

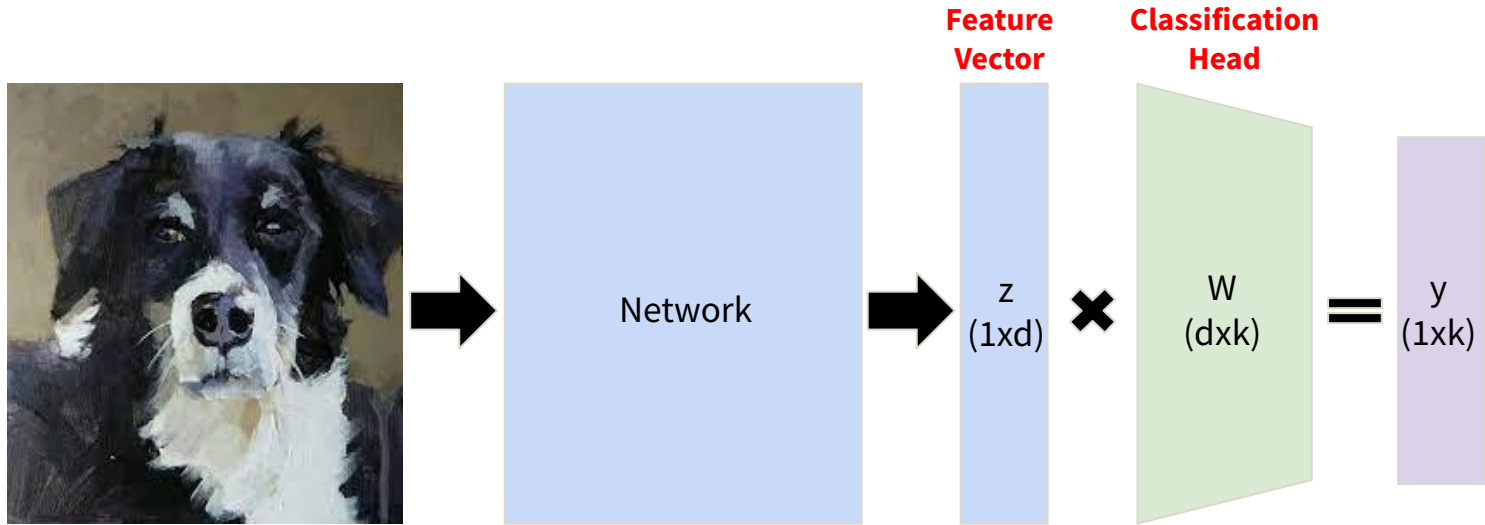
# Zero-Shot Generalization

B9145: RELIABLE STATISTICAL LEARNING

# Roadmap

- Recap last discussion
- Zero-shot generalization
  - Motivation
  - CLIP
  - GPT

# Review: NN Terminology



$d$  = hidden dimension  
 $k$  = number of classes (train)

Feature vector a.k.a. representation, embedding  
Classification head a.k.a. linear classifier

# Recap: Generalization

Our model learns from training data, but we want it to be **robust** to shifts in the input distribution and **flexible** enough to perform many tasks



(A) Cow: **0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98



(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97



(C) No Person: 0.97, Mammal: **0.96**, Water: 0.94, Beach: 0.94, Two: 0.94

**Fig. 1. Recognition algorithms generalize poorly to new environments.** Cows in ‘common’ contexts (e.g. Alpine pastures) are detected and classified correctly (A), while cows in uncommon contexts (beach, waves and boat) are not detected (B) or classified poorly (C). Top five labels and confidence produced by ClarifAI.com shown.



## Recap: Seeking generalization

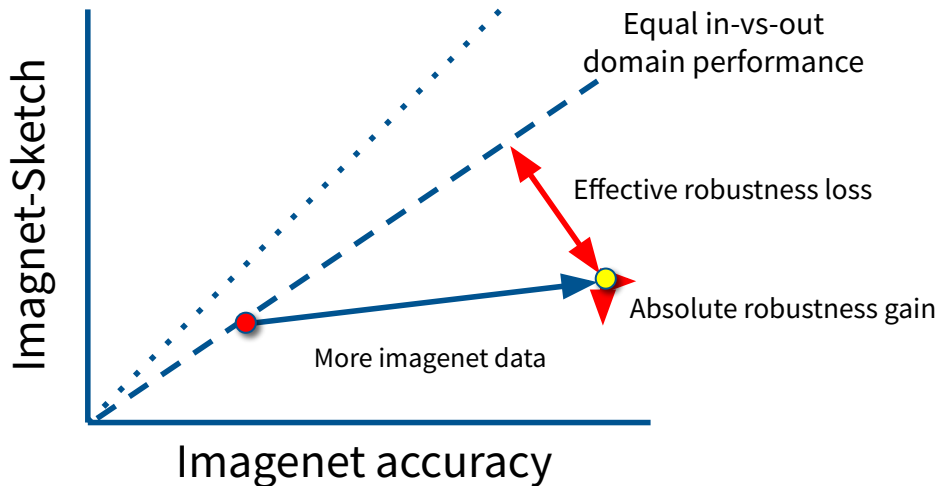
- Most robustness interventions (e.g. new model architectures, collecting more i.i.d. data, adversarial training) do not increase effective and relative robustness, or come with unacceptable losses in absolute robustness

# Analyzing absolute vs effective robustness

**Absolute:** OOD performance

**Effective:** OOD performance beyond what can be predicted by ID performance

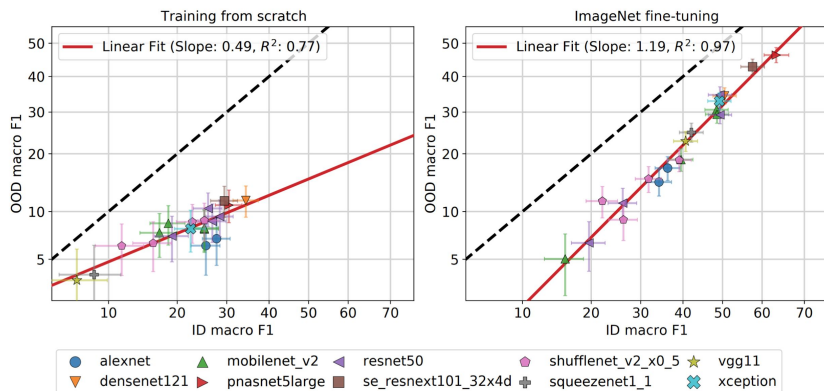
**Relative:** OOD performance gained by applying robustness intervention



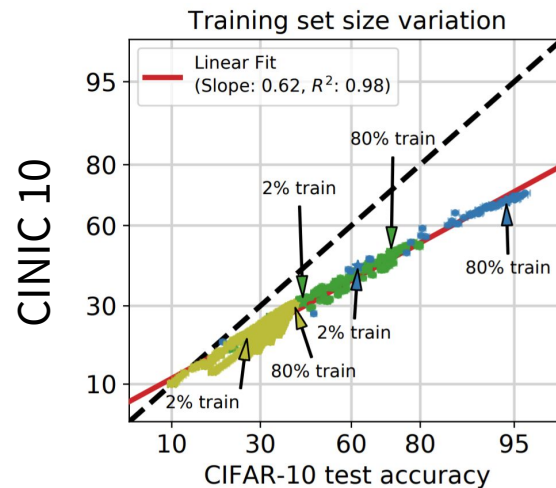
- Adding data may increase *absolute robustness* but decrease *effective robustness*
- Robustness intervention may increase *effective robustness* but decrease *absolute robustness*

# Accuracy on the Line

**Changes in architecture do not increase effective robustness...**



**...nor does adding more in-domain data**



# Adversarial training

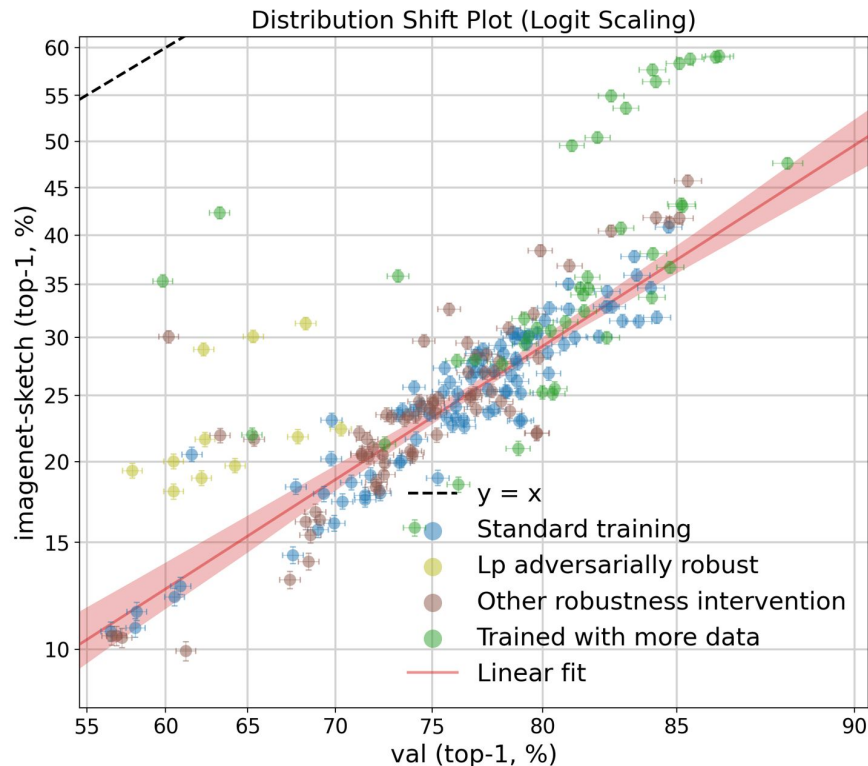
Adversarial perturbations can cause a model to fail

- But we can train to be resilient to this

This leads to substantial effective robustness gains

- Drop in standard accuracy shifts points to the left
- Increase in robust accuracy shift points off the line

Adversarial examples improve effective (but not absolute) robustness.



