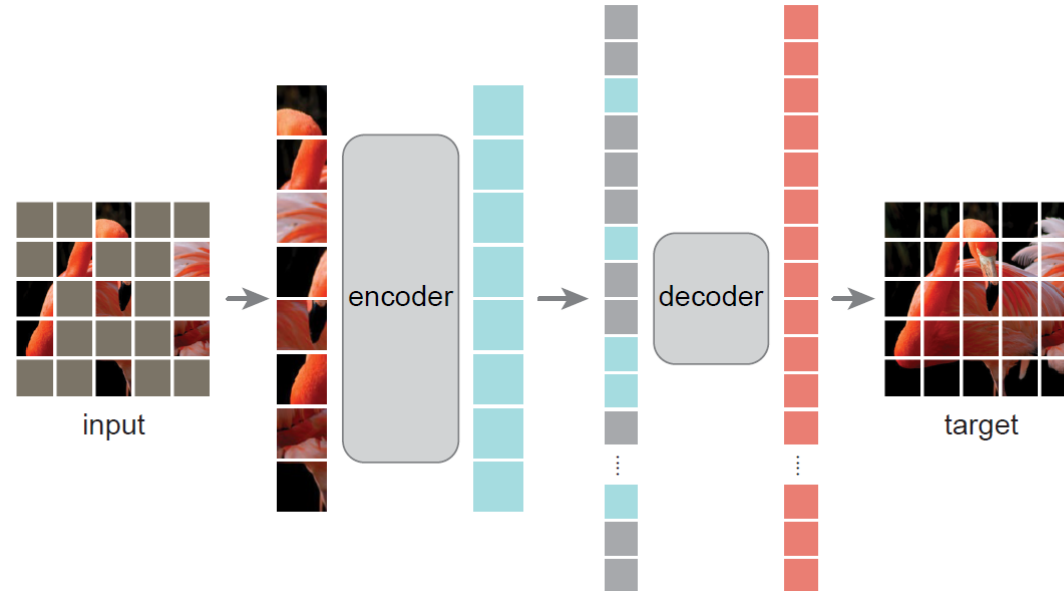


Figure 1. **Our MAE architecture.** During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

Masked AutoEncoders

MAE encoder: Our encoder is a ViT but applied only on visible, unmasked patches.

MAE decoder: The input to the MAE decoder is the full set of tokens consisting of (i) encoded visible patches, and (ii) mask tokens.



Experiments: Baseline

We do self-supervised pre-training on the ImageNet-1K (IN1K) [13] training set. Then we do supervised training to evaluate the representations with (i) end-to-end fine-tuning or (ii) linear probing. We report top-1 validation accuracy of a single 224×224 crop. Details are in Appendix A.1.

Baseline: ViT-Large. We use ViT-Large (ViT-L/16) [16] as the backbone in our ablation study. ViT-L is very big (an order of magnitude bigger than ResNet-50 [25]) and tends to overfit. The following is a comparison between ViT-L trained from scratch vs. fine-tuned from our baseline MAE:

scratch, original [16]	scratch, our impl.	baseline MAE
76.5	82.5	84.9



config	value
optimizer	AdamW
base learning rate	1e-4
weight decay	0.3
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
batch size	4096
learning rate schedule	cosine decay
warmup epochs	20
training epochs	300 (B), 200 (L/H)
augmentation	RandAug (9, 0.5) [12]
label smoothing [45]	0.1
mixup [58]	0.8
cutmix [57]	1.0
drop path [26]	0.1 (B), 0.2 (L/H)
exp. moving average (EMA)	0.9999

Experiments: Masking Ratio

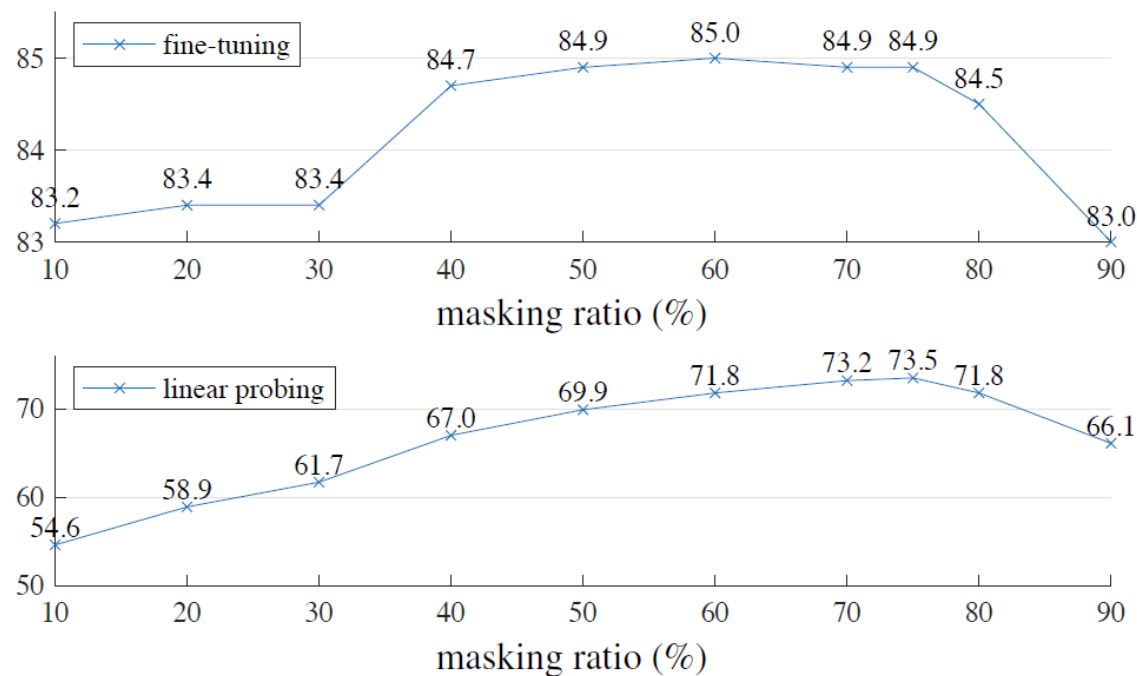


Figure 5. **Masking ratio.** A high masking ratio (75%) works well for both fine-tuning (top) and linear probing (bottom). The y-axes are ImageNet-1K validation accuracy (%) in all plots in this paper.

