2024-11-02

Machine Learning

Lecture 10: Introduction to Vision Transformers and Vision Language Models

Nevin L. Zhang

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology

This set of notes is based on inputs from Weiyan Xie and internet resources.

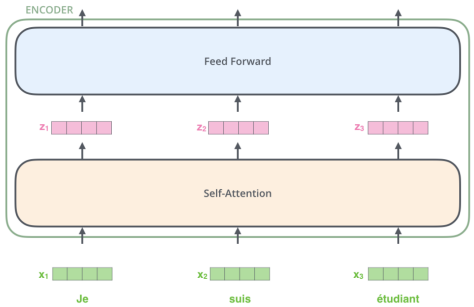
Outline

Text encoder + Image encoder

Bridging text & image ⇒ Vision language mode

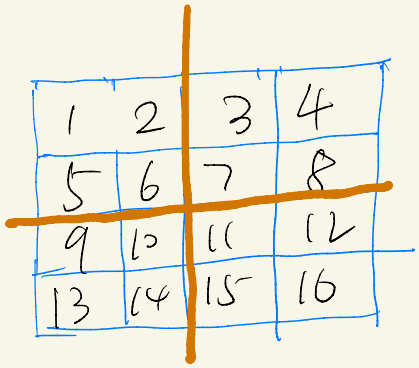
Vision Transformer (ViT)[1]

- An image is a 3D tensor of pixel values. Self-attention layer maps a collection of vectors to another collection of vectors.



How to apply self-attention to images?

[1] Dosovitskiy *et al* 2021, AN IMAGE IS WORTH 16X16 WORDS ...



(1, 2, 5, 6)

(3, 4, 7, 8)

(9, 10, 13, 14)

(11, 12, 15, 16)

Image ⟹ patches ⟹ vectors

H x W

4 x 4 x C

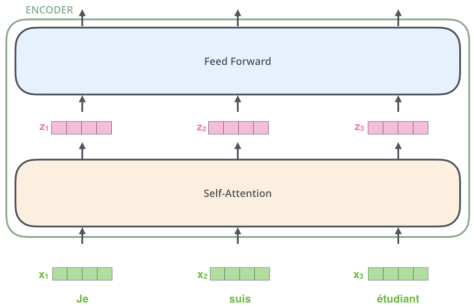
flattened into

p x p : 2 x 2

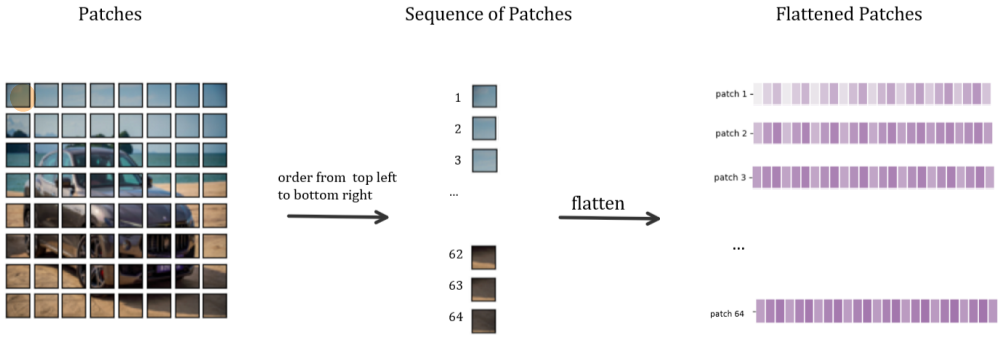
$\frac{4 \times 4}{2 \times 2} = 4$

Vision Transformer (ViT)[1]

- An image is a 3D tensor of pixel values. Self-attention layer maps a collection of vectors to another collection of vectors.

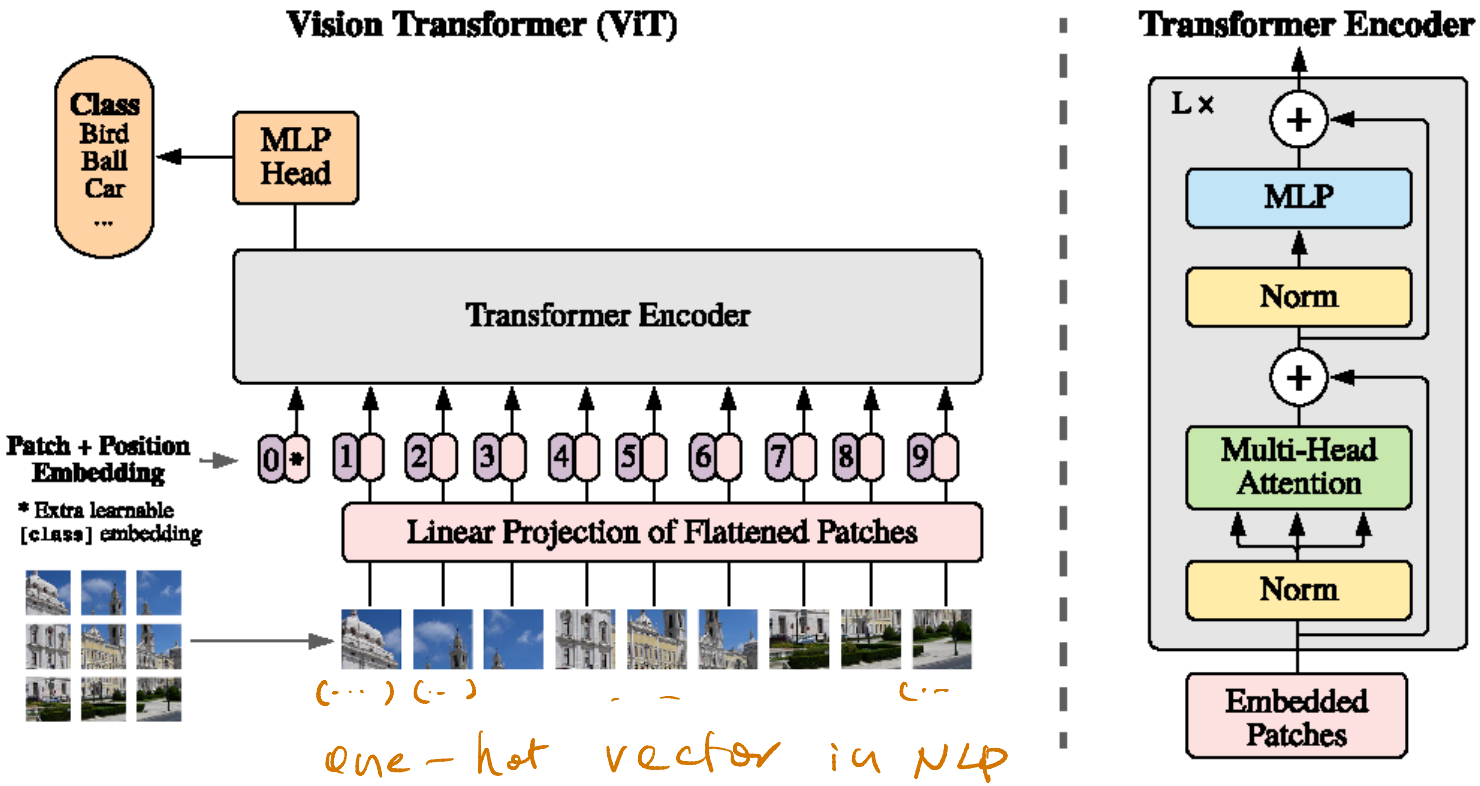


- Split an image into fixed-size (16x16) patches, and flatten them into vectors



[1] Dosovitskiy *et al* 2021, AN IMAGE IS WORTH 16X16 WORDS ...