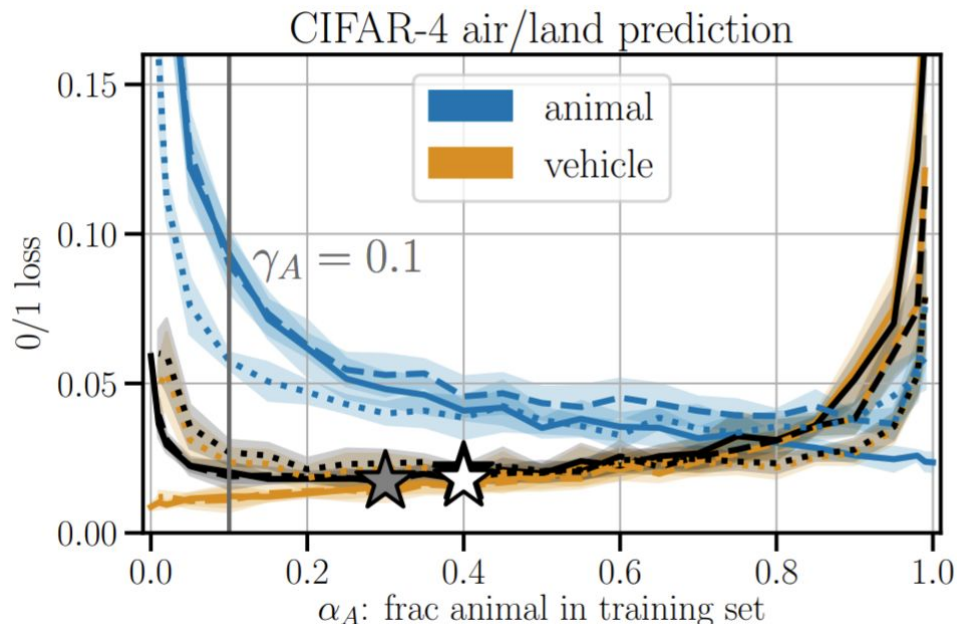


## Recap: Seeking generalization

- Most robustness interventions (e.g. new model architectures, collecting more i.i.d. data, adversarial training) do not increase effective and relative robustness, or come with unacceptable losses in absolute robustness
- Smart data collection strategies and taking advantage of unlabeled data for pre-training can help

# Smart data collection

Picking the right ‘mix’ of data sources can lead to substantial improvements.



[Rolf+ 2021]

**Takeaways:** If we want similar performance across groups, not having any animals/vehicles = catastrophic. Want > 50% animals.

# Self-Supervised Learning

Take a (massive) unlabeled dataset and create a supervised learning problem

→ Objective does not matter, goal is **feature learning**

**SimCLR:** Contrastive learning - predict whether views are derived from same image

NT-Xent

$$\mathbf{u}^T \mathbf{v}^+ / \tau - \log \sum_{\mathbf{v} \in \{\mathbf{v}^+, \mathbf{v}^-\}} \exp(\mathbf{u}^T \mathbf{v} / \tau)$$

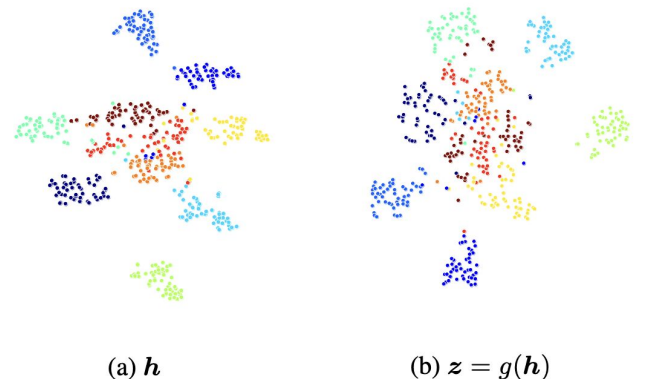
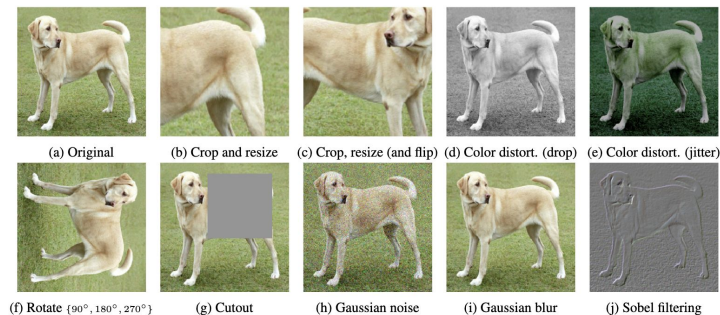
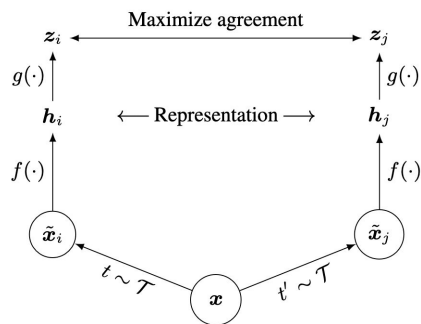


Figure B.4. t-SNE visualizations of hidden vectors of images from a randomly selected 10 classes in the validation set.

## Recap: Seeking generalization

- Most robustness interventions (e.g. new model architectures, collecting more i.i.d. data, adversarial training) do not increase effective and relative robustness, or come with unacceptable losses in absolute robustness
- Smart data collection strategies and taking advantage of unlabeled data for pre-training can help
- Models fail to generalize because information about the training domain leaks into features

$$\boxed{E_{p(t), p(\theta)}[l(\theta, t)]} \leq \boxed{E_{p(t), p(\theta|t)}[l(\theta, t)]} + \boxed{\mathcal{O}(\sqrt{I(\theta, t)})}$$

cross-domain loss      in-domain loss      information used

# Spurious Correlations

Misleading heuristics that work for most training examples but do not hold for the general case

Ex.


- waterbirds over water
- blond hair, female
- contradiction, negation

Sagawa+ 2020


**Common training examples**

**Waterbirds**


y: waterbird  
a: water background



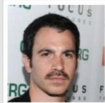
y: landbird  
a: land background



y: blond hair  
a: female



y: dark hair  
a: male




**Test examples**

y: waterbird  
a: land background



y: blond hair  
a: male



**MultiNLI**

y: contradiction  
a: has negation

(P) The economy could be still better.  
(H) The economy has never been better.

y: entailment  
a: no negation

(P) Read for Slate's take on Jackson's findings.  
(H) Slate had an opinion on Jackson's findings.

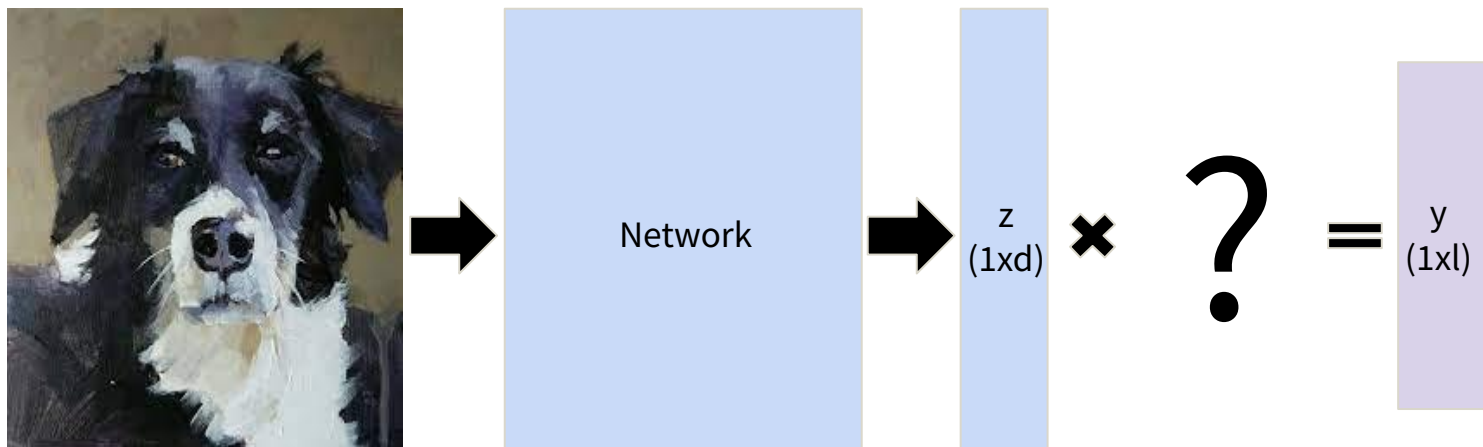
y: entailment  
a: has negation

(P) There was silence for a moment.  
(H) There was a short period of time where no one spoke.