Ablation Experiments

blocks	ft	lin
1	84.8	65.5
2	84.9	70.0
4	84.9	71.9
8	84.9	73.5
12	84.4	73.3

(a) **Decoder depth.** A deep decoder can improve linear probing accuracy.

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	85.4	73.9
PCA	84.6	72.3
dVAE token	85.3	71.6

(d) **Reconstruction target.** Pixels as reconstruction targets are effective.

dim	ft	lin
128	84.9	69.1
256	84.8	71.3
512	84.9	73.5
768	84.4	73.1
1024	84.3	73.1

(b) **Decoder width.** The decoder can be narrower than the encoder (1024-d).

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

(e) **Data augmentation.** Our MAE works with minimal or no augmentation.

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

(c) **Mask token.** An encoder without mask tokens is more accurate and faster (Table 2).

case	ratio	ft	lin
random	75	84.9	73.5
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

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

Ablation Experiments

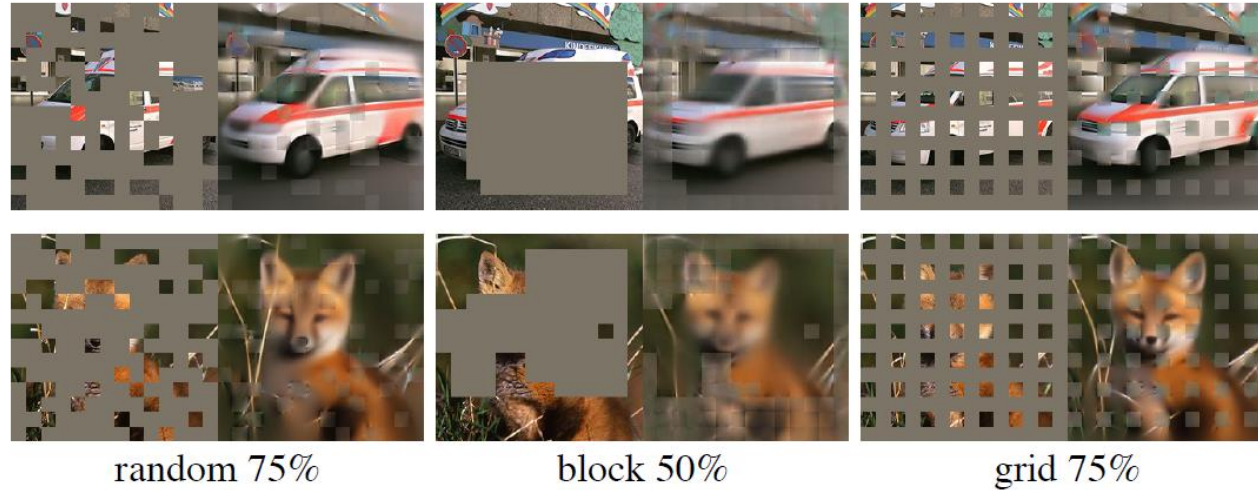
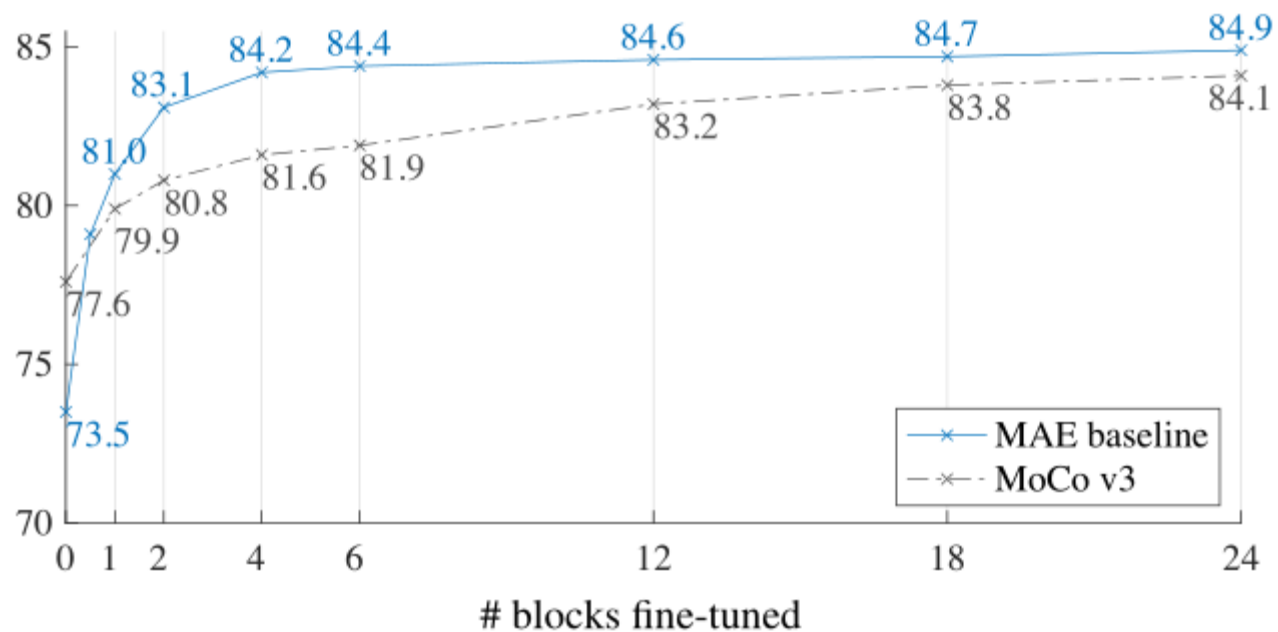


Figure 6. **Mask sampling strategies** determine the pretext task difficulty, influencing reconstruction quality and representations (Table 1f). Here each output is from an MAE trained with the specified masking strategy. Left: random sampling (our default). Middle: block-wise sampling [2] that removes large random blocks. Right: grid-wise sampling that keeps one of every four patches. Images are from the validation set.