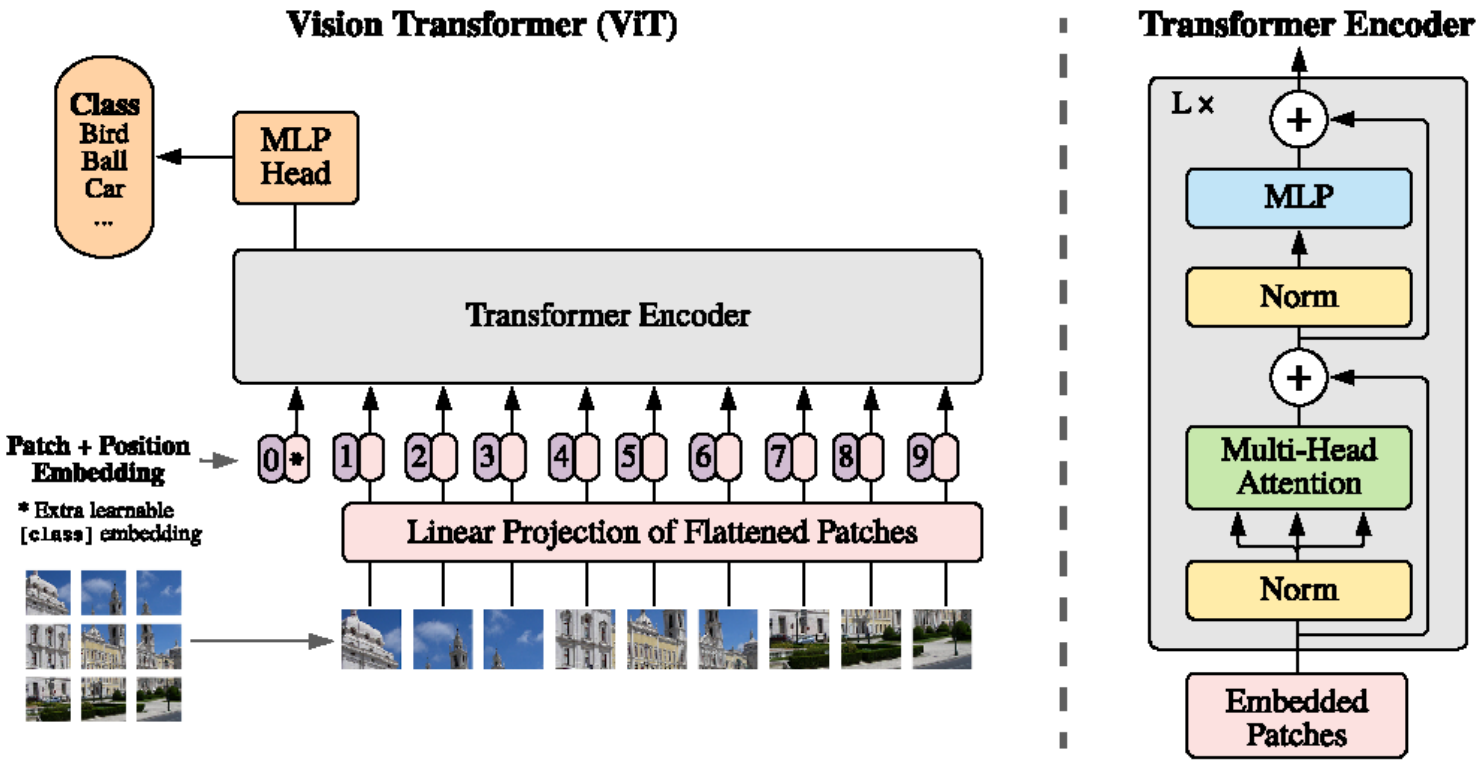
Vision Transformer (ViT)



- The flattened patches are passed through a linear project layer to obtain initial patch embeddings (akin to word embedding)
- Position embeddings are then added, and the resulting sequence is fed to a standard Transformer encoder.
- Unlike in NLP, position embeddings in ViTs are learned.

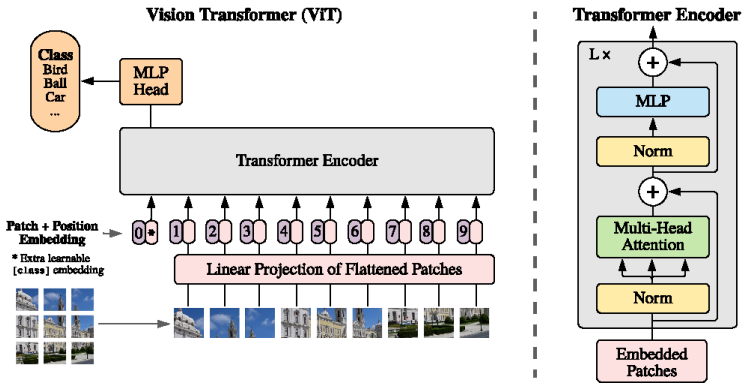
can also be manually set

Vision Transformer (ViT)



- In order to perform classification, an extra learnable "classification token" is added to the sequence.

Vision Transformer (ViT)



$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{1}$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \qquad \ell = 1 \ldots L \tag{3}$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \tag{4}$$

Input: X — $H \times W \times C$

patch size: $P \times P$

# of patches: $\frac{HW}{P^2}$

flatten patch x_p: $1 \times P^2C$

Linear projection: E $P^2C \times D$

$x_p E$: $1 \times D$

x_{class}: $1 \times D$

$$z_0 = \begin{bmatrix} x_{class} \\ x_p^1 E \\ \vdots \\ x_p^N E \end{bmatrix}$$

$(N+1) \times D$

$+ E_{pos}$: $(N+1) \times D$

$z_0 \to z'_1 \to z_1 \to z'_2 \to z_2$

$\ldots \to z_L \to y$

Multi-head Self-Attention

layer Norm : pre-layer Norm

Image high dimension

$$\mathbf{z}'_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1},$$

standard Transformer architecture :

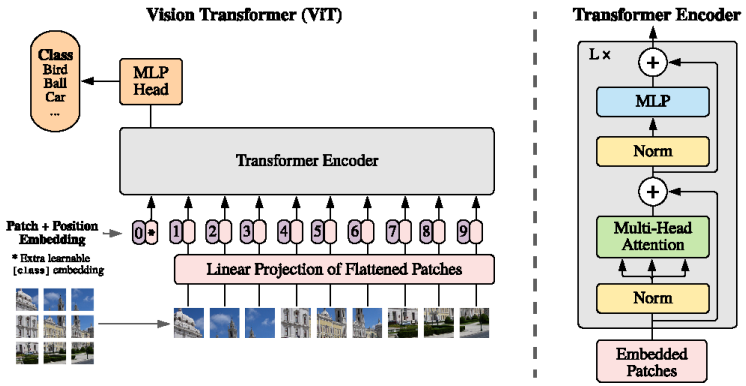
post-layer Norm

residual connection

$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell,$$

para's shared by all patch vectors

Vision Transformer (ViT)



$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{1}$$

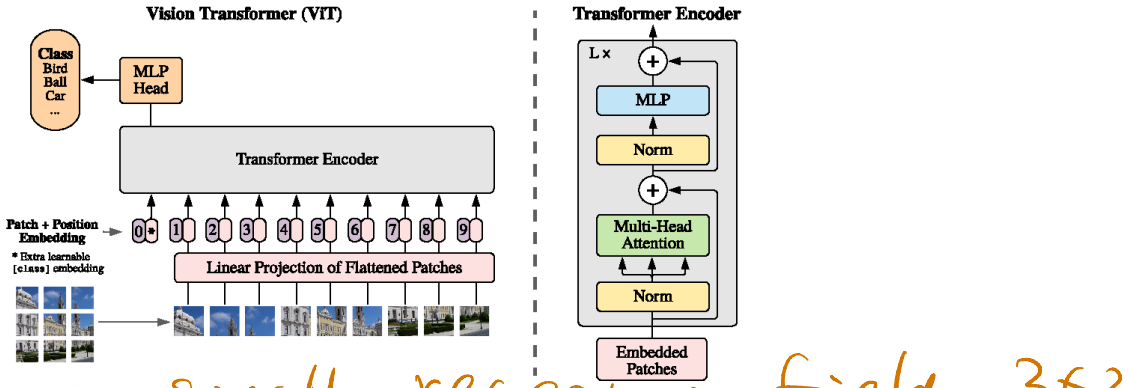
$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \qquad \ell = 1 \ldots L \tag{3}$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \tag{4}$$

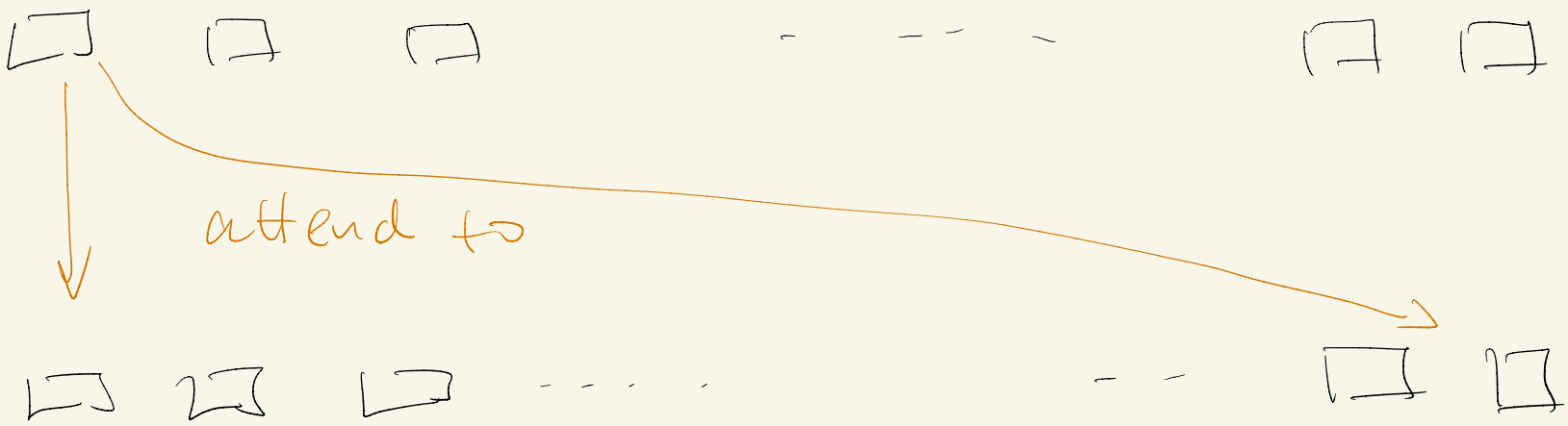
- Input $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ is split into $N = HW/P^2$ patches of resolution $P \times P$
- Each path $\mathbf{x}_p$ is flattened into vector, $\mathbf{x}_p \in \mathbb{R}^{1 \times (P^2 \cdot C)}$, and is multiplied with $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$ to become a D-dimensional vector
- $\mathbf{E}$, $\mathbf{E}_{pos}$ and class token embedding $\mathbf{x}_{class}$ are learnable.
- MSA: Multihead self-attention; LN: Layernormalization
- MLP contains two layers with GELU non-linearity.

