Multimodality Helps Unimodality: Cross-Modal Few-Shot Learning with Multimodal Models

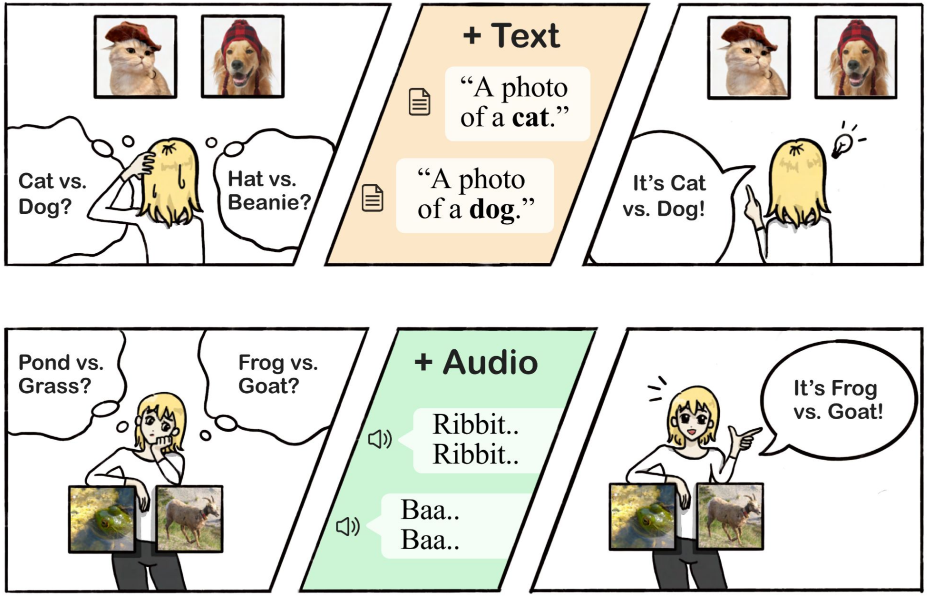
Zhiqiu Lin*, Samuel Yu*, Zhiyi Kuang, Deepak Pathak, Deva Ramanan

THU-AM-271

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

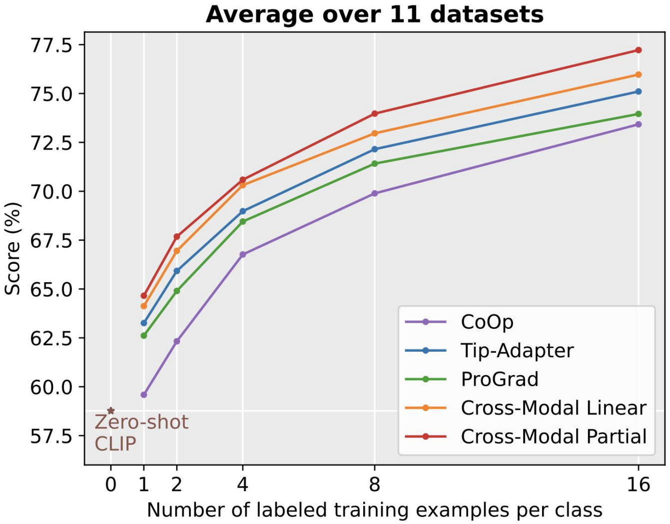
Preview

Unimodal few-shot learning is **underspecified**



Multimodal training data can help resolve this ambiguity!