		Average Accuracy		Worst-Group Accuracy	
		ERM		ERM	
Standard Regularization	Waterbirds	Train	100.0		100.0
		Test	97.3		60.0
	CelebA	Train	100.0		99.9
		Test	94.8		41.1
	MultiNLI	Train	99.9		99.9
		Test	82.5		65.7

## Few-shot learning before ~2020

Where do we get classification matrix  $W$  if we have a different set of classes from training to test, or we trained on a synthetic objective?



$d$  = hidden dimension  
 $l$  = number of classes (test)

## Zero-shot generalization

Motivated by:

$$\underbrace{E_{p(t), p(\theta)}[l(\theta, t)]}_{\text{cross-domain loss}} \leq \underbrace{E_{p(t), p(\theta|t)}[l(\theta, t)]}_{\text{in-domain loss}} + \underbrace{\mathcal{O}(\sqrt{I(\theta, t)})}_{\text{information used}}$$

We seek a model that is never exposed to the spurious correlations of a particular source domains

→ Then domain accuracy should mostly just be about how hard the domain is.

# CLIP: Jointly embedding images and text

## Feature embedding model

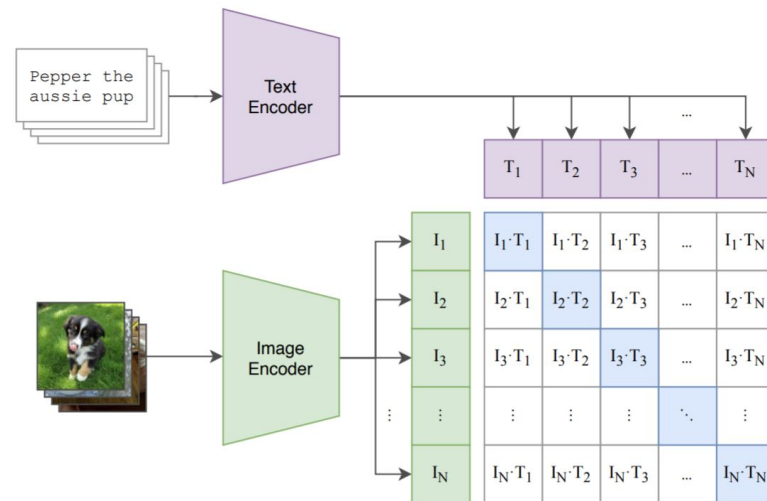
### Encoders

- Image: ResNet, ViT
- Text: Transformer

### Train ‘contrastively’

- large batches (32K)
- positive example: paired caption
- negative example: all other captions

(1) Contrastive pre-training





# Training CLIP

```
# 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
```

(1) Contrastive pre-training

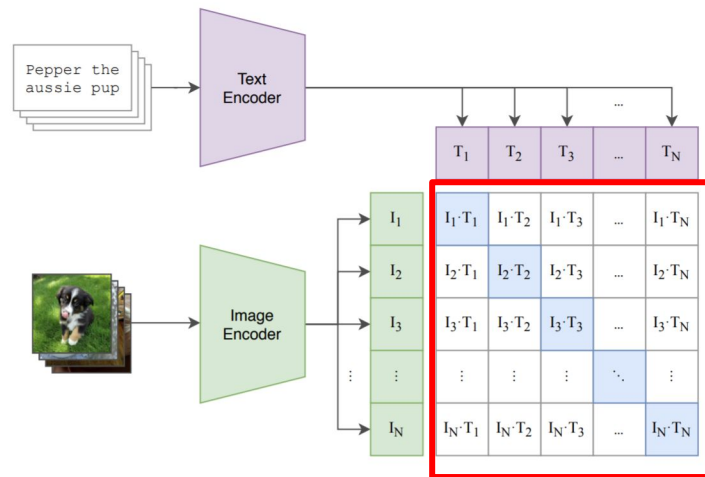


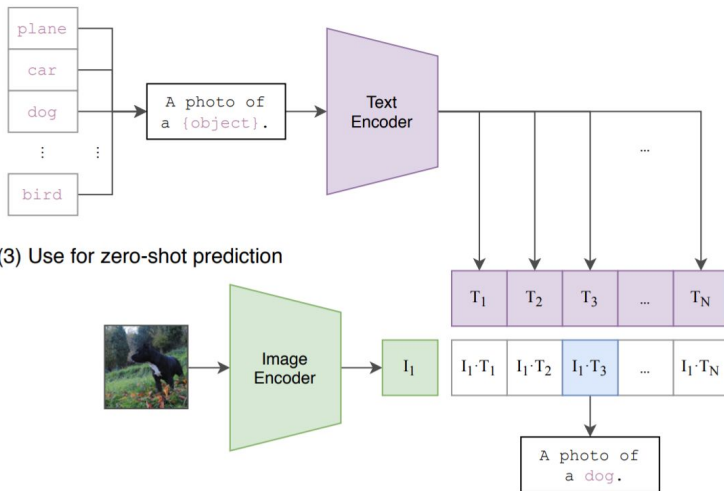
Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

# Generalizing with CLIP

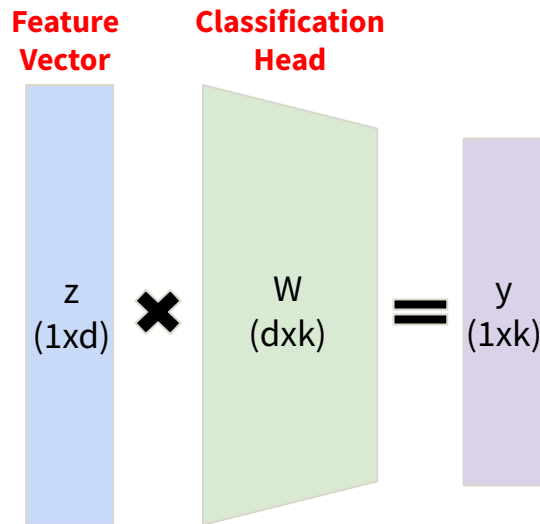
Assume we have an embedding space where the representation of an image is close to valid captions of the image

→ Then embedding of our  $k$  text labels to  $d$ -dimensional vectors gives us  $W$ !

(2) Create dataset classifier from label text



(3) Use for zero-shot prediction



# Creating the CLIP classifier

For each class label:

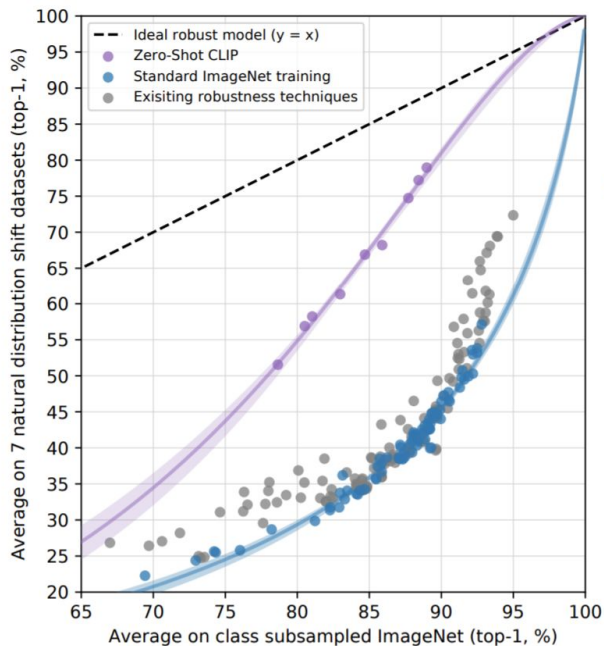
1. Add to templates
2. Get d-dimensional representations of templates+label
3. Average over representations of filled templates

```
3 templates7 = '''itap of a {}.a bad photo of the {}.a origami {}.a photo of the large {}.
4 a {} in a video game.art of the {}.a photo of the small {}.''.split('\n')
5
6 def zeroshot_classifier(classnames, templates):
7     with torch.no_grad():
8         zeroshot_weights = []
9         for classname in tqdm.tqdm(classnames):
10            texts = [template.format(classname) for template in templates] #format with class
11            texts = clip.tokenize(texts).cuda() #tokenize
12            class_embeddings = model.encode_text(texts) #embed with text encoder
13            class_embeddings /= class_embeddings.norm(dim=-1, keepdim=True)
14            class_embedding = class_embeddings.mean(dim=0)
15            class_embedding /= class_embedding.norm()
16            zeroshot_weights.append(class_embedding)
17            zeroshot_weights = torch.stack(zeroshot_weights, dim=1).cuda()
18        return zeroshot_weights
19
20
21 zeroshot_weights = zeroshot_classifier(cifar100.classes, templates7)
22 zeroshot_weights_ = zeroshot_classifier(cifar100.classes, templates80)
23 zeroshot_weights.shape, zeroshot_weights_.shape
```

```
100%|██████████| 100/100 [00:09<00:00, 10.50it/s]
100%|██████████| 100/100 [01:36<00:00, 1.04it/s]
(torch.Size([512, 100]), torch.Size([512, 100]))
```