ViT has Less Image-Specific Inductive Bias

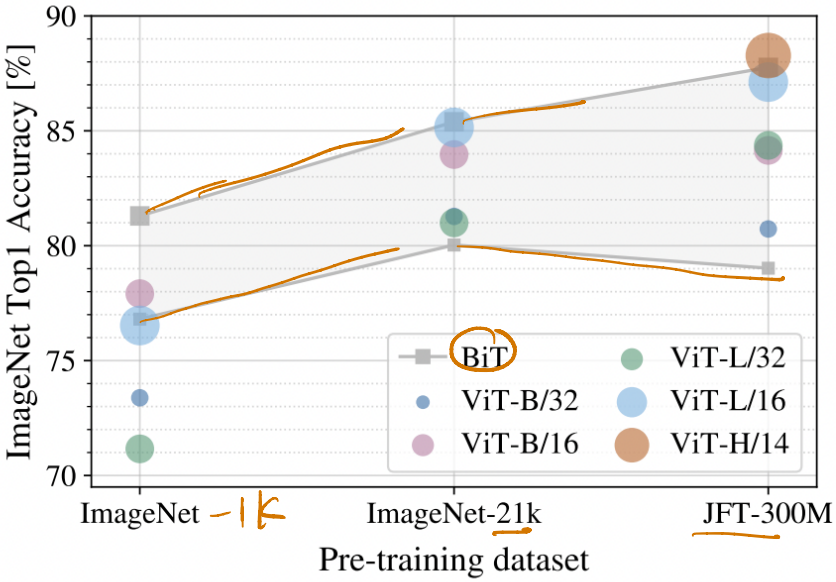


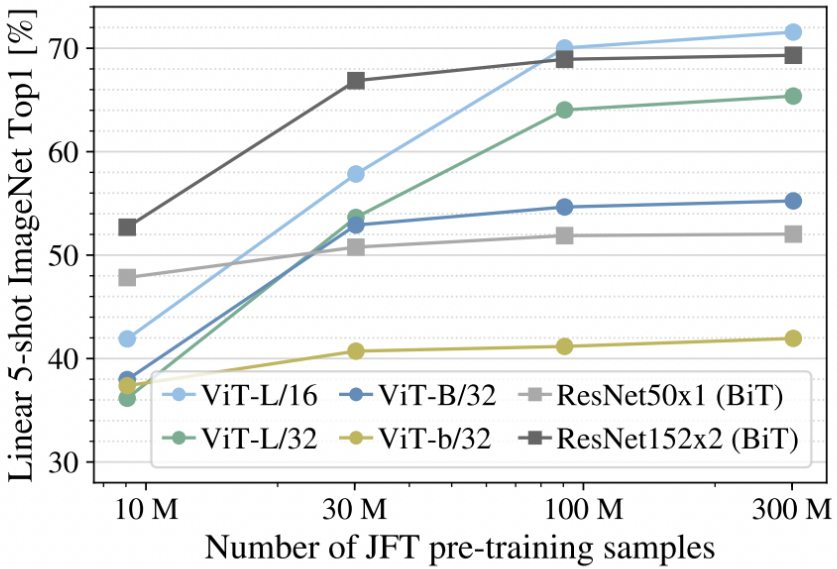
each kernel has small receptive field 3×3

- In CNNs, **locality**, **two-dimensional neighborhood structure**, and **translation equivariance** are baked into each layer throughout the whole model. same kernel applied to diff location
- In ViTs, only MLP layers are local and translationally equivariant, while the self-attention layers are global.
- As they make weak assumptions, ViTs generally need more data to train than CNNs.


attend to

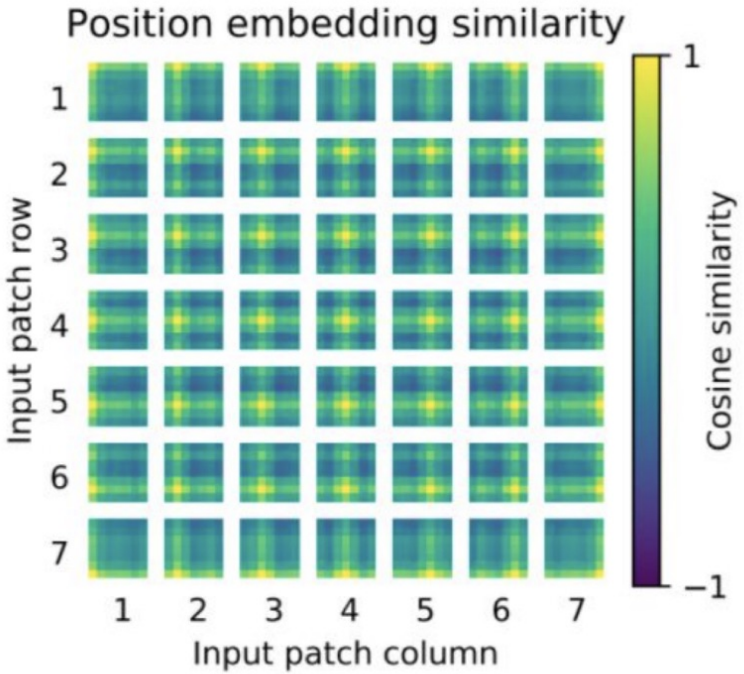
ViT has Less Image-Specific Inductive Bias





- Left: While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.
- Right: ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training.

Learning Patterns of ViT



16 x 16 patches

Here: show 7x7 patches (subset)

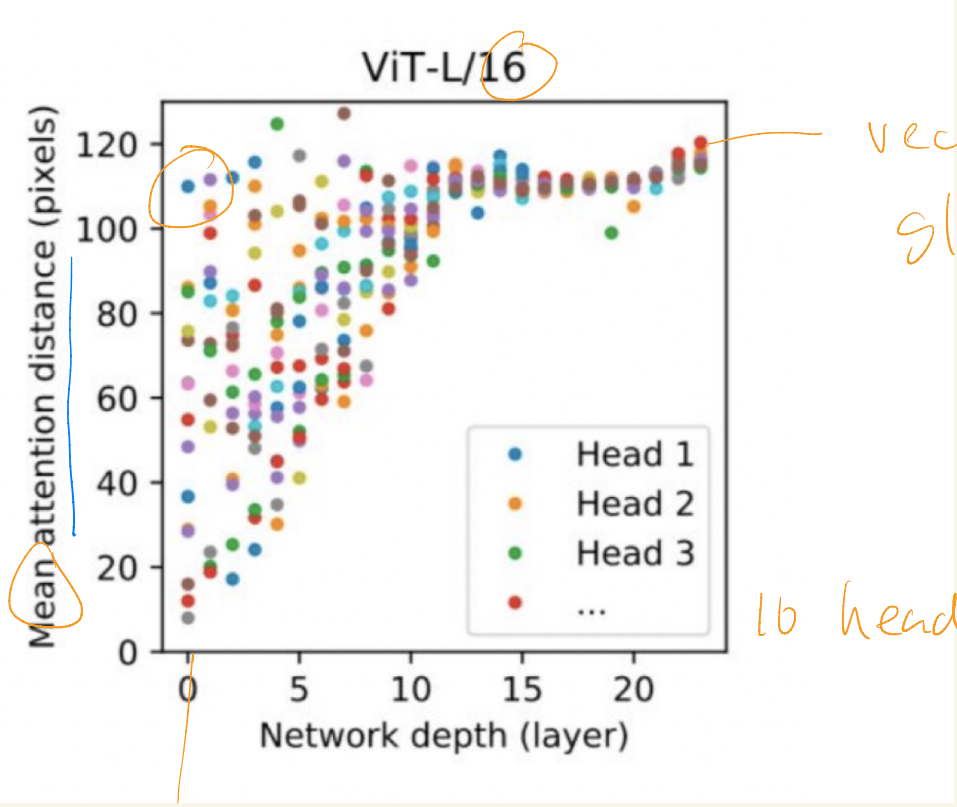
For each path at (i, j)

cosine(E_{ij}, $E_{i'j'}$)