Partial Fine-tuning

Partial fine-tuning protocol: fine-tune the last several layers while freezing the others.





03

Multi-modal Learning

多模态学习

Spring 2023

Motivation

Issues:

NLP: Text-to-Text → Zero-shot transfer

CV: Label → **Crowd-Labeled** datasets

Target:

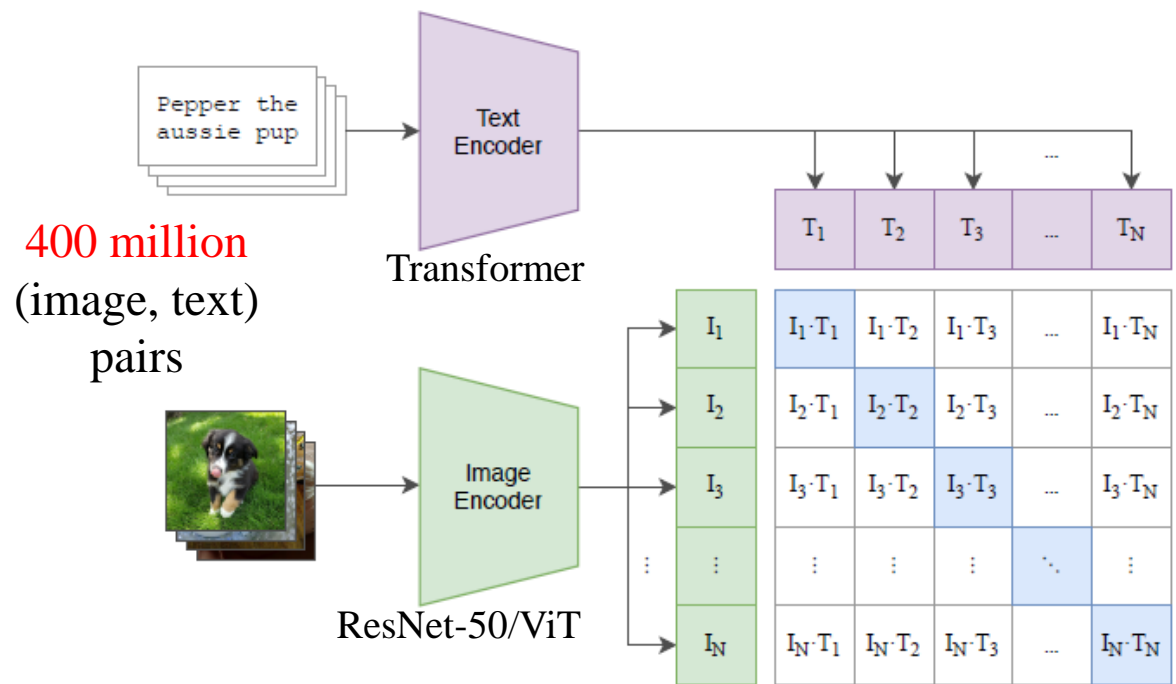
learn directly from web text result in a similar breakthrough in CV

Solution:

CLIP: Contrastive Language-Image Pretraining

Architecture

(1) Contrastive pre-training



Row: one-hot →
Col: One-hot →

```
# 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]
```

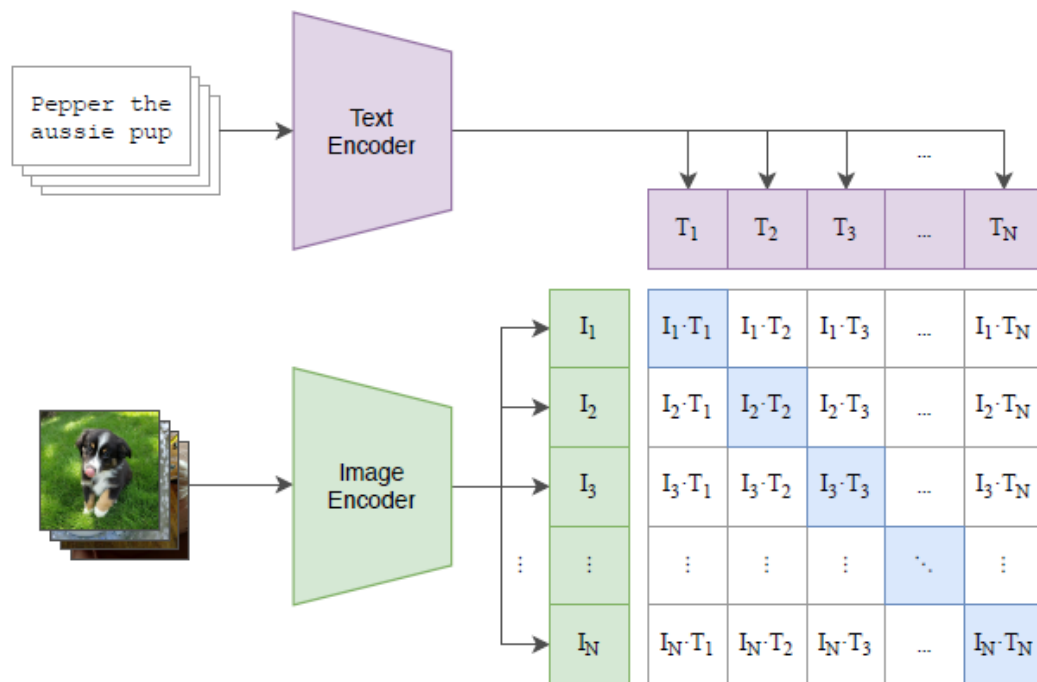
```
# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)
```