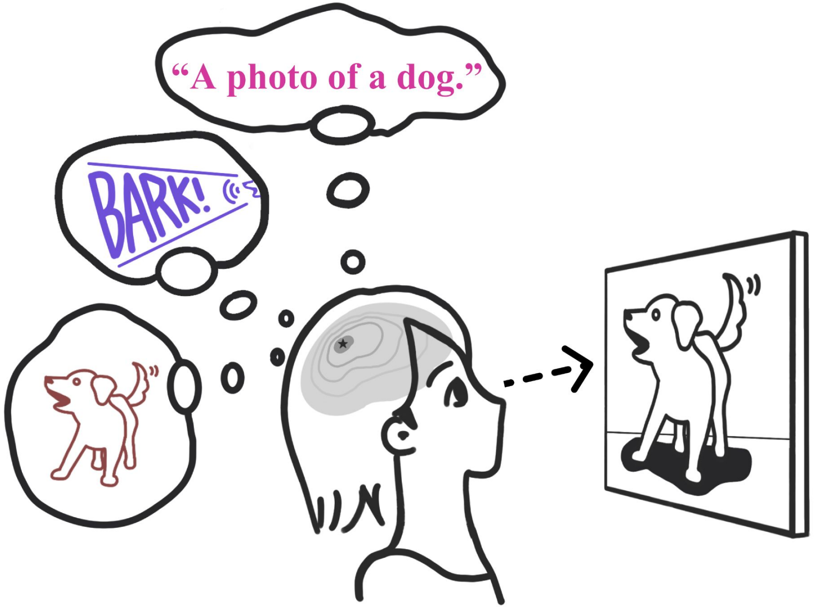
Preview



Method	Number of shots					Train speed
	1	2	4	8	16	
Zero-Shot CLIP (58.8)	-	-	-	-	-	-
Linear Probing	36.7	47.6	57.2	65.0	71.1	<1min
WiSE-FT [100]	59.1	61.8	65.3	68.4	71.6	<1min
CoOp [113]	59.6	62.3	66.8	69.9	73.4	14hr
ProGrad [114]	62.6	64.9	68.5	71.4	74.0	17hr
Tip-Adapter† [111]	63.3	65.9	69.0	72.2	75.1	5min
Cross-Modal Linear Probing	64.1	67.0	70.3	73.0	76.0	<1min
Cross-Modal Partial Finetuning	**64.7**	**67.2**	**70.5**	**73.6**	**77.1**	<3min

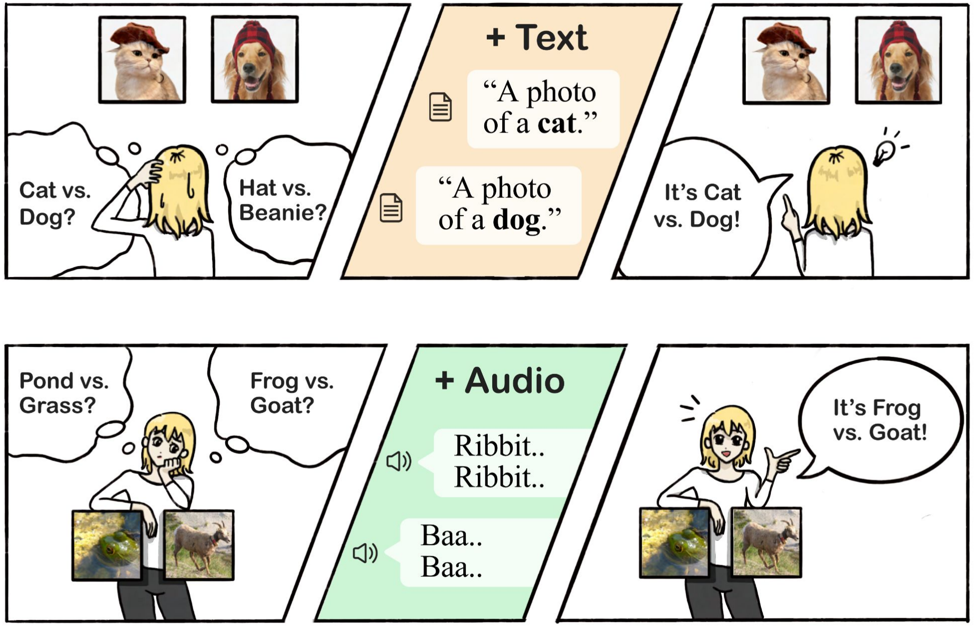
We achieve SOTA few-shot CLIP image classification performance

Human perception is inherently **cross-modal**.



When we perceive from one modality (such as **vision**), the same neurons will be triggered in our cerebral cortex as if we are perceiving the object from other modalities (such as **language** and **audio**).