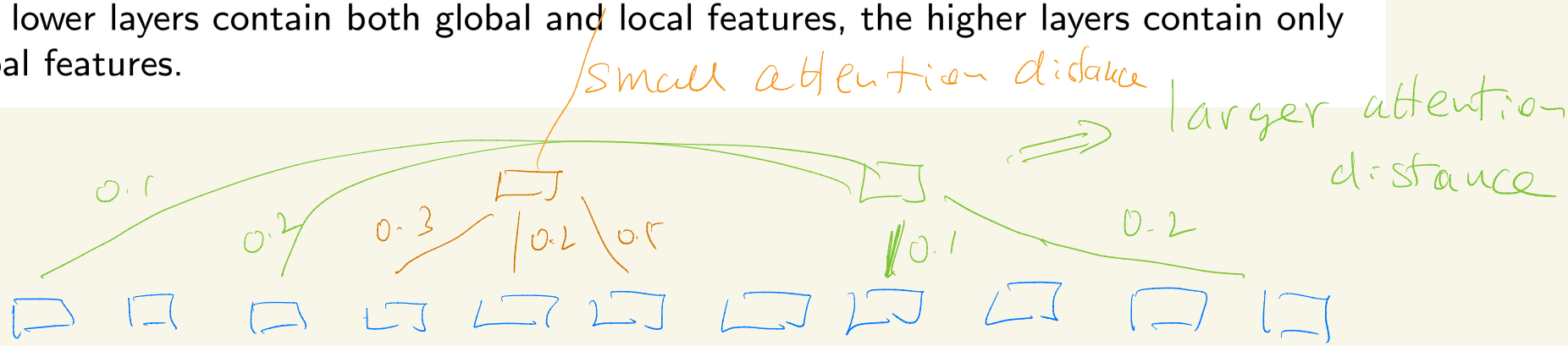
$i', j' \in \{1, \ldots 7\}$

- ViT learns the grid like structure of the image patches via its position embeddings. Each position embedding is most similar to others in the same row and column, indicating that the model has recovered the grid structure of the original images.

The attention distance was computed for 128 example images by averaging the distance between the query pixel and all other pixels, weighted by the attention weight.

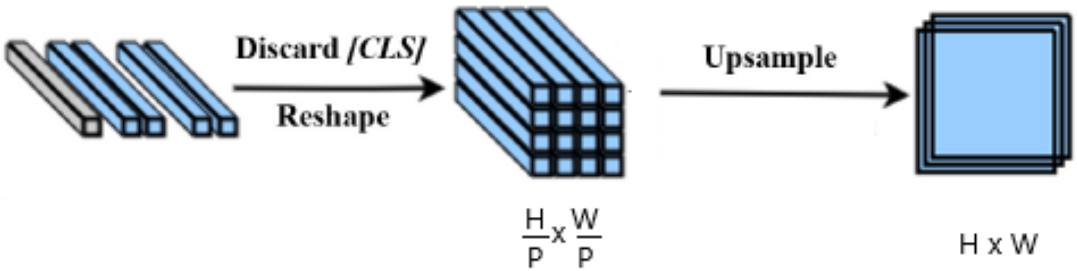


- The lower layers contain both global and local features, the higher layers contain only global features.



ViT Feature Maps

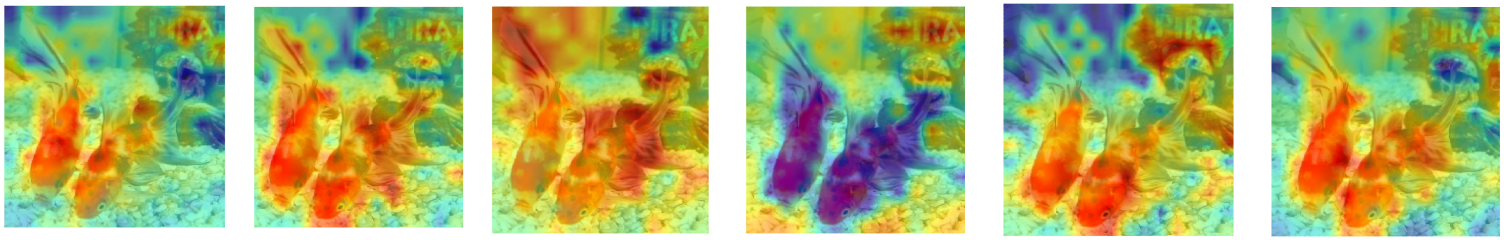
ViT regards an image as a collect of patches. We might ask what feature it extracts in the pixel space.



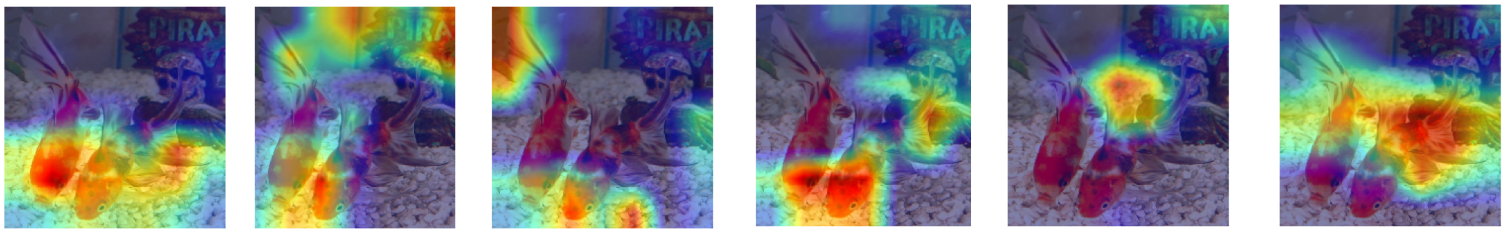
- Patch embeddings at a self-attention layer can be arranged into a 3D tensor, with the (x; y)-coordinates indicating their spatial information and the z-coordinate representing their semantic contents.
- A frontal slice (with a fixed z value) of the tensor can be upsampled to the size of the input image and the resulting heatmap is known as a ViT feature map.

ViT feature maps are generally more meaningful than ResNet feature maps.

Feature maps randomly sampled from those of ViT-B/16

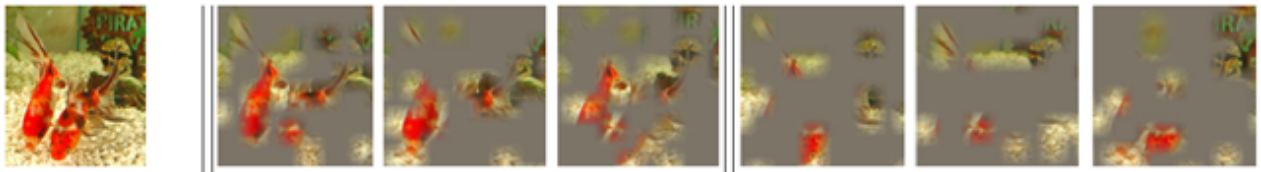


Feature maps randomly sampled from those of ResNet50



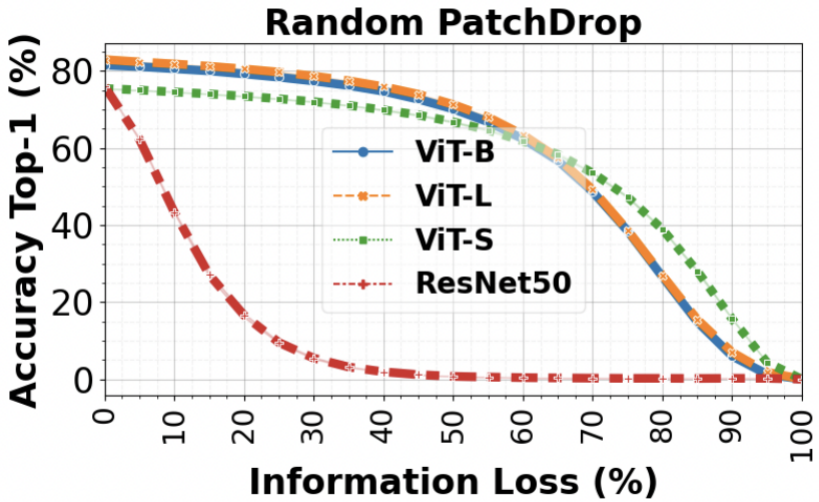
more interpretable

ViTs vs. CNNs: ViTs are More Robust than CNNs



ResNet50	P(goldfish)=0.947	0.998	0.997	0.996	<u>0.149</u>	<u>0.091</u>	<u>0.179</u>
ViT	P(goldfish)=0.989	0.998	0.996	0.998	0.980	0.951	0.978