← d x k matrix

# Observations from a zero-shot model (CLIP)



	Dataset Examples	ImageNet ResNet101	Zero-Shot CLIP	$\Delta$ 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%

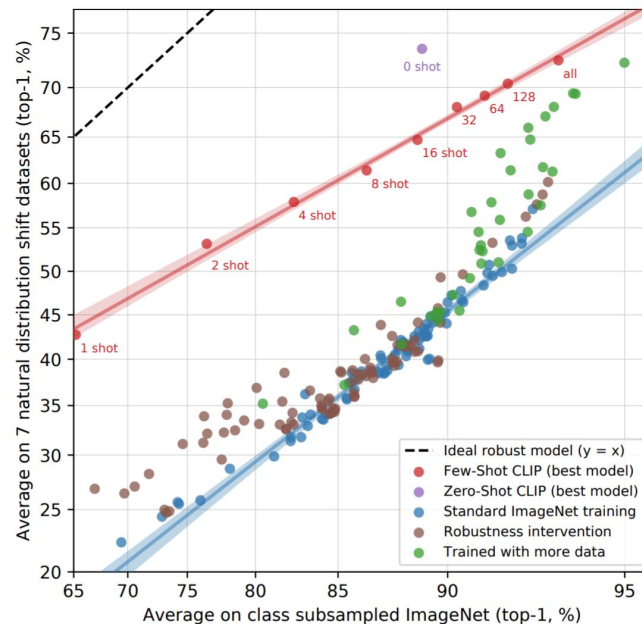
## Few shot robustness

Few-shot performance also shows similar trends.

As we add data (1-shot to 128-shot to all)

- absolute robustness increases.
- relative robustness decreases.

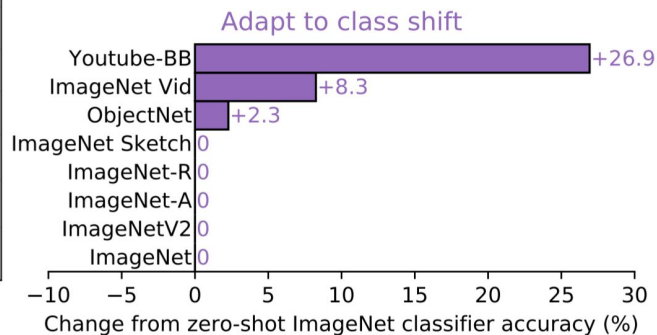
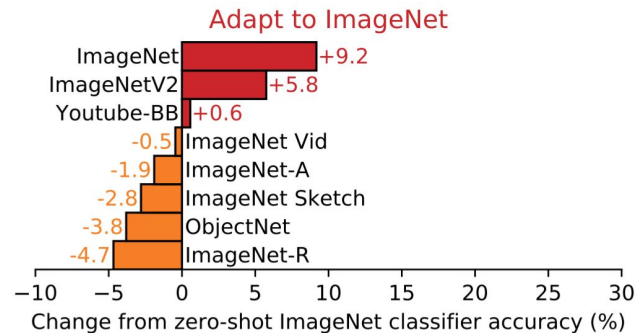
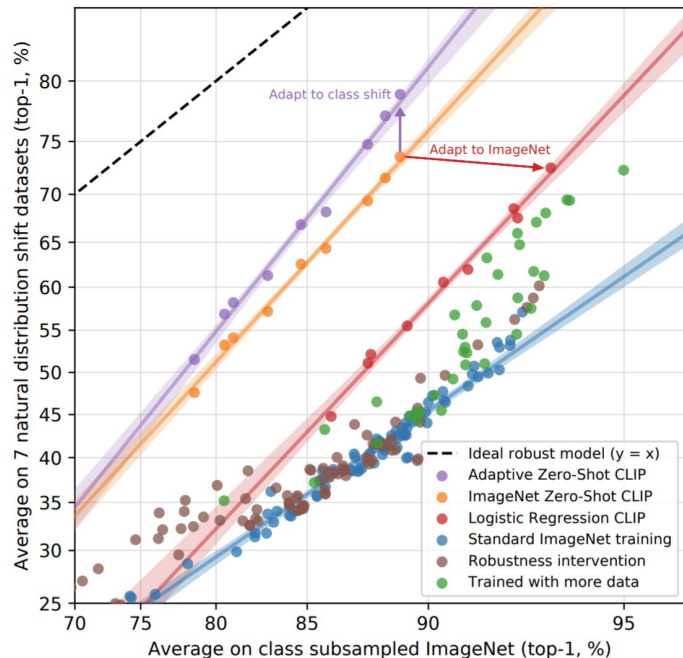
‘Zero shot and few shot models are inherently robust’



**Figure 15. Few-shot CLIP also increases effective robustness compared to existing ImageNet models but is less robust than zero-shot CLIP.** Minimizing the amount of ImageNet training data used for adaption increases effective robustness at the cost of decreasing relative robustness. 16-shot logistic regression CLIP matches zero-shot CLIP on ImageNet, as previously reported in Figure 7, but is less robust.

# More robustness observations

Fine-tuning on imagenet data kills these robustness gains (red line)

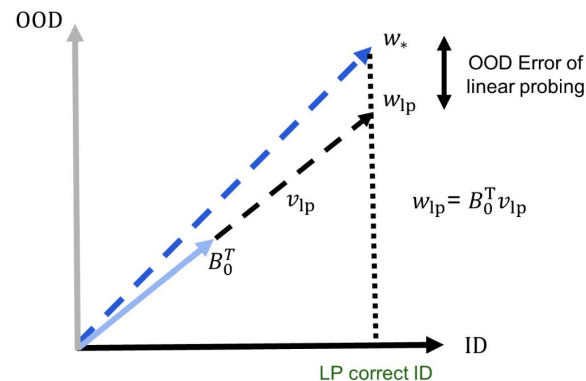


Problems are not a lack of data!

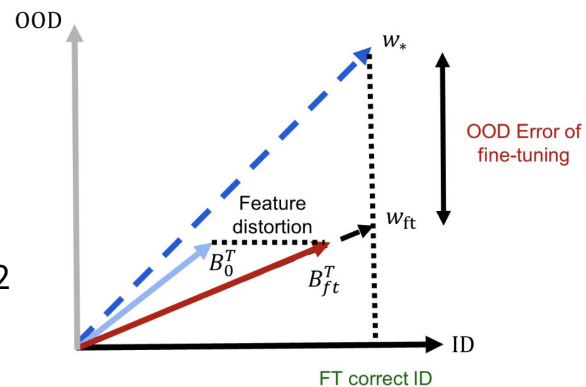
# Feature distortion under fine-tuning

1. **Features get distorted:** Representations of ID training data are updated while those of OOD data change less
2. **Distorted features can lead to higher OOD error:** Classification head is optimized for use with updated feature extractor, performs poorly on less changed features of OOD points

Kumar+ 2022



(a) Toy example (Linear probing)



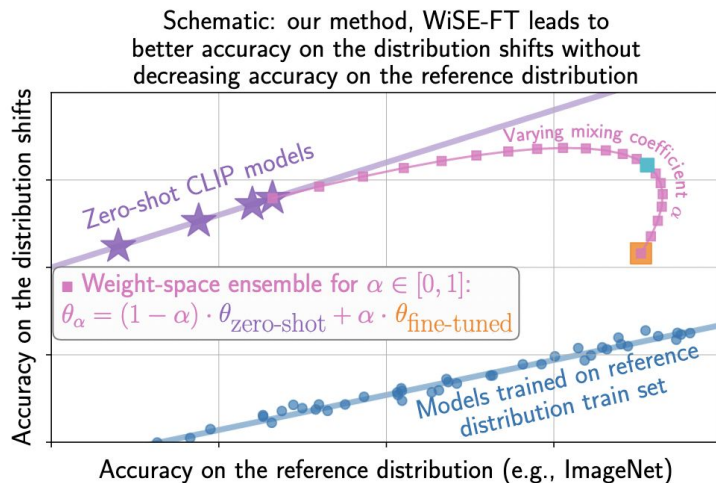
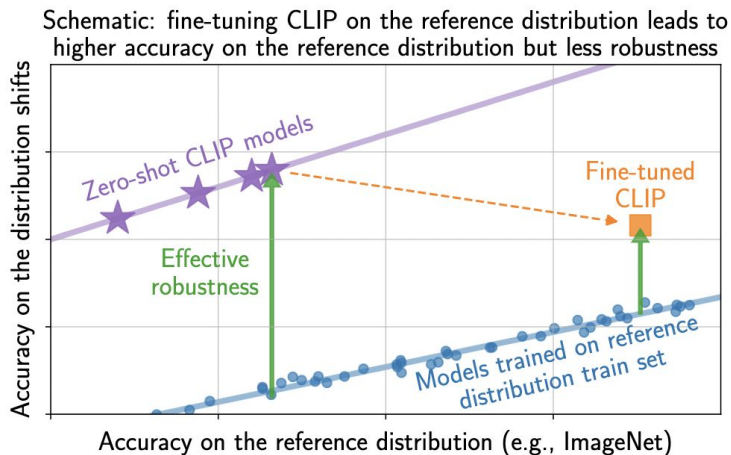
(b) Toy example (fine-tuning)

# Robust fine-tuning of zero-shot models

Problem: Fine-tuning kills robustness gains of CLIP

→ WiSE-FT (Wortsman 2022): ensemble the weights of the zero-shot and fine-tuned models by simple linear interpolation

$$\text{wse}(x, \alpha) = f(x, (1 - \alpha) \cdot \theta_0 + \alpha \cdot \theta_1)$$