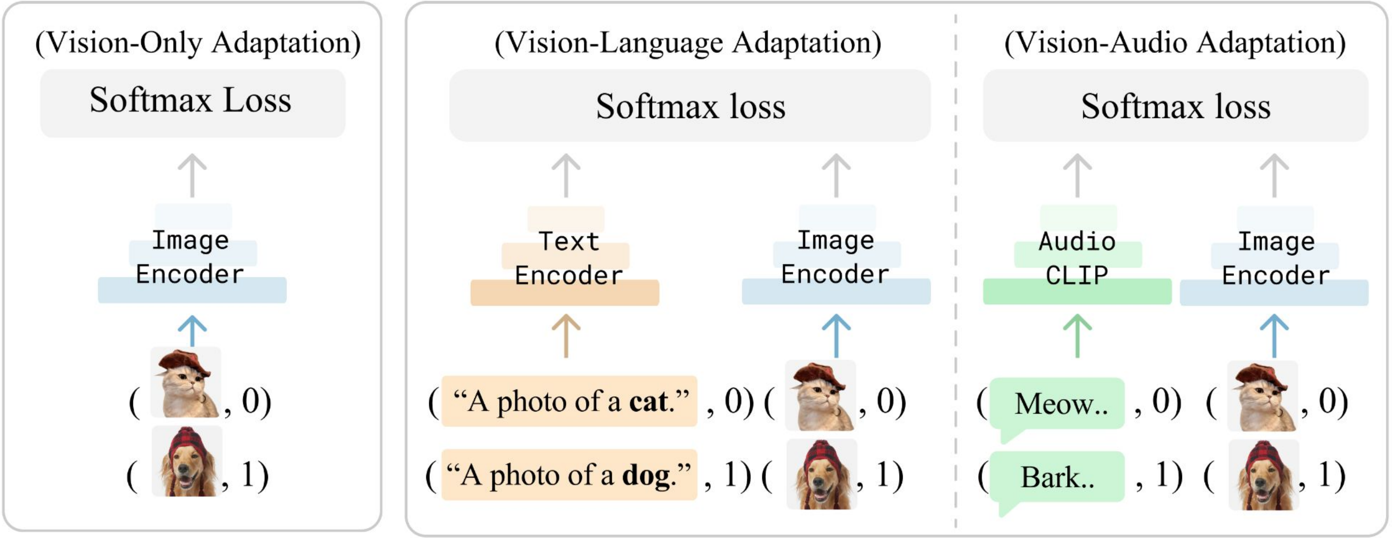
Few-shot learning can be more **specified** with **cross-modalities**.



Unimodal few-shot learning setups are often **under-specified**: even for simple binary image classification tasks, it is unclear whether the class target is the *animal*, the *hat*, or the *background scene* if we are given only one-shot images.

On the other hand, adding language or audio can help clarify the image classification setup.

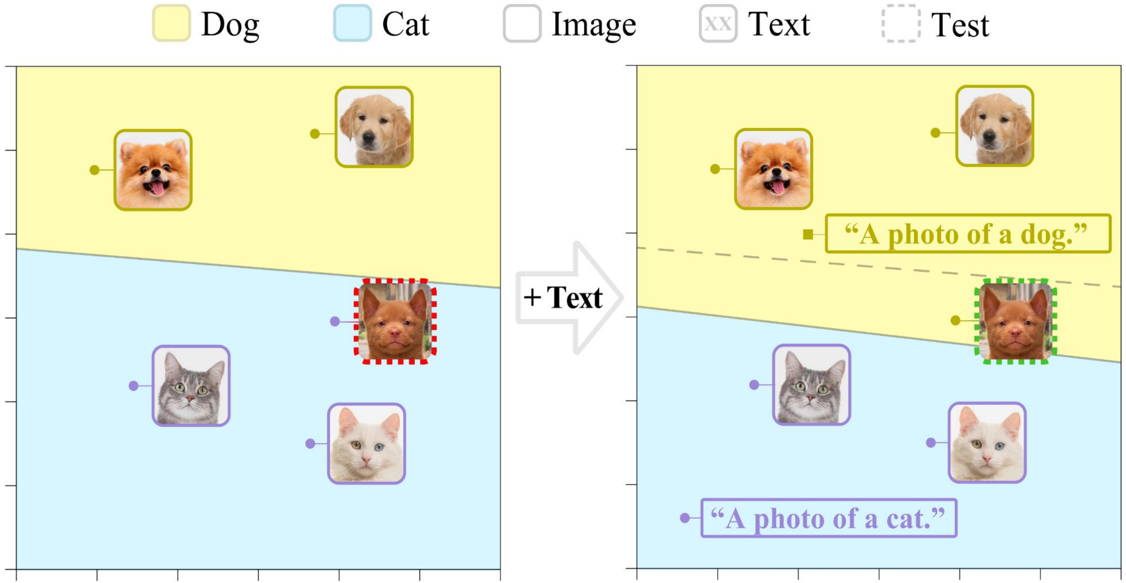
Like human perception, multimodal foundation models such as *CLIP* and *AudioCLIP* encode different modalities to the **same representation space**.



We introduce **Cross-Modal Adaptation** to address few-shot learning with multimodal models by:

- Repurposing information from other modalities like class names and audio clips as training samples.
- Minimizing the softmax (cross-entropy) loss over the augmented (n+1)-shot training set.

Build a better *visual* dog classifier by *reading* about dogs!



Text samples help regularize the decision boundary: a “*dog*” label with 2-shot *images* can be more performant than 3-shot *images*.