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

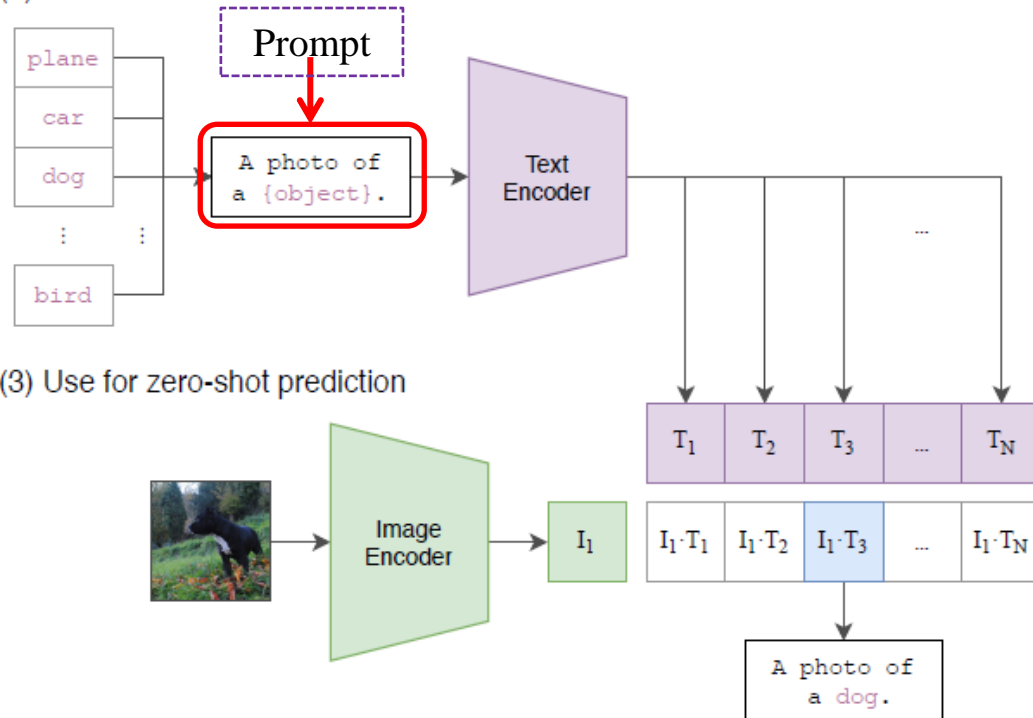
```
# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss = (loss_i + loss_t)/2
```

Architecture

(1) Contrastive pre-training



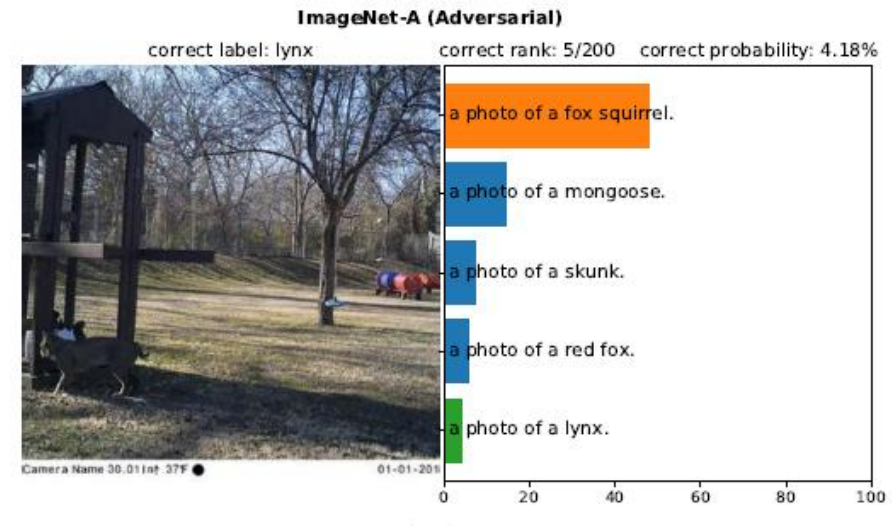
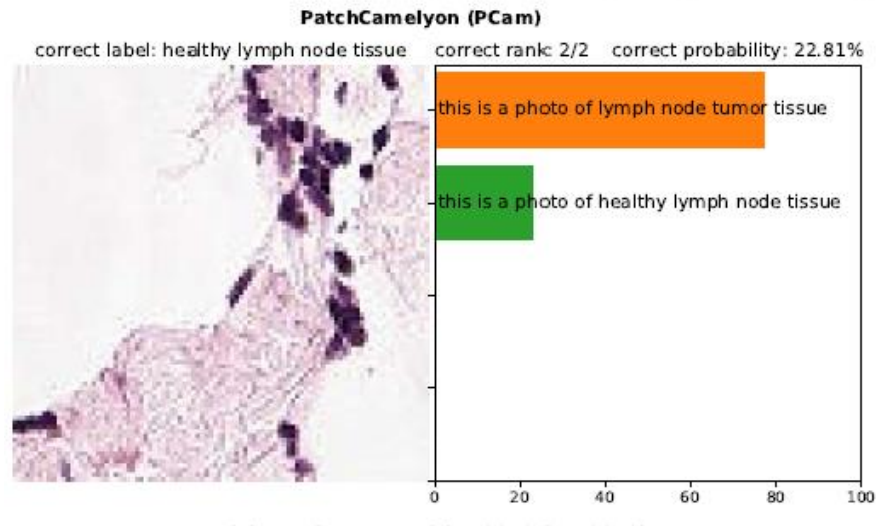
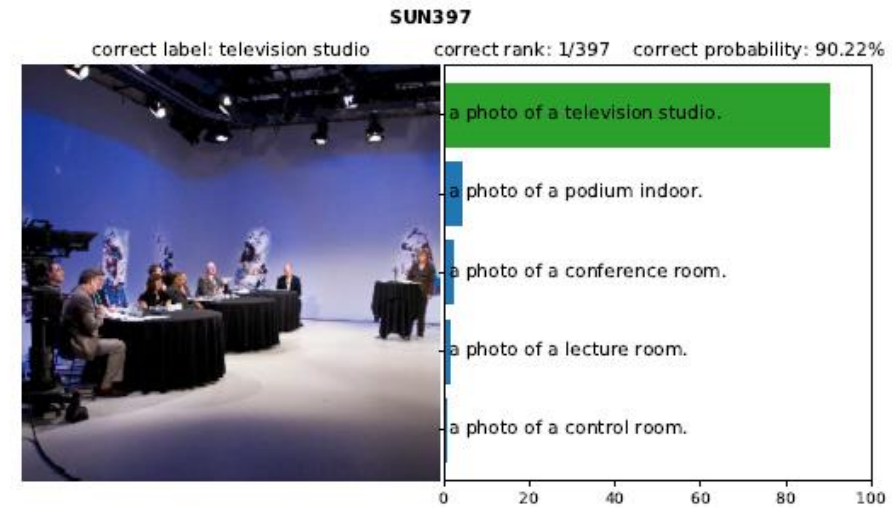
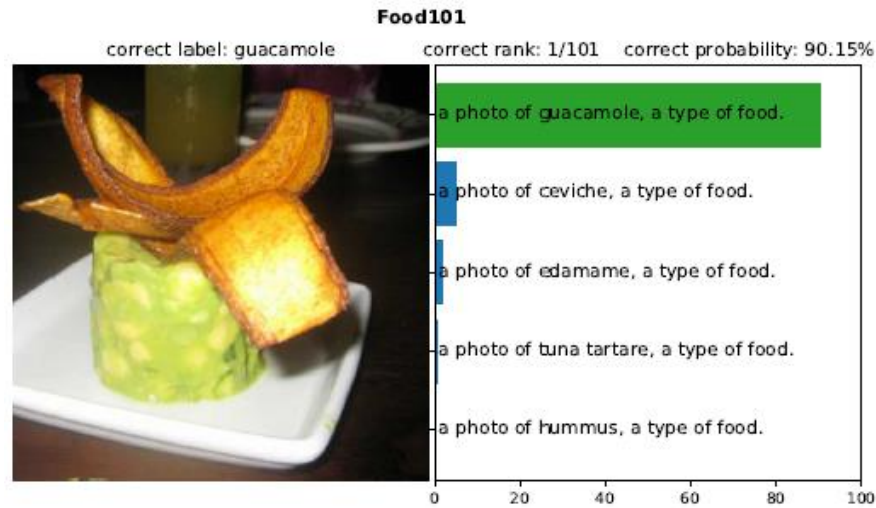
(2) Create dataset classifier from label text