# Dataset Design and Robustness of CLIP

Quality Not Quantity: On the Interaction between Dataset Design and Robustness of CLIP  
Nguyen (2023)

Web-crawled datasets have led to remarkable generalization capabilities in recent image-text models such as CLIP or Flamingo, but little is known about the dataset creation processes

- Reproducibility
- Potential presence of harmful content
- **Hard to identify effective methods for assembling pre-training datasets**

**(i)** How much do different web data sources vary in their induced robustness?

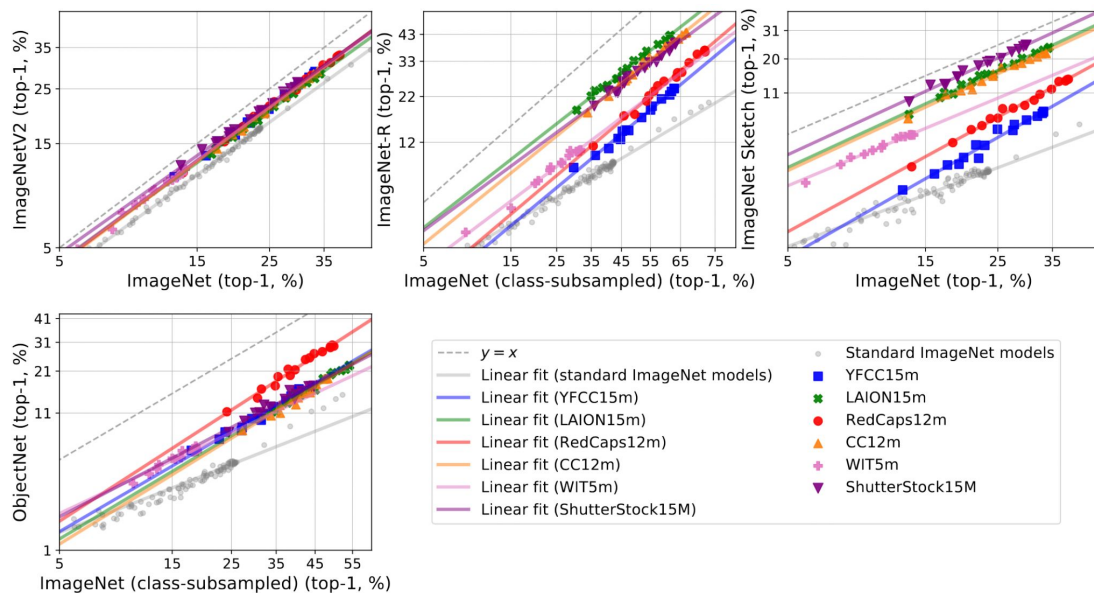
**(ii)** Do dataset combinations lead to better robustness?

(iii) Can filtering with an existing image-text model improve data quality?

# Dataset Design and Robustness of CLIP

**Q:** How much do different web data sources vary in their induced robustness?

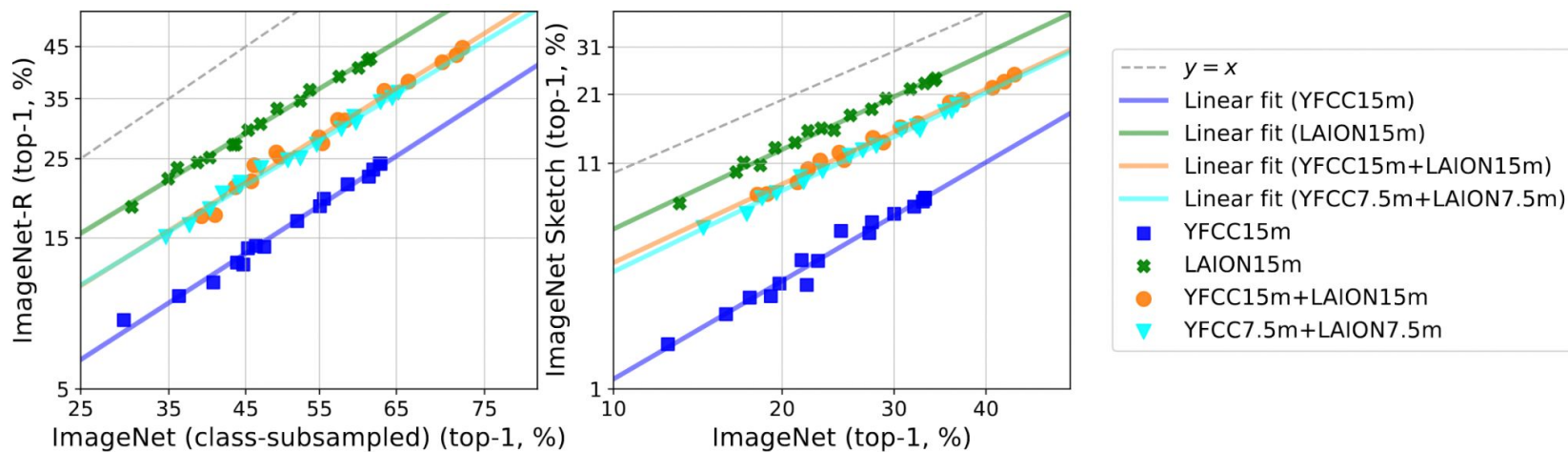
**A:** Performance (both in terms of accuracy and the slope of the linear trend) of the pre-training data varies widely across shifts, with no single data source dominating.



# Dataset Design and Robustness of CLIP

**Q:** Do dataset combinations lead to better robustness?

**A:** Combining multiple sources does not necessarily yield better models, but rather dilutes the robustness of the best individual data source.

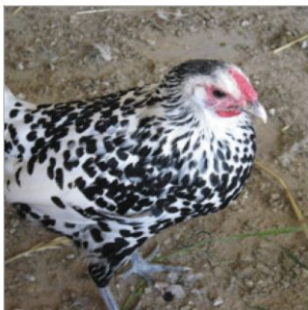


# Visual Classification via Description from LLM

By only using the category name, FSL w/ CLIP neglects to use rich context information available via language

- Gives no intermediate understanding of why a category is chosen
- Provides no mechanism for adjusting the criteria used towards this decision.

Menon & Vondrick (2022) use class descriptions from LLMs classify based on descriptive features

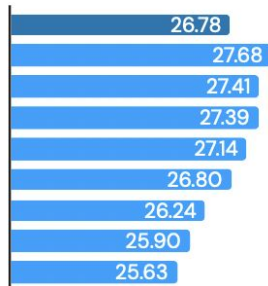


Our top prediction: **Hen**

and we say that because...

Average

- two legs
- red, brown, or white feathers
- a small body
- a small head
- two wings
- a tail
- a beak
- a chicken

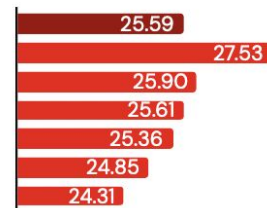


CLIP's top prediction: **Dalmatian**

but we don't say that because...

Average

- black or liver-colored spots
- erect ears
- long legs
- short, stiff hair
- a long, tapering tail
- a long, slender muzzle



# Visual Classification via Description from LLM

Richer class descriptions can help mitigate bias!



Figure 6: (left) CLIP only compares to the word ‘wedding’, yielding biased results – it only correctly recognizes the first row. The descriptor-based approach provides a way to address the bias, by expanding the initial set of descriptors (only the top) to be more inclusive with prior knowledge. (right) Modifying the descriptors to be more inclusive causes accuracy to significantly improve on sub-groups.

## Robustness in Modern NLP

Up until now, we have focused on robustness in modern computer vision  
→What about Natural Language Processing?

Modern NLP is focused on zero-shot and few-shot generalization via a paradigm called **In-Context Learning** applied to **large language models**  
→popularized by GPT-3 (Brown 2021)  
→language model can perform arbitrary tasks!