We achieve SOTA performance via an embarrassingly simple cross-modal linear probe

Method	Number of shots					Train speed
	1	2	4	8	16	
Zero-Shot CLIP (58.8)	-	-	-	-	-	-
Linear Probing	36.7	47.6	57.2	65.0	71.1	<1min
WiSE-FT [100]	59.1	61.8	65.3	68.4	71.6	<1min
CoOp [113]	59.6	62.3	66.8	69.9	73.4	14hr
ProGrad [114]	62.6	64.9	68.5	71.4	74.0	17hr
Tip-Adapter† [111]	63.3	65.9	69.0	72.2	75.1	5min
Cross-Modal Linear Probing	64.1	67.0	70.3	73.0	76.0	<1min
Cross-Modal Partial Finetuning	**64.7**	**67.2**	**70.5**	**73.6**	**77.1**	<3min

And maintain **high** training efficiency

Cross-modal adaptation consistently improves prior art

Method	Number of shots				
	1	2	4	8	16
Linear Probing	36.7	47.6	57.2	65.0	71.1
Cross-Modal Linear Probing	64.1	67.0	70.3	73.0	76.0
Δ	27.4	19.4	13.1	8.0	4.9
WiSE-FT [100]	59.1	61.8	65.3	68.4	71.6
Cross-Modal WiSE-FT	63.8	66.4	69.0	71.7	74.1
Δ	4.7	4.6	3.7	3.3	2.5
CoOp [113]	59.6	62.3	66.8	69.9	73.4
Cross-Modal Prompting	62.0	64.9	68.6	71.4	74.0
Δ	2.4	2.6	1.8	1.5	0.6
Tip-Adapter† [111]	63.3	65.9	69.0	72.2	75.1
Cross-Modal Adapter	64.4	67.6	70.8	73.4	75.9
Δ	1.1	1.7	1.8	1.2	0.8

We curate the first *audiovisual* few-shot learning benchmark

Included Dataset	ESC-50 [77] Class	ImageNet [15] Class
ImageNet-ESC-19	rooster	rooster
	hen	hen
	chirping-birds	chickadee
	frog	tree frog
	dog	otterhound
	cat	egyptian cat
	insects	fly
	crickets	cricket
	pig	pig
	sheep	big-horn sheep
	airplane	airliner
	train	high-speed train
	chainsaw	chainsaw
	keyboard-typing	computer keyboard
	clock-alarm	digital clock
	mouse-click	computer mouse
	vacuum-cleaner	vacuum cleaner
	clock-tick	wall clock
	washing-machine	washing machine
ImageNet-ESC-27	can-opening	can opener
	church-bells	church bells
	crackling-fire	fire screen
	toilet-flush	toilet seat
	water-drops	sink
	drinking-sipping	water bottle
	pouring-water	water jug
	sea-waves	sandbar

Build a better *visual* dog classifier by *listening* to dog barks!
And build a better *audio* dog classifier by *looking* at dog photos!

Dataset	Method	Image Classification		
		1-shot	2-shot	4-shot
ImageNet-ESC-19	Image-Only Linear	68.0	75.7	83.1
	Image-Audio Linear	**69.3**	**76.7**	**83.2**
ImageNet-ESC-27	Image-Only Linear	60.1	71.8	**79.0**
	Image-Audio Linear	**60.9**	**73.3**	78.9

Dataset	Method	Audio Classification		
		1-shot	2-shot	4-shot
ImageNet-ESC-19	Audio-Only Linear	31.2	41.1	48.5
	Audio-Image Linear	**35.7**	**45.9**	**51.6**
ImageNet-ESC-27	Audio-Only Linear	28.2	39.0	47.1
	Audio-Image Linear	**35.0**	**43.5**	**48.5**

Why does it work? **Representer Theorem**

$$w_y = \sum_i \alpha_{iy} \phi_{m_i}(x_i) = \sum_{m \in M} w_y^m, \quad \text{where}$$

$$w_y^m = \sum_{\{i : m_i = m\}} \alpha_{iy} \phi_m(x_i).$$

We **separately optimize** the weight for each text feature instead of setting a **fixed** weight for all text features (WiSE-FT)

Implementation is simple

```
# w:  linear layer initialized with text features
# T: temperature scaling (default is 100)
for _ in iteration:
    # Randomly sample images and texts
    im, im_labels = image_loader.next()
    tx, tx_labels = text_loader.next()

    # Extract image and text features
    im_f = image_encoder(im)
    tx_f = text_encoder(tx)

    # Put in same batch then L2 normalize
    features = cat((im_f, tx_f), dim=0)
    features = normalize(features, dim=1)
    labels = cat((im_labels, tx_labels), dim=0)

    # Compute softmax (cross entropy) loss
    logits = w(features)
    loss = cross_entropy_loss(logits / T, labels)
    loss.backward()

    # Update linear layer
    update(w.params)
    # [optional] Update (partial or full) encoders
    update(image_encoder.params)
    update(text_encoder.params)
```

Thank You!

THE ROBOTICS INSTITUTE

Carnegie Mellon University

Website

Paper