(3) Use for zero-shot prediction

Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Predictions



Visualization of Predictions

Architecture

The largest ResNet model, RN50x64, took *18 days* to train on **592 V100** GPUs;
the largest Vision Transformer took *12 days* on **256 V100** GPUs.

Model	Learning rate	Embedding dimension	Input resolution	ResNet blocks	width	Text Transformer layers	width	heads
RN50	5×10^{-4}	1024	224	(3, 4, 6, 3)	2048	12	512	8
RN101	5×10^{-4}	512	224	(3, 4, 23, 3)	2048	12	512	8
RN50x4	5×10^{-4}	640	288	(4, 6, 10, 6)	2560	12	640	10
RN50x16	4×10^{-4}	768	384	(6, 8, 18, 8)	3072	12	768	12
RN50x64	3.6×10^{-4}	1024	448	(3, 15, 36, 10)	4096	12	1024	16


Table 19. CLIP-ResNet hyperparameters

Model	Learning rate	Embedding dimension	Input resolution	Vision Transformer layers	width	heads	Text Transformer layers	width	heads
ViT-B/32	5×10^{-4}	512	224	12	768	12	12	512	8
ViT-B/16	5×10^{-4}	512	224	12	768	12	12	512	8
ViT-L/14	4×10^{-4}	768	224	24	1024	16	12	768	12
ViT-L/14-336px	2×10^{-5}	768	336	24	1024	16	12	768	12

Table 20. CLIP-ViT hyperparameters

Hyperparameter	Value
Batch size	32768
Vocabulary size	49408
Training epochs	32
Maximum temperature	100.0
Weight decay	0.2
Warm-up iterations	2000
Adam β_1	0.9
Adam β_2	0.999 (ResNet), 0.98 (ViT)
Adam ϵ	10^{-8} (ResNet), 10^{-6} (ViT)

Table 18. Common CLIP hyperparameters



NVIDIA Tesla V100 Volta GPU
Accelerator 32GB Graphics Card
Brand Nvidia Corporation
★★★★☆ 4 ratings | 9 answered questions
Price \$9,999.00
Pay \$555.50/month for 18 months, interest-free upon approval for the Amazon Rewards Visa Card

Brand Nvidia Corporation
Graphics Coprocessor NVIDIA Tesla V100
Graphics Processor NVIDIA
Manufacturer
Graphics Ram Size 32 GB
GPU Clock Speed 1380 MHz