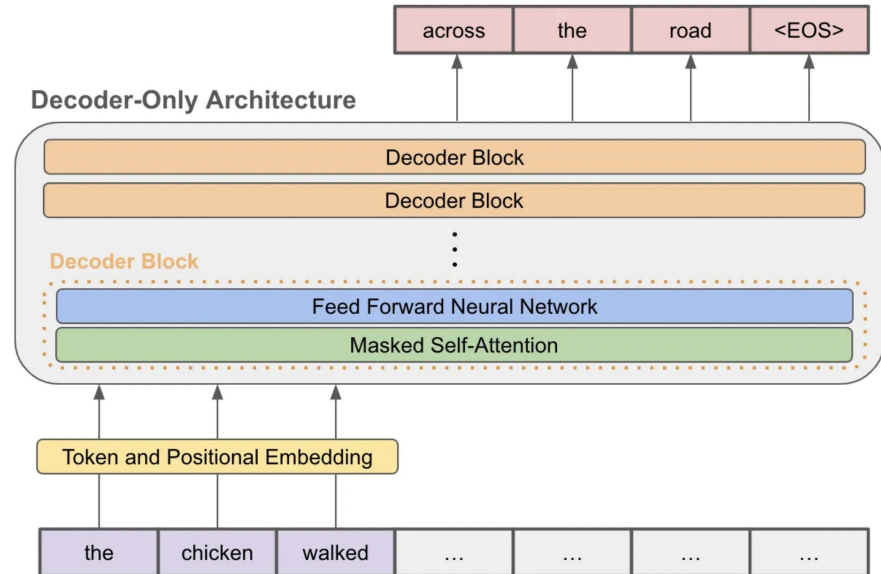
# Language Modeling

Objective: Predict most likely word conditioned on some input string

$$p(x) = \prod_{i=1}^n p(s_i | s_1, \dots, s_{i-1})$$

**Generative** language models are trained on massive corpora to predict the next word

Language is generated left-to-right, one word at a time (e.g. GPT family)



# In Context Learning

Learn to perform many tasks without any gradient updates and given zero or a few examples

Prompt components:

- task description
- examples
- query

“Learn” the task being performed from prompt

- Relevant context
- Label space
- Answer format
- Input-output correspondence

---

## Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

---

## One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```



# In Context Learning Examples

## Sentiment

Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // \_\_\_\_\_



## Topic

Circulation revenue has increased by 5% in Finland. // Finance

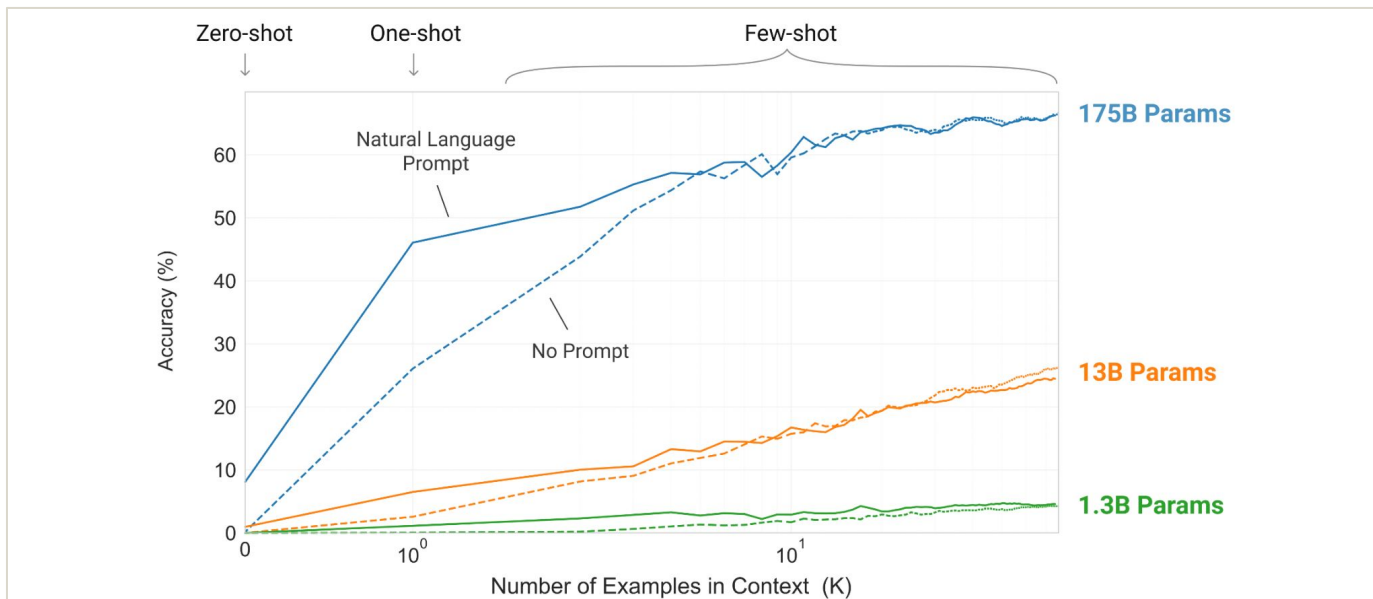
They defeated ... in the NFC Championship Game. // Sports

Apple ... development of in-house chips. // Tech

The company anticipated its operating profit to improve. // \_\_\_\_\_



# GPT-3 Performance



**Figure 1.2: Larger models make increasingly efficient use of in-context information.** We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper “in-context learning curves” for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

# Instruction Tuning

**CLIP:** Zero-shot across different object classes via language embedding.

**Instruction Tuning:** Zero-shot across different *tasks* via language.

## Finetune on many tasks (“instruction-tuning”)

<u>Input (Commonsense Reasoning)</u>	<u>Input (Translation)</u>
Here is a goal: Get a cool sleep on summer days. How would you accomplish this goal? OPTIONS: <input type="checkbox"/> -Keep stack of pillow cases in fridge. <input type="checkbox"/> -Keep stack of pillow cases in oven.	Translate this sentence to Spanish: The new office building was built in less than three months.
<u>Target</u> keep stack of pillow cases in fridge	<u>Target</u> El nuevo edificio de oficinas se construyó en tres meses.
Sentiment analysis tasks	
Coreference resolution tasks	
...	



## Inference on unseen task type

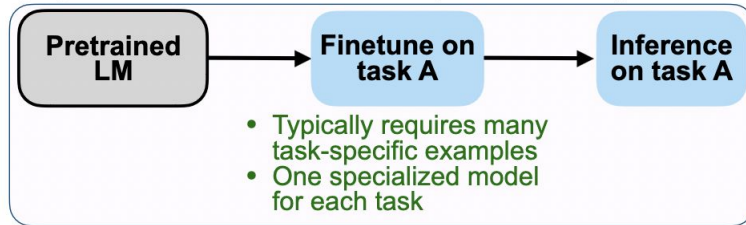
<u>Input (Natural Language Inference)</u>
Premise: At my age you will probably have learnt one lesson. Hypothesis: It's not certain how many lessons you'll learn by your thirties. Does the premise entail the hypothesis? OPTIONS: <input type="checkbox"/> -yes <input type="checkbox"/> -it is not possible to tell <input type="checkbox"/> -no
<u>FLAN Response</u> It is not possible to tell

# Instruction Tuning

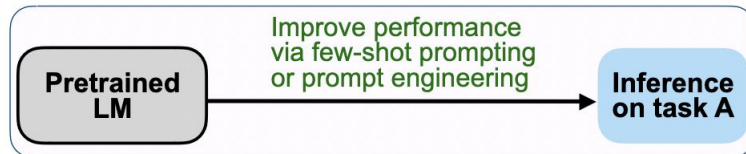
**CLIP:** Zero-shot across different object classes via language embedding.

**Instruction Tuning:** Zero-shot across different *tasks* via language.

## (A) Pretrain–finetune (BERT, T5)



## (B) Prompting (GPT-3)



## (C) Instruction tuning (FLAN)

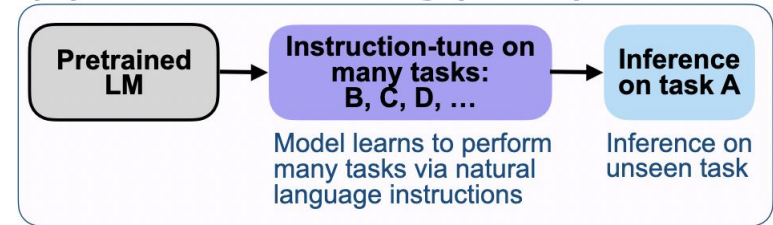


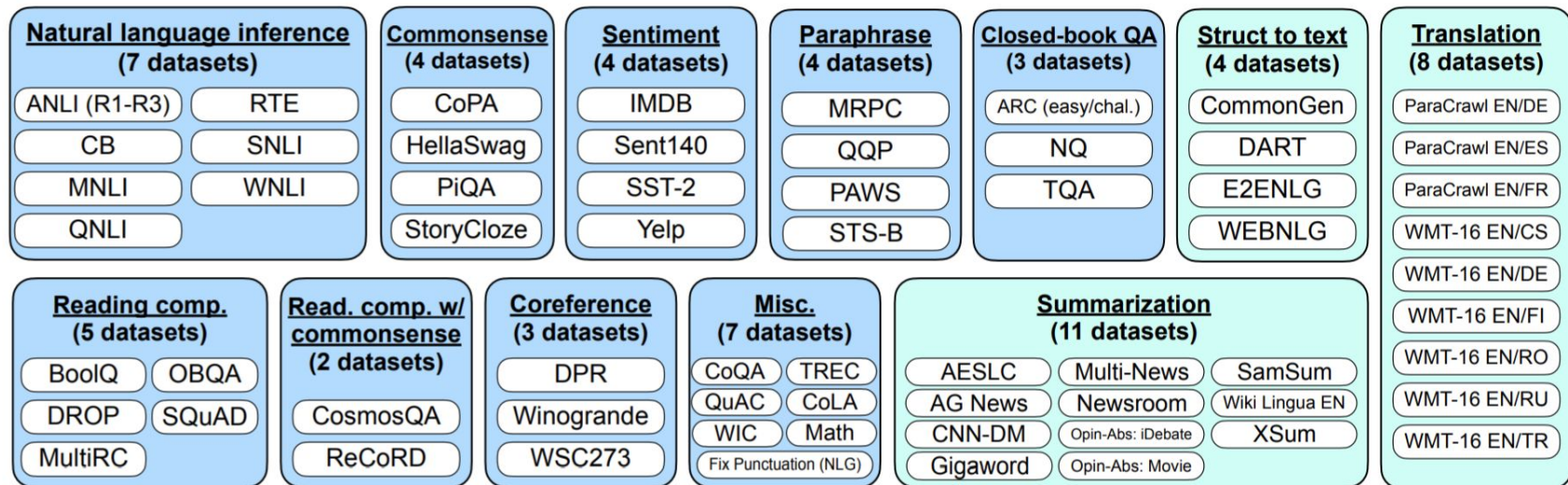
Figure 2: Comparing instruction tuning with pretrain–finetune and prompting.