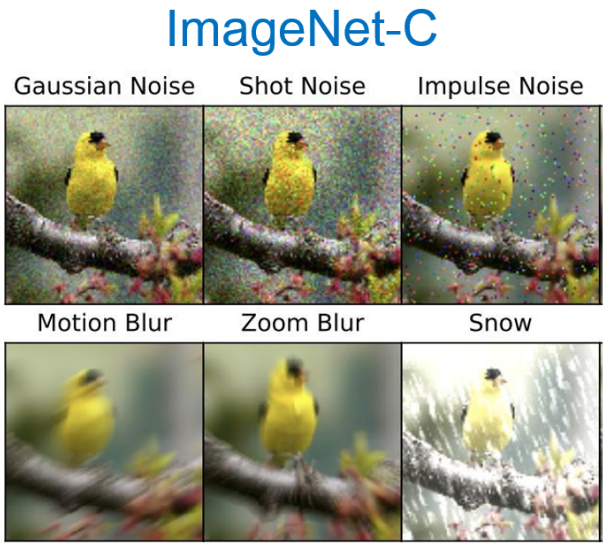
Fig. 3. Example to compare the situation of overdetermination between ResNet50 and ViT (The <u>underline</u> probabilities: the low probabilities for the masked images produced by ResNet50 when ViT still gives high prediction probabilities).

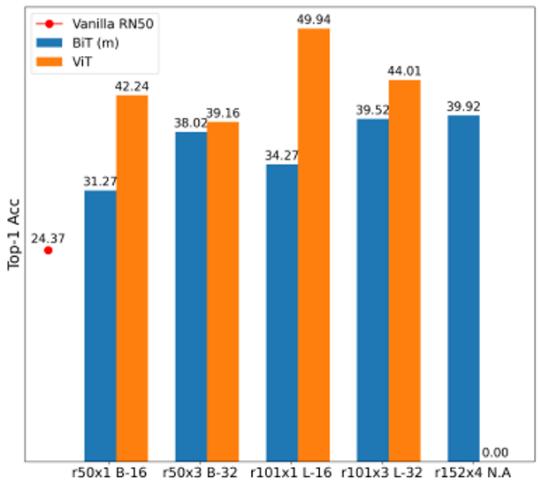


Naseer, et al. Intriguing properties of vision transformers. 2021.

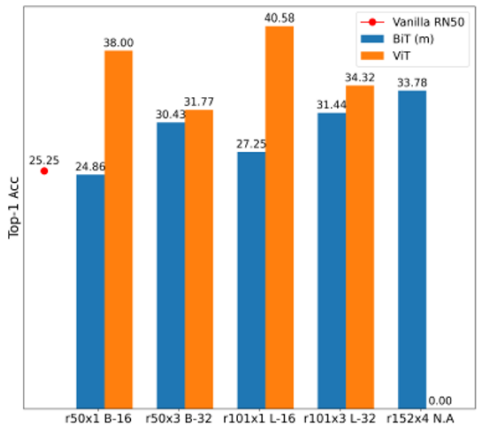
ViTs vs. CNNs: ViTs are More Robust than CNNs

ViT more robust to distribution /domain shift



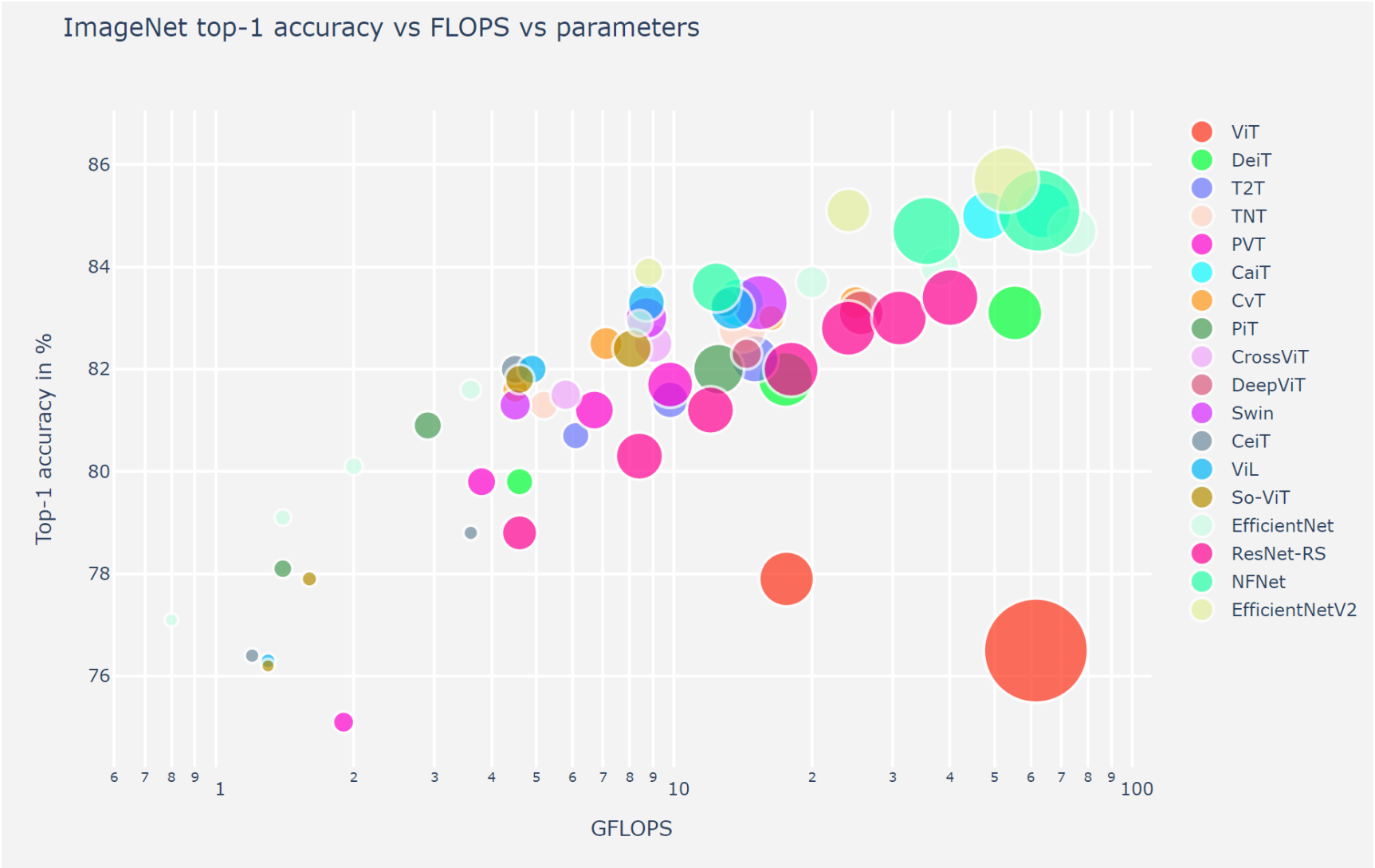


out-of-distribution (OOD) data



Paul, Sayak, and Pin-Yu Chen. Vision transformers are robust learners. 2022.

Other ViT Models



https://iaml-it.github.io/posts/2021-04-28-transformers-in-vision/

Outline

CLIP - Contrastive Language-Image Pre-training [2]

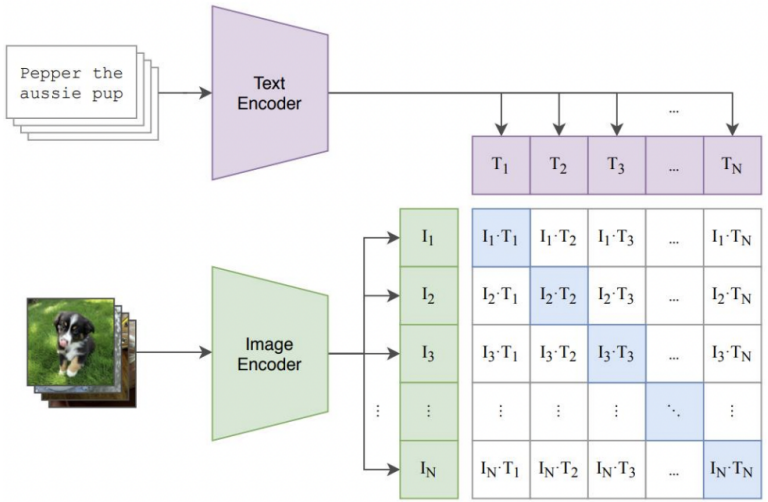
- ResNet and ViT: Trained on labeled images
- Transformer, BERT, GPT: Trained on texts
- CLIP: Trained on 400 million image-text pairs

It is one type of **vision-language models (VLMs)**

[2] Radford, Alec, et al, Learning Transferable Visual Models From Natural Language Supervision, ICML 2021.