Compare with similar items
New (2) from \$9,999.00 & FREE Shipping

Performance

Zero-shot CLIP: 16 out of 27

(3) Use for zero-shot prediction

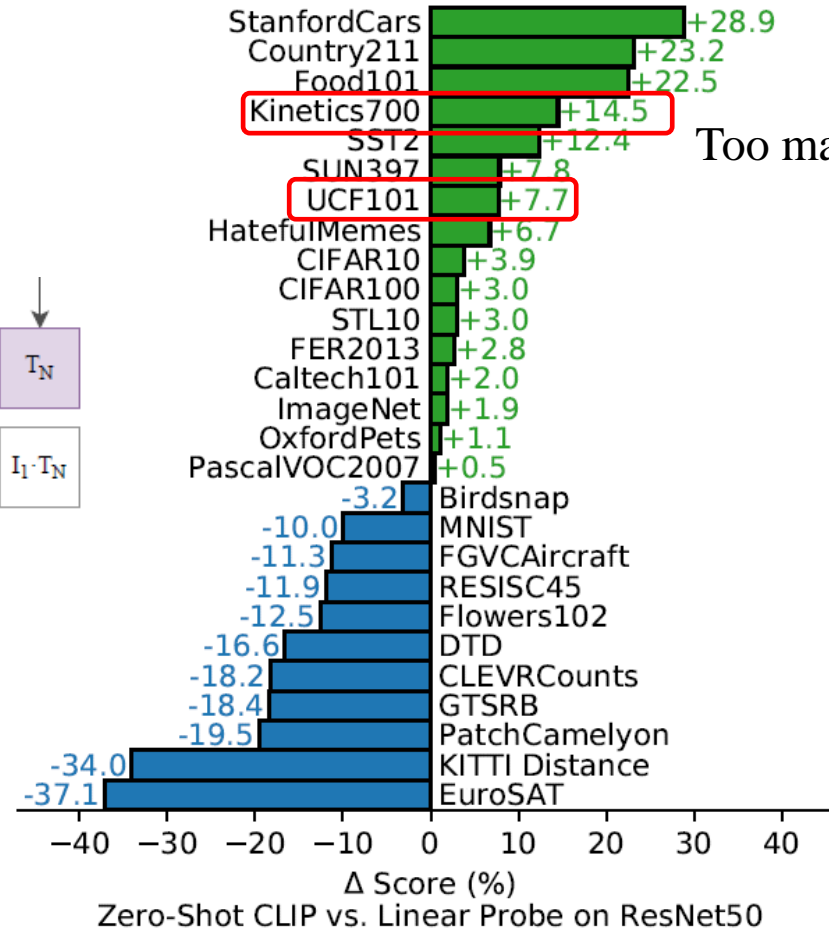
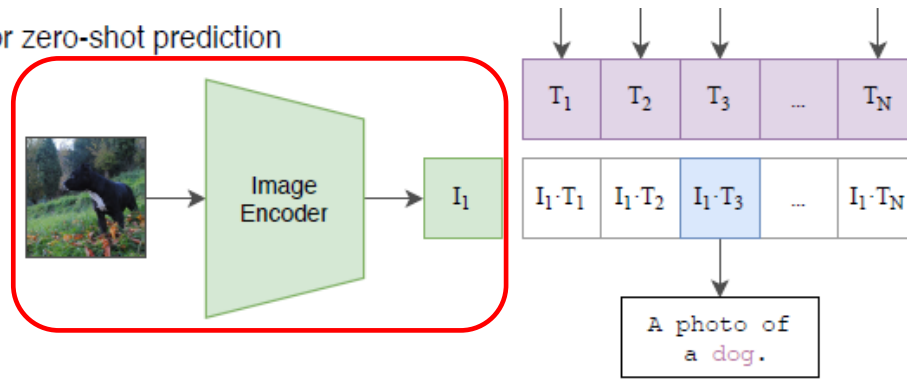


Figure 5. Zero-shot CLIP is competitive with a fully supervised baseline. Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