# How does this relate to robustness?

**CLIP:** zero-shot learning to avoid dataset biases

**Instruction-tuning:** zero-shot learning to avoid task biases

Define a **task** with a set of datasets, split into **train and test tasks**



# Instruction Tuning

These zero shot models are inherently robust. The key is to make them perform well

Finetune on many tasks (“instruction-tuning”)

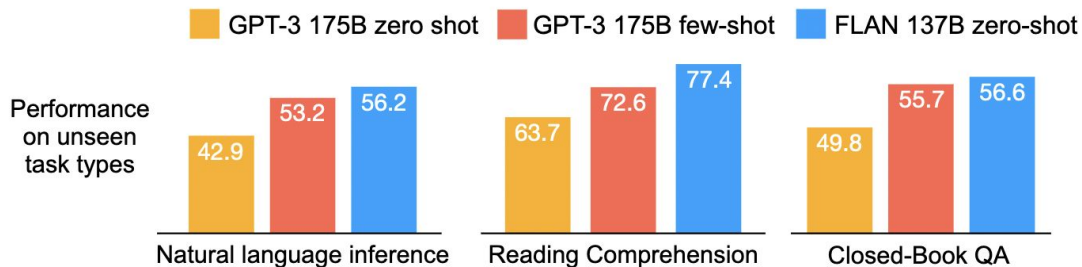
**Input (Commonsense Reasoning)**  
Here is a goal: Get a cool sleep on summer days.  
How would you accomplish this goal?  
OPTIONS:  
 -Keep stack of pillow cases in fridge.  
 -Keep stack of pillow cases in oven.  
**Target**  
keep stack of pillow cases in fridge

**Input (Translation)**  
Translate this sentence to Spanish:  
The new office building was built in less than three months.  
**Target**  
El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks  
Coreference resolution tasks  
...

Inference on unseen task type

**Input (Natural Language Inference)**  
Premise: At my age you will probably have learnt one lesson.  
Hypothesis: It's not certain how many lessons you'll learn by your thirties.  
Does the premise entail the hypothesis?  
OPTIONS:  
 -yes  -it is not possible to tell  -no  
**FLAN Response**  
It is not possible to tell



# Benefits of massive multitasking + zero-shot learning

Remarkably good zero-shot performance now achievable: within 10% of supervised.

	READING COMPREHENSION			CLOSED-BOOK QA			
	BoolQ acc.	MultiRC F1	OBQA acc.	ARC-e acc.	ARC-c acc.	NQ EM	TQA EM
Supervised model	91.2 <sup>a</sup>	88.2 <sup>a</sup>	85.4 <sup>a</sup>	92.6 <sup>a</sup>	81.1 <sup>a</sup>	36.6 <sup>a</sup>	60.5 <sup>a</sup>
Base LM 137B zero-shot	81.0	60.0	41.8	76.4	42.0	3.2	21.9
· few-shot	79.7	59.6	50.6	80.9	49.4	22.1	63.3
GPT-3 175B zero-shot	60.5	72.9	57.6	68.8	51.4	14.6	64.3
· few-shot	77.5	74.8	65.4	70.1	51.5	29.9	71.2
FLAN 137B zero-shot							
- average template	80.2 $\blacktriangle$ 2.7 std=3.1	74.5 $\uparrow$ 2.4 std=3.7	77.4 $\blacktriangle$ 12.0 std=1.3	79.5 $\blacktriangle$ 8.6 std=0.8	61.7 $\blacktriangle$ 10.2 std=1.4	18.6 $\uparrow$ 4.0 std=2.7	66.5 $\uparrow$ 2.2 std=2.6
- best dev template	82.9 $\blacktriangle$ 5.4	77.5 $\blacktriangle$ 2.7	78.4 $\blacktriangle$ 13.0	79.6 $\blacktriangle$ 8.7	63.1 $\blacktriangle$ 11.6	20.7 $\uparrow$ 6.1	68.1 $\uparrow$ 3.8

Table 2: Results on reading comprehension and closed-book question answering. For FLAN, we report both the average of up to ten templates, as well as the best dev template. The triangle  $\blacktriangle$  indicates improvement over few-shot GPT-3. The up-arrow  $\uparrow$  indicates improvement only over zero-shot GPT-3. <sup>a</sup>T5-11B.

# Chain Of Thought Prompting

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models - Wei et al. (2022)

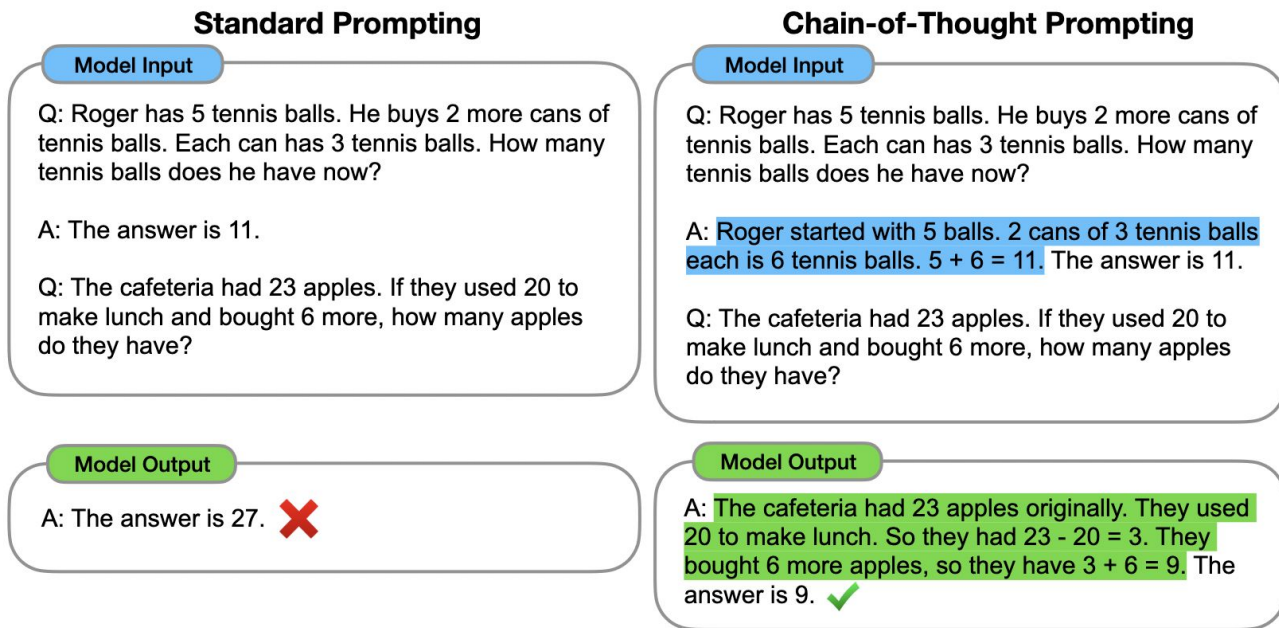


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.



# Reinforcement Learning From Human Feedback

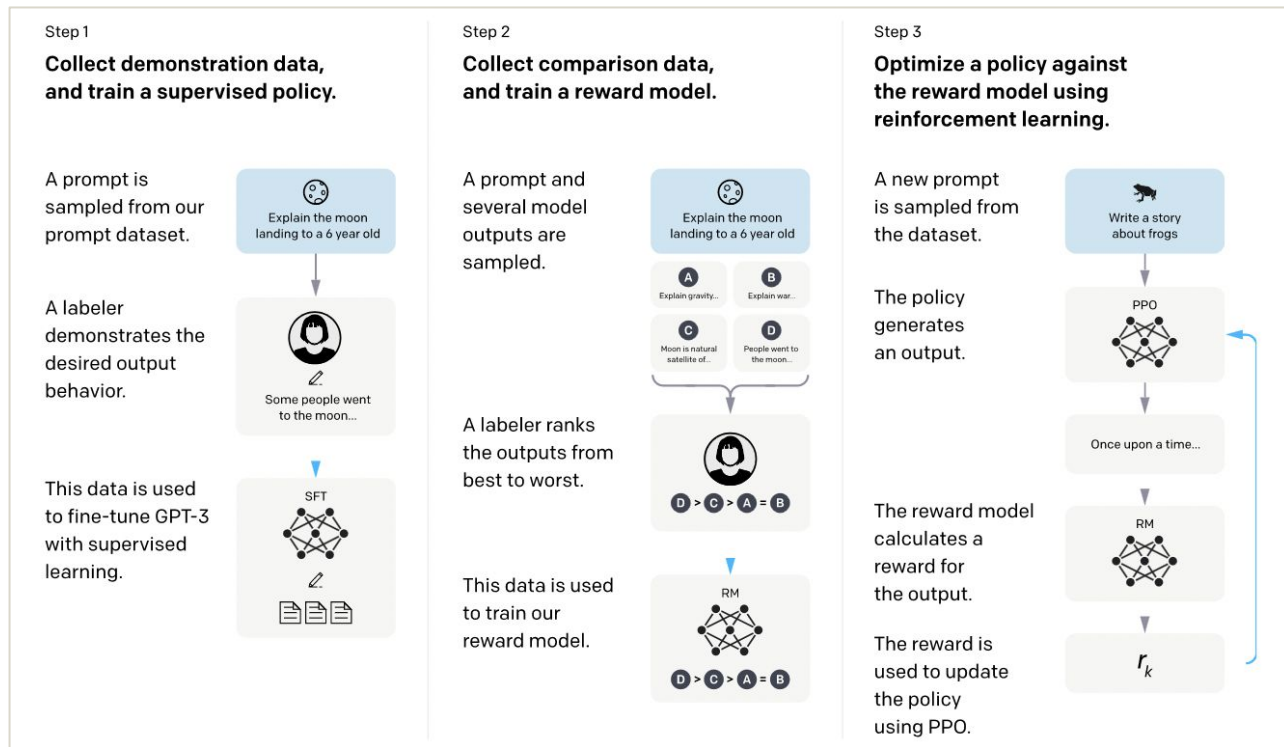
Instruction tuning relies on typical NLP datasets to generate ICL examples

Under RLHF, collect prompts and desired outputs from humans

→ Align with human preferences

→ Key ingredient in ChatGPT, etc

Is RL necessary?