CLIP - Contrastive Language-Image Pre-training

Batch Size of N




```
# 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
```

- The CLIP contrastive loss function (for each batch with N pairs):

Fixed I_i

$I_i \cdot T_1$

$I_i \cdot T_2$

$\vdots$

$I_i \cdot T_N$

$e^{I_i \cdot T_1} / \sum_{j=1}^{N} e^{I_i \cdot T_j}$

$e^{I_i \cdot T_2}$ ...

$\vdots$

$e^{I_i \cdot T_N} / \sum_{j=1}^{N} e^{I_i \cdot T_j}$

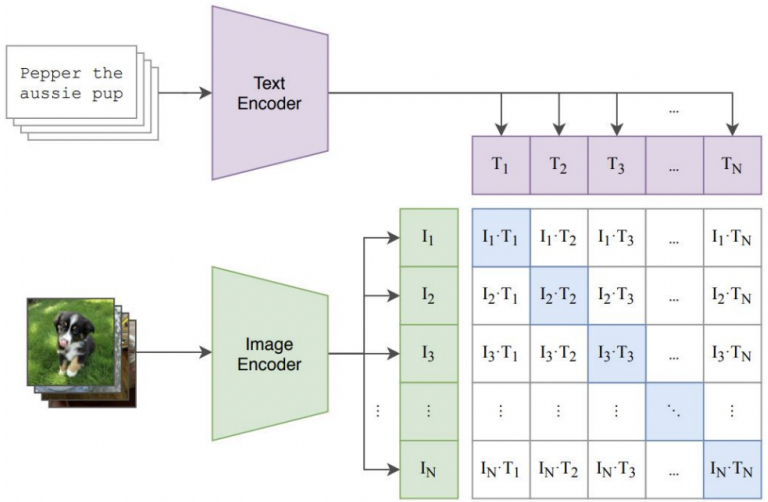
probability dist

T_1	0.1
T_2	0.2
$\vdots$	
T_N	0.5
	+ 1.00

inner product

$I_i^T T_j$

CLIP - Contrastive Language-Image Pre-training




```
# 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
```

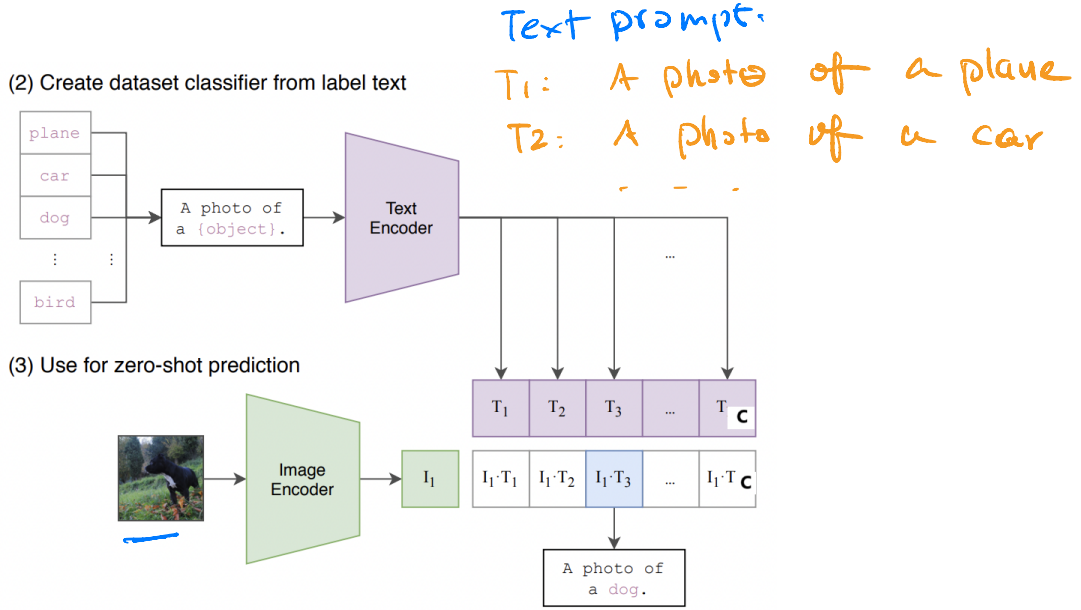
- The CLIP contrastive loss function (for each batch with N pairs):

$$L_{\text{CLIP}} = -\frac{1}{2N}\sum_{i=1}^{N}\log\frac{\exp(I_i \cdot T_i)}{\sum_{j=1}^{N}\exp(I_i \cdot T_j)} - \frac{1}{2N}\sum_{j=1}^{N}\log\frac{\exp(I_j \cdot T_j)}{\sum_{i=1}^{N}\exp(I_i \cdot T_j)}$$

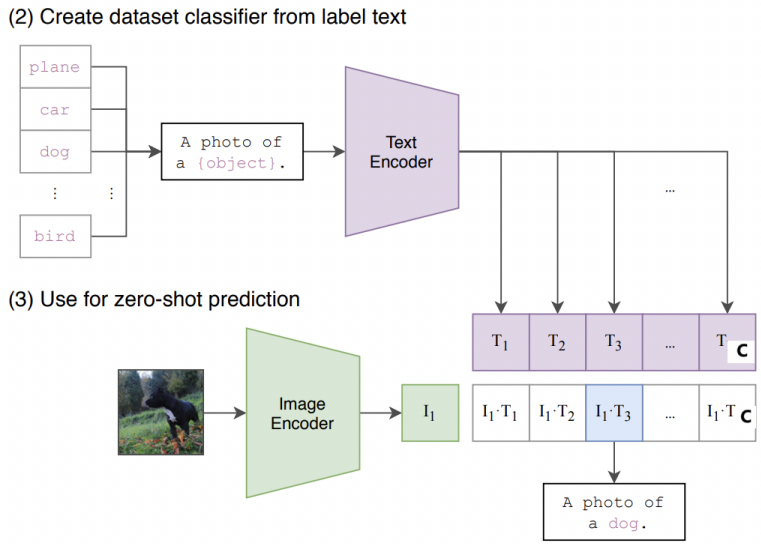
$\log P(T_i \mid I_i)$ $\qquad$ $\log P(I_j \mid T_j)$

  - For each image I_i, the paired text T_i should have the maximum probabilities
  - For each text T_j, the paired image T_j should have the maximum probabilities

CLIP for Zero-shot Image Classification



CLIP for Zero-shot Image Classification



- Create the text prompt T_c for each class c: "A photo of a {c}."
- Probability of a class c is given by

$$\frac{\exp(I \cdot T_c)}{\sum_{c'=1}^{C} \exp(I \cdot T'_c)}$$

- Essentially, T_c is the vector of weights for the class c.

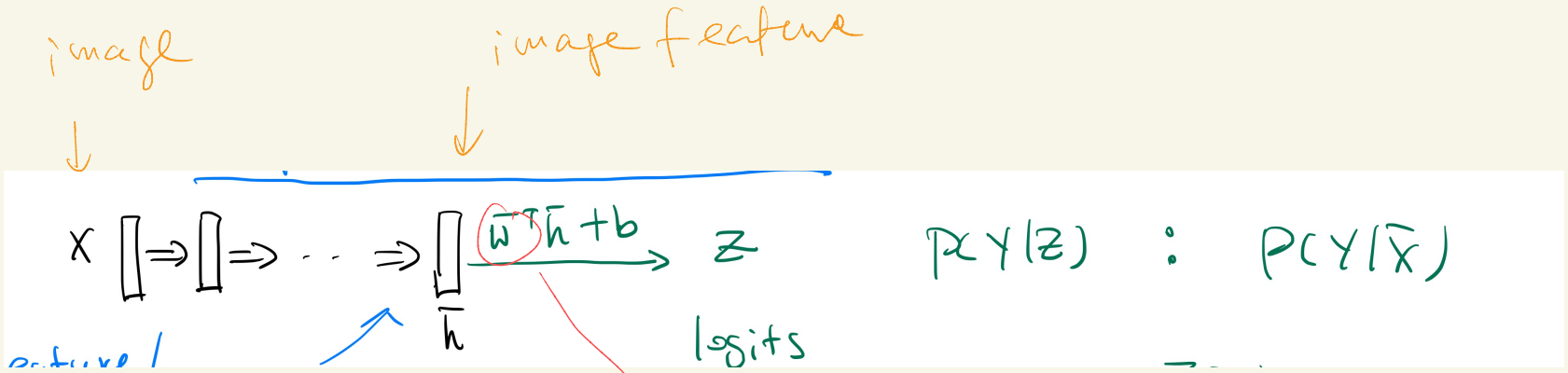
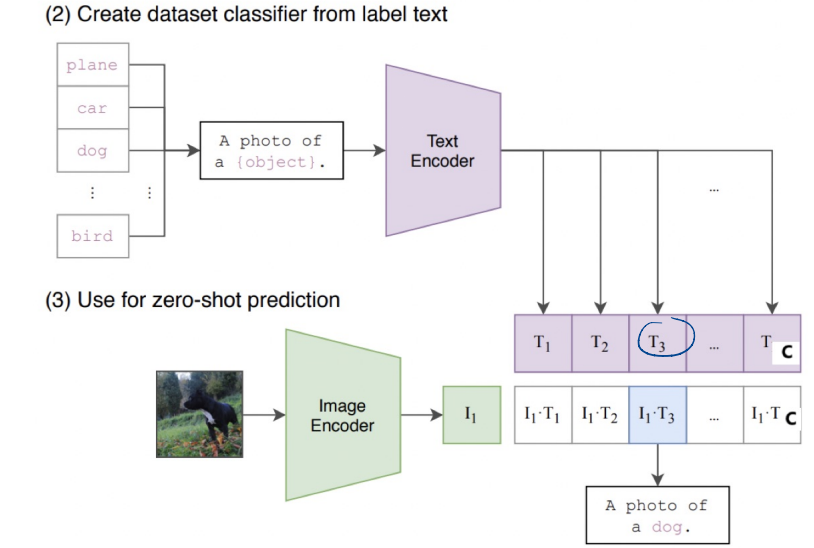