Performance

(3) Use for zero-shot prediction

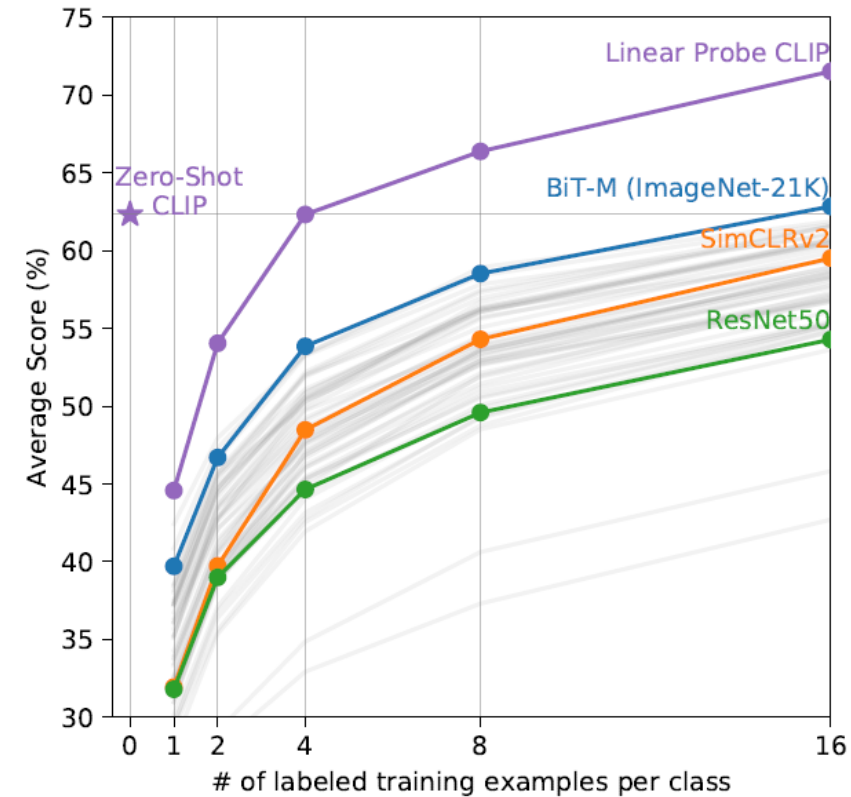
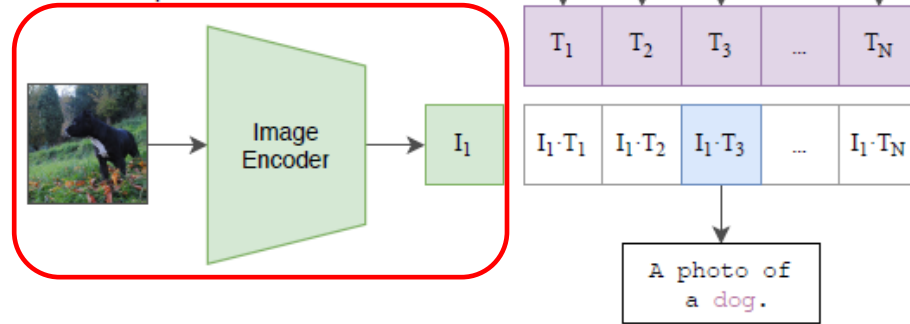
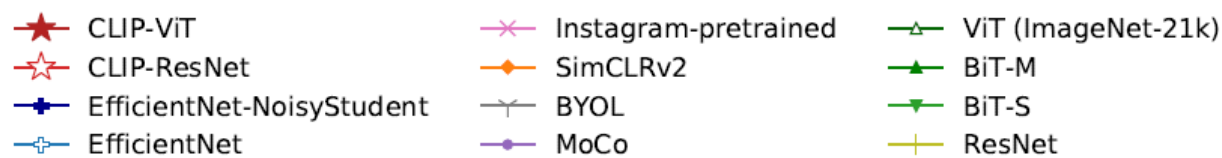
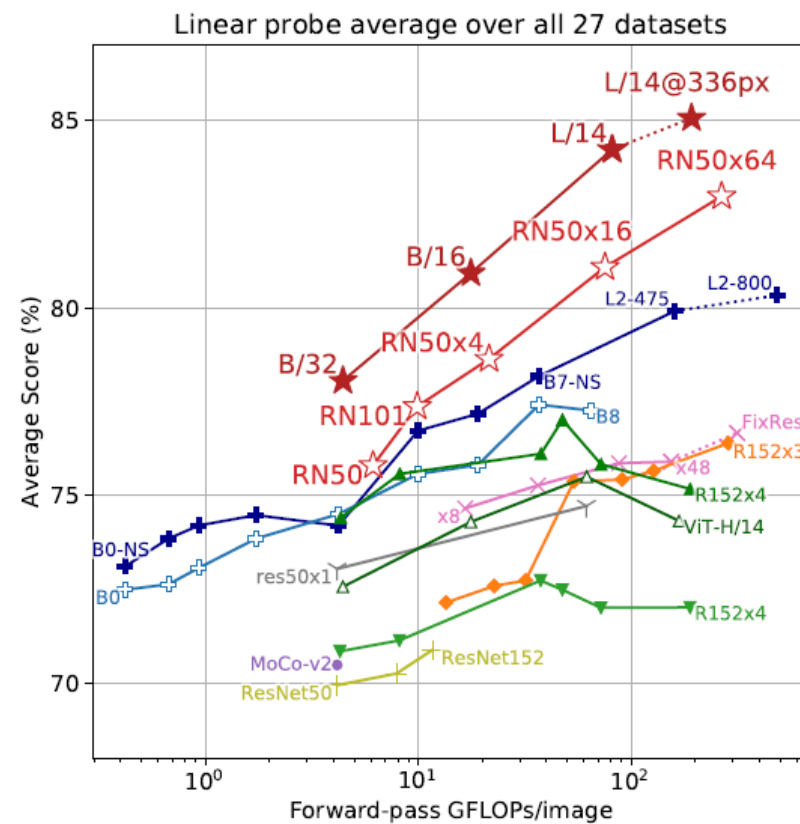
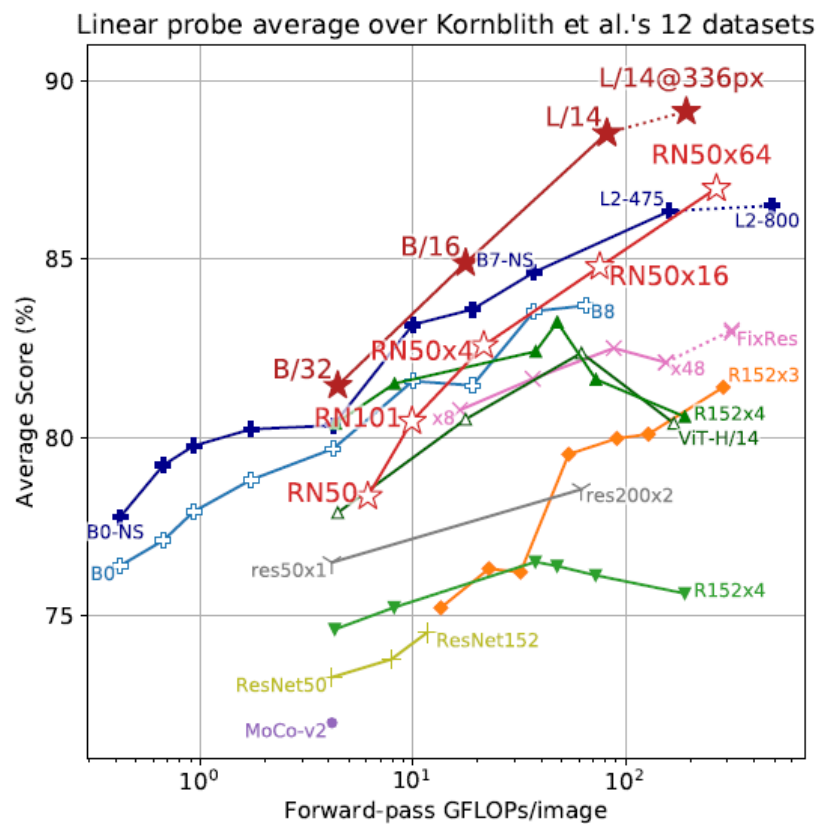


Figure 6. Zero-shot CLIP outperforms few-shot linear probes. Zero-shot CLIP matches the average performance of a 4-shot linear classifier trained on the same feature space and nearly matches the best results of a 16-shot linear classifier across publicly available models. For both BiT-M and SimCLRv2, the best performing model is highlighted. Light gray lines are other models in the eval suite. The 20 datasets with at least 16 examples per class were used in this analysis.