# Multi-modal training with GATO

We can turn image patches into “token embeddings” and apply transformer (ex. ViT) → What if we mix tokens from different modalities (text, vision, joint trajectories, game controls, etc.)?

Everything is a token if you squint hard enough!

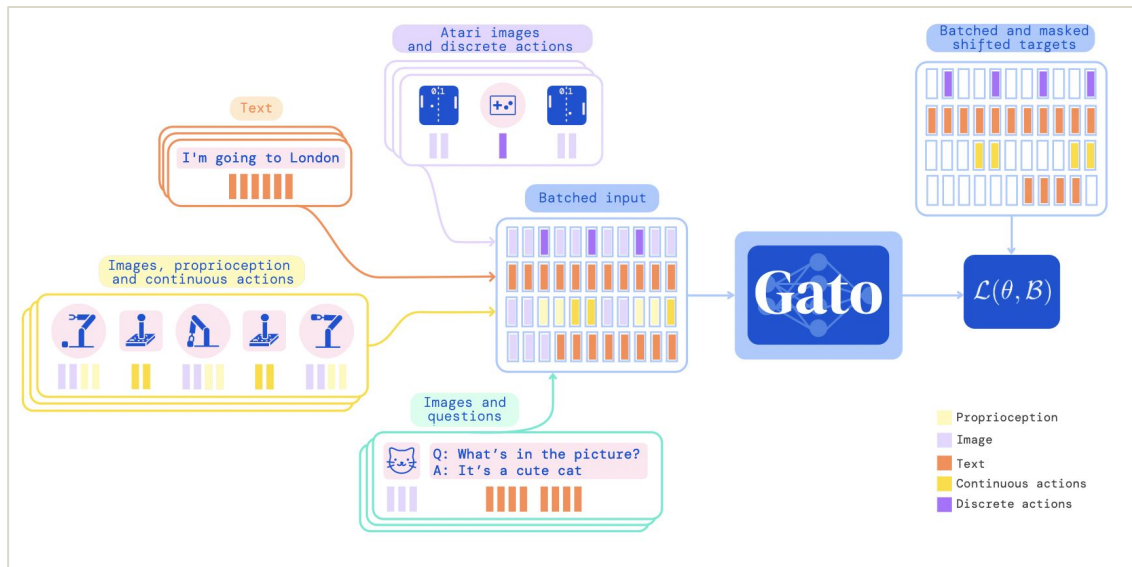
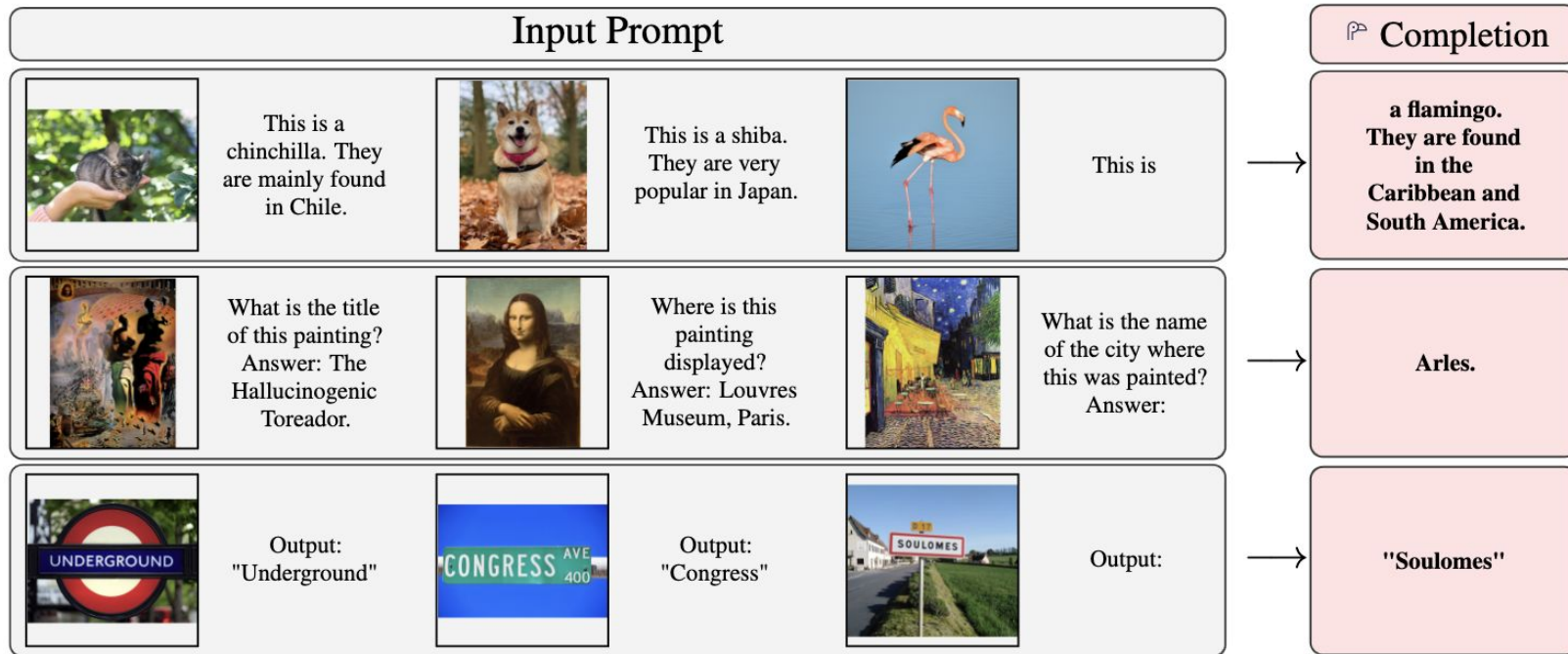







Figure 2: **Training phase of Gato.** Data from different tasks and modalities is serialized into a flat sequence of tokens, batched, and processed by a transformer neural network akin to a large language model. Masking is used such that the loss function is applied only to target outputs, i.e. text and various actions.

# Multi-modal zero-shot learning with Flamingo



# Multi-modal zero-shot learning with Flamingo

 <p>This is a picture of two teddy bears on the moon.</p> <p>What are they doing?</p> <p>They are having a conversation.</p> <p>What object are they using?</p> <p>It looks like a computer.</p> <p>Is this surprising?</p> <p>Yes, it is surprising.</p> <p>Why is this picture surprising to you?</p> <p>I think it is surprising because teddy bears are not usually found on the moon.</p>	 <p>What is the common thing about these three images?</p> <p>They are all flamingos.</p> <p>What is the difference between these three images?</p> <p>The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.</p>	 <p>This is an apple with a sticker on it.</p> <p>What does the sticker say?</p> <p>The sticker says "iPod".</p> <p>Where is the photo taken?</p> <p>It looks like it's taken in a backyard.</p> <p>Do you think it is printed or handwritten?</p> <p>It looks like it's handwritten.</p> <p>What color is the sticker?</p> <p>It's white.</p>	 <p>This is a cityscape. It looks like Chicago.</p> <p>What makes you think this is Chicago?</p> <p>I think it's Chicago because of the Shedd Aquarium in the background.</p>  <p>What about this one? Which city is this and what famous landmark helped you recognise the city?</p> <p>This is Tokyo. I think it's Tokyo because of the Tokyo Tower.</p>
---	---	---	--

# Multi-modal zero-shot learning with GPT-4

---

## Example of GPT-4 visual input:

---

User      What is funny about this image? Describe it panel by panel.



Source: <https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/>

GPT-4      The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

---

**Table 3.** Example prompt demonstrating GPT-4's visual input capability. The prompt consists of a question about an image with multiple panels which GPT-4 is able to answer.

# Key commonalities between CLIP and zero-shot LLMs

## Key takeaways

- Zero-shot models are *inherently* robust.
- One path to building effective robust models is to build effective zero-shot ones
- Language is a common interface across tasks
  - › Progress in large language models is causing an explosion in zero-shot learning progress across vision, robotics, etc.