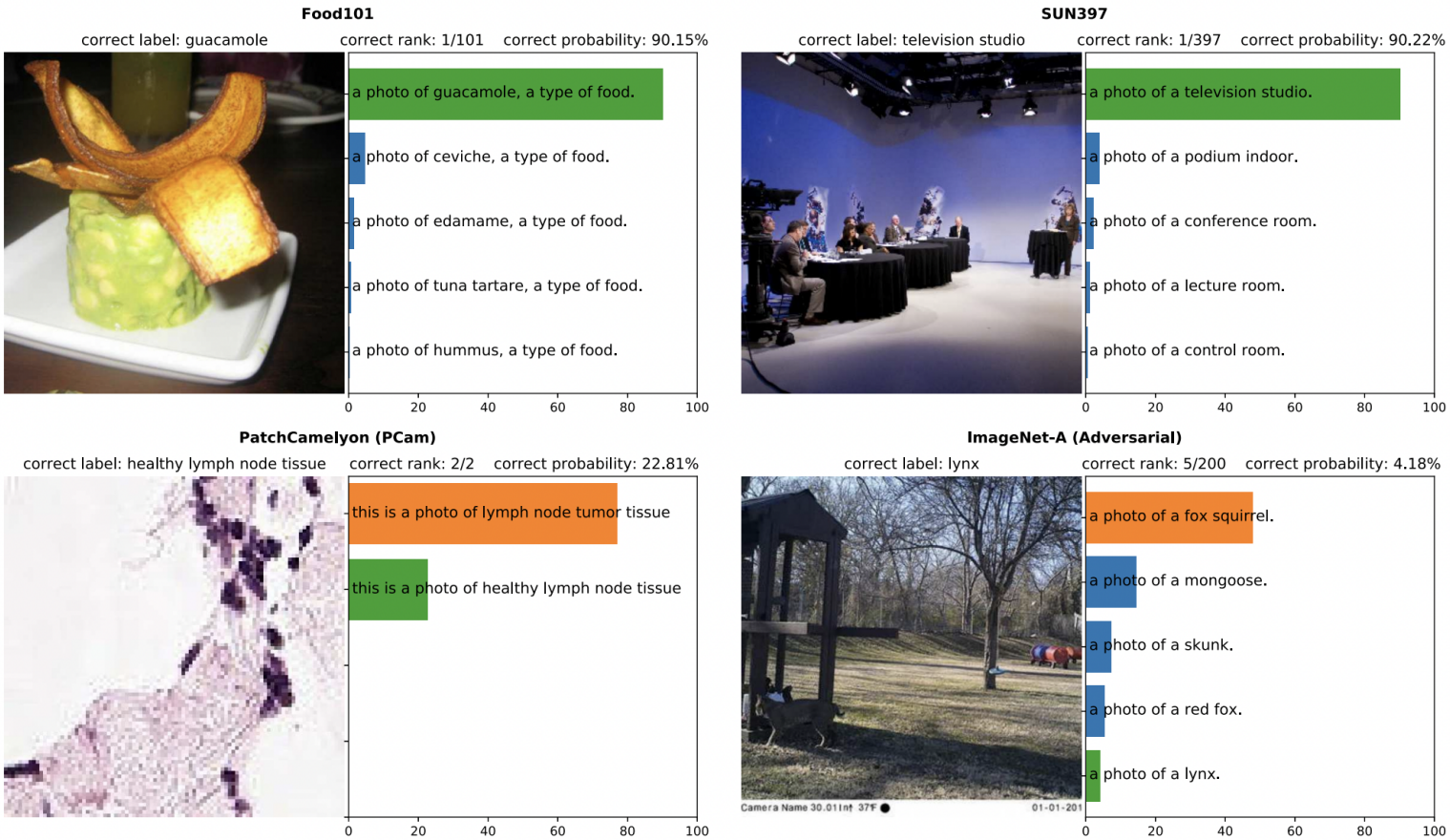
image feature

Same role

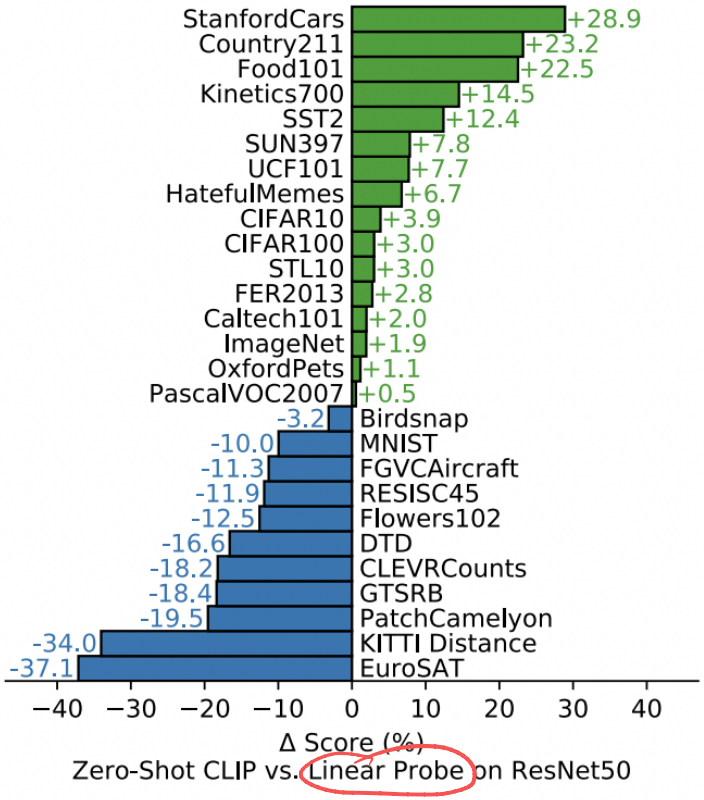
$$\frac{\exp(I \cdot T_c)}{\sum_{c'=1}^{C} \exp(I \cdot T'_c)}$$