Performance



Performance

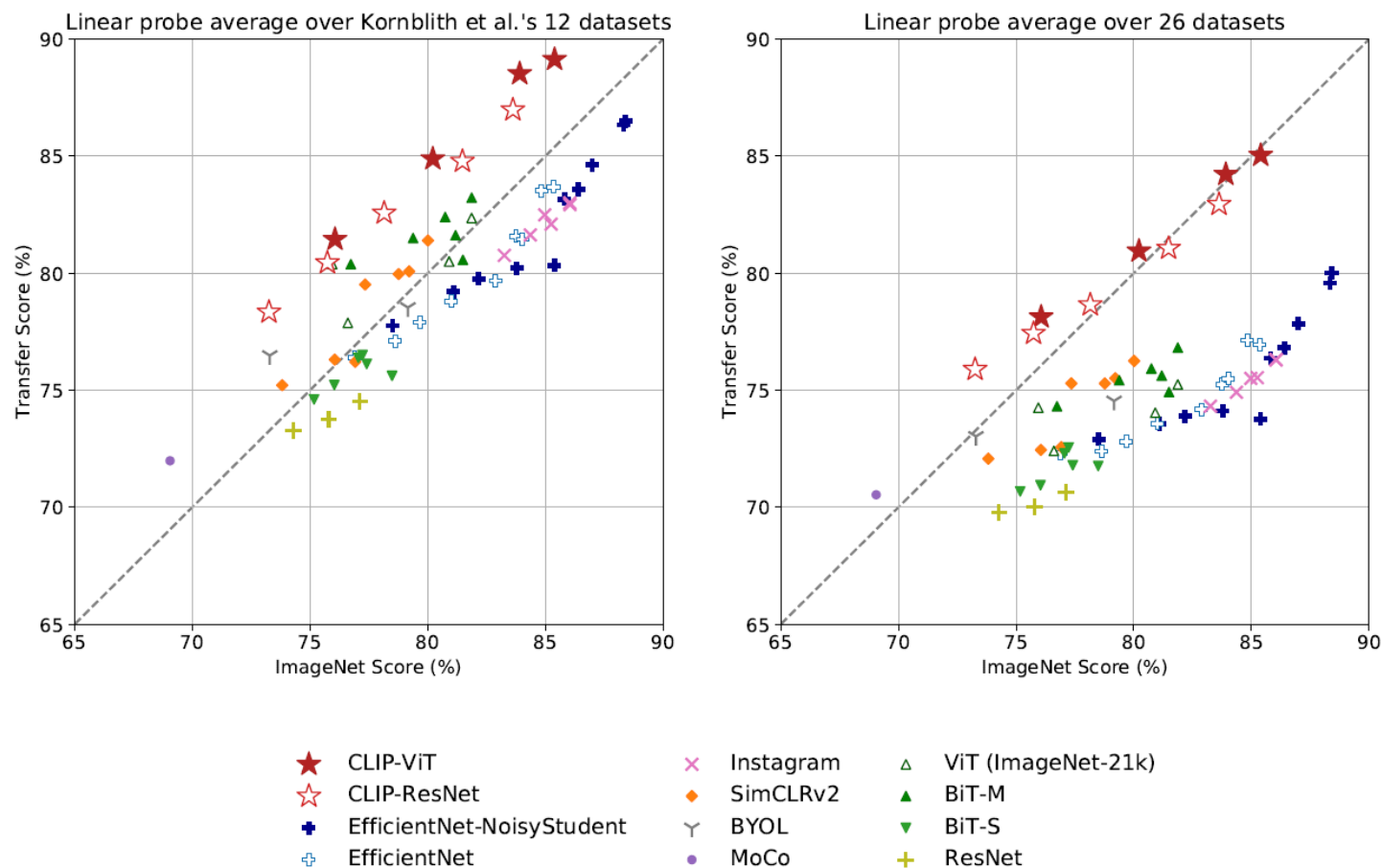
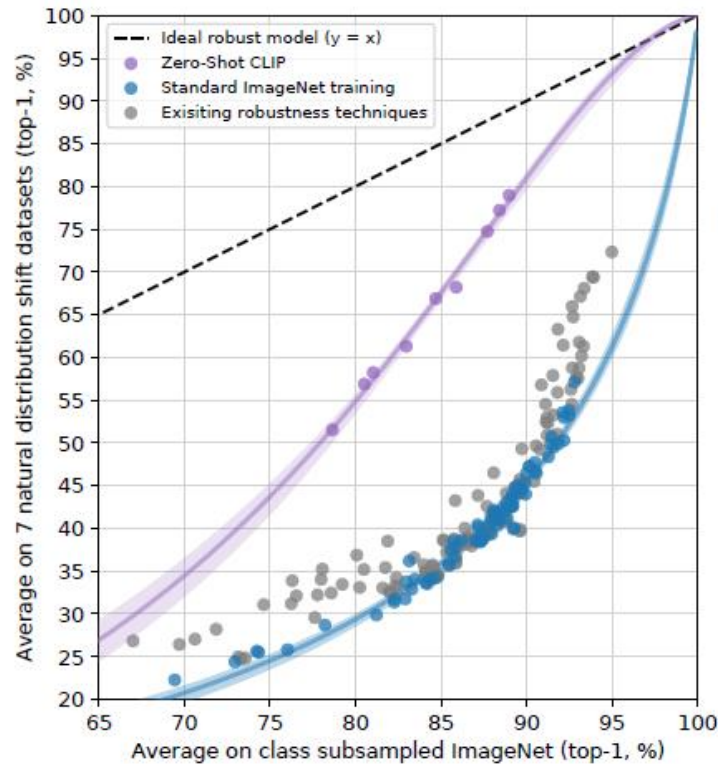


Figure 12. CLIP's features are more robust to task shift when compared to models pre-trained on ImageNet. For both dataset splits, the transfer scores of linear probes trained on the representations of CLIP models are higher than other models with similar ImageNet performance. This suggests that the representations of models trained on ImageNet are somewhat overfit to their task.

Performance



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%	

Figure 13. Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models. (Left) An ideal robust model (dashed line) performs equally well on the ImageNet distribution and on other natural image distributions. Zero-shot CLIP models shrink this “robustness gap” by up to 75%. Linear fits on logit transformed values are shown with bootstrap estimated 95% confidence intervals. (Right) Visualizing distribution shift for bananas, a class shared across 5 of the 7 natural distribution shift datasets. The performance of the best zero-shot CLIP model, ViT-L/14@336px, is compared with a model that has the same performance on the ImageNet validation set, ResNet-101.

Q&A



Spring 2023