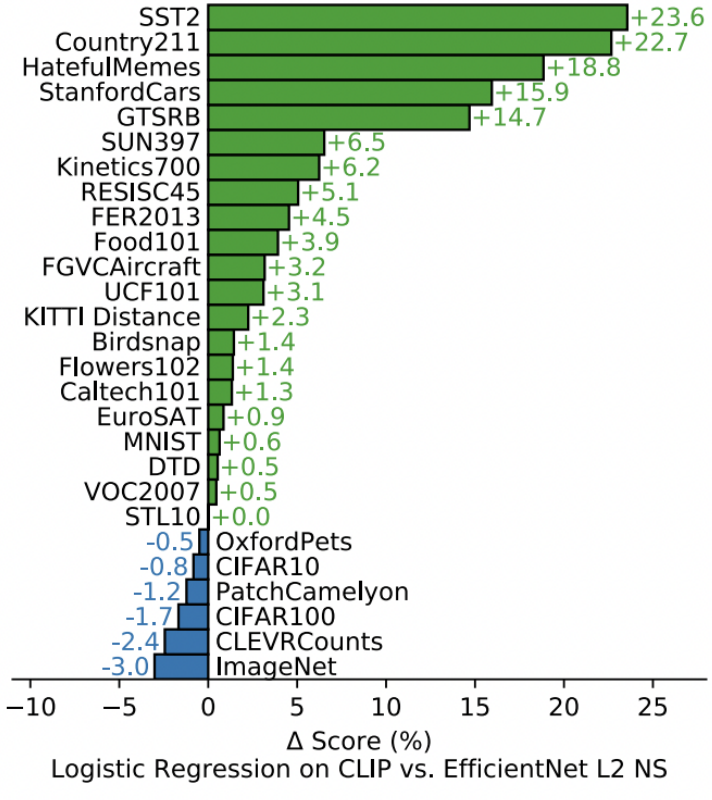


CLIP - Zero-shot Classification



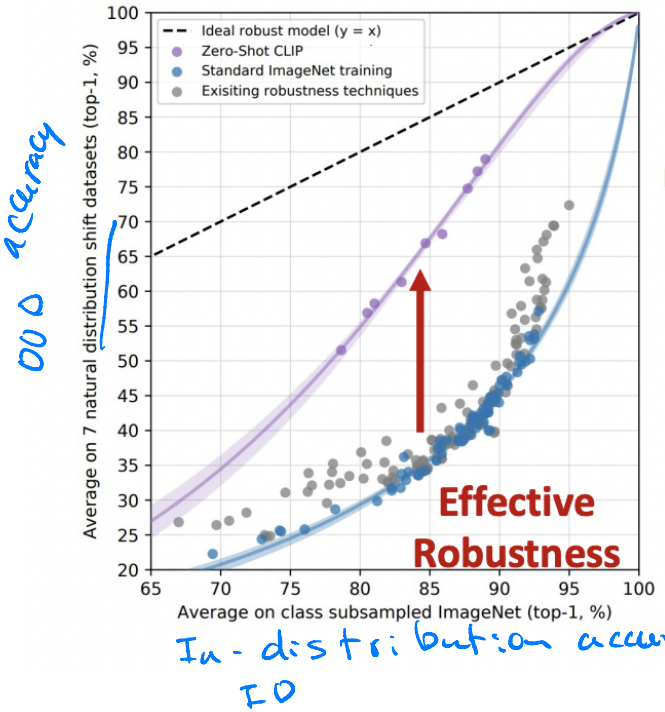
CLIP - Zero-shot transfer

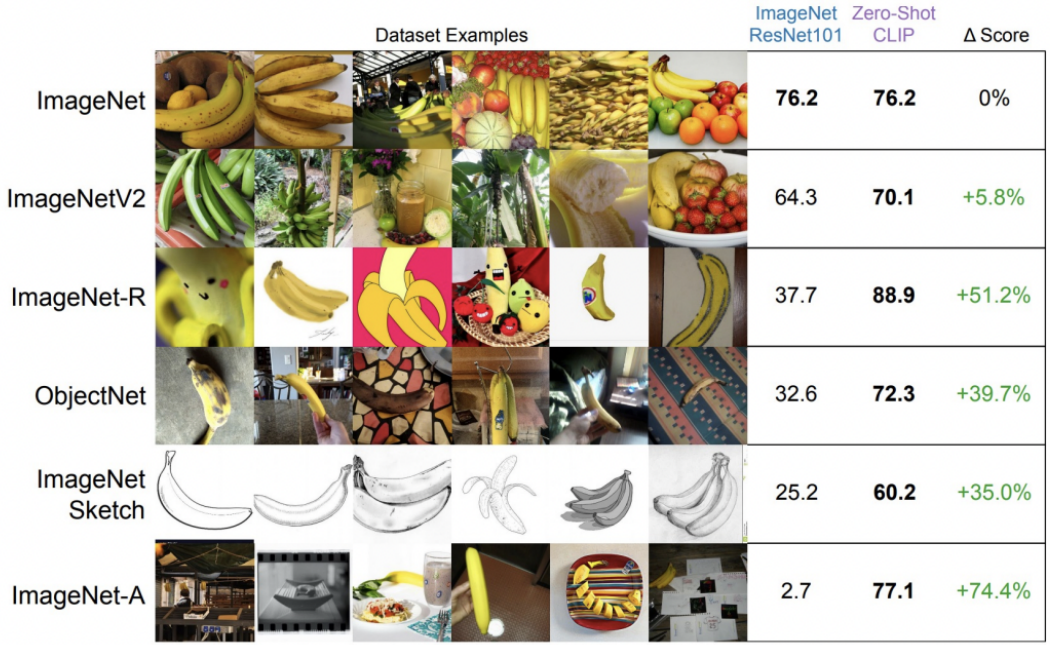




- A zero-shot CLIP classifier shows a competitive performance with a fully supervised linear classifier fitted on ResNet-50 features (pre-trained on ImageNet). If classes Apply it to 2-class problem
- Linear-probing with CLIP image features outperform the best ImageNet model.

CLIP classifier is more robust to natural distributional shift

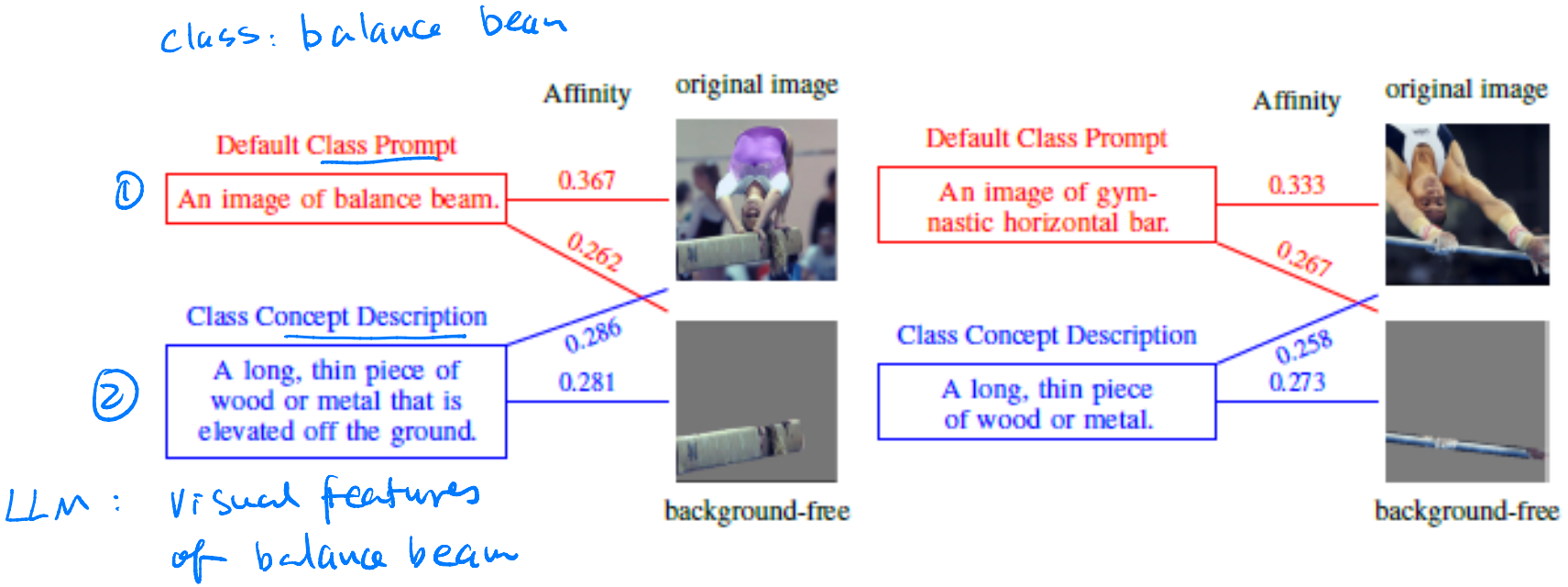




- Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models.
- An ideal robust model (dashed line in the left graph) performs equally well on the ImageNet distribution and on other natural image distributions. Zero-shot CLIP models shrink this "robustness gap" by up to 75%.

Alternative Way to Use CLIP for Image Classification [3]

- Prompt LLM for produce a textual concept description the core visual features of a class
- Use the concept description as text prompt for the class in classification
- Leads to more robust classification



[3] Li, Xie et al 2024, Dual Risk Minimization: Towards Next-Level Robustness in Fine-tuning Zero-Shot Models, NeuRIPS

CLIP - Scaling with larger training datasets

	ALIGN (Jia et al., 2021)	**CLIP** (Radford et al., 2021)	**BASIC** (ours)
ImageNet	76.4	76.2	**85.7** (+9.3)
ImageNet-A	75.8	77.2	**85.6** (+8.4)
ImageNet-R	92.2	88.9	**95.7** (+3.5)
ImageNet-V2	70.1	70.1	**80.6** (+10.5)
ImageNet-Sketch	64.8	60.2	**76.1** (+11.3)
ObjectNet	72.2	72.3	**82.3** (+10.1)
Average	74.5	74.2	**84.3** (+10.1)

- CLIP uses carefully filtered 400M image-text pairs from web.
- ALIGN [Jia et al., 2020] collected noisy 1.8B image-text pairs to scale CLIP.
- BASIC [Pham et al., 2021] used 6.6B image-text pairs with bigger model size.

Jia, Chao, et al. Scaling up visual and vision-language representation learning with noisy text supervision. ICML, 2021.

Pham, Hieu, et al. "Combined scaling for zero-shot transfer learning." Neurocomputing 555 (2023): 126658.