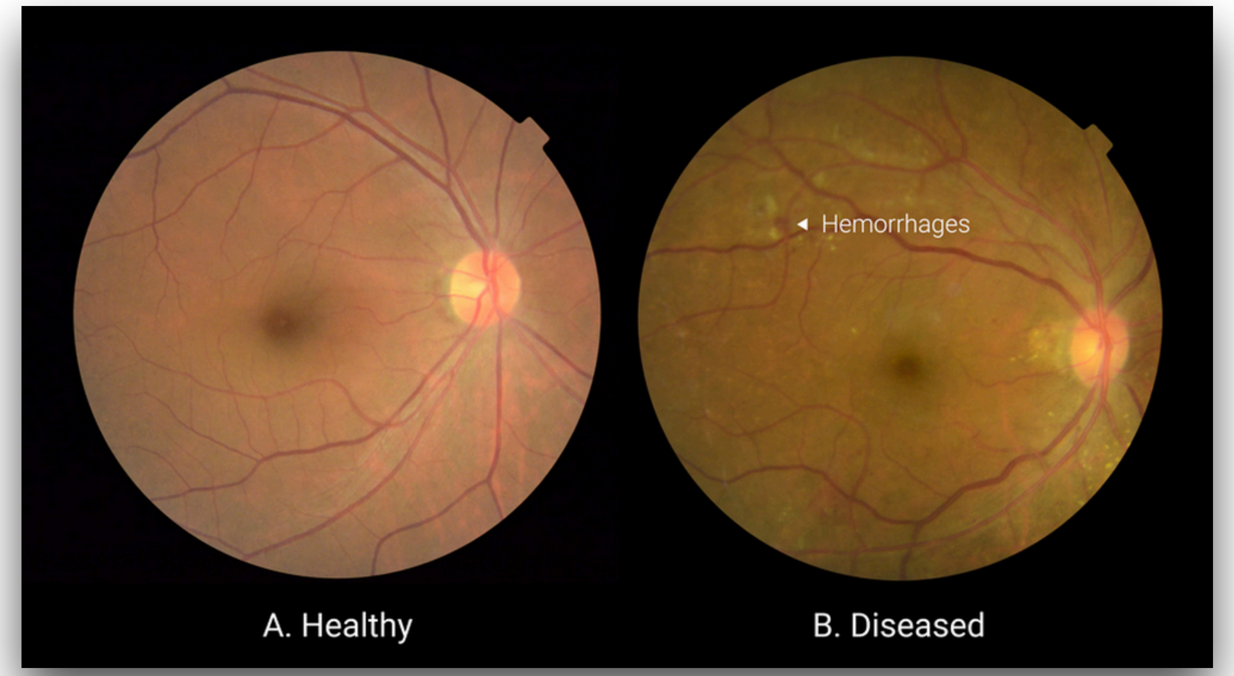
Distribution shift, multimodal models

493 / 599   May 23 2023

Ludwig Schmidt

Challenges on the way from research to the real world

Transportation



Health care

Robotics

Chat assistants

➡ Need **reliable** machine learning

[Deng, Dong, Socher, Li, Li, Fei-Fei'09]
[Russakovsky, Deng, Su, Krause, Satheesh, Ma, Huang, Karpathy, Khosla, Bernstein, Berg, Fei-Fei'15]

Robustness on ImageNet

Lots of progress on ImageNet over the past 10 years, but models are still not robust.

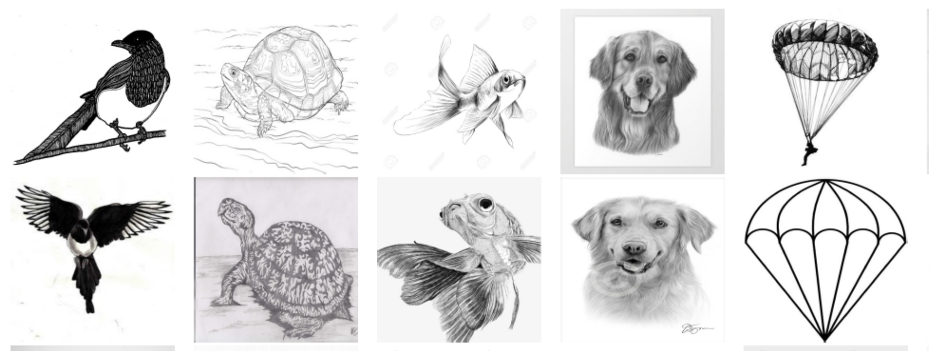
Evaluation: **new test sets**

ImageNetV2

[Recht, Roelofs, Schmidt, Shankar ’19]

ObjectNet

[Barbu, Mayo, Alverio, Luo, Wang, Gutfreund, Tenenbaum, Katz ’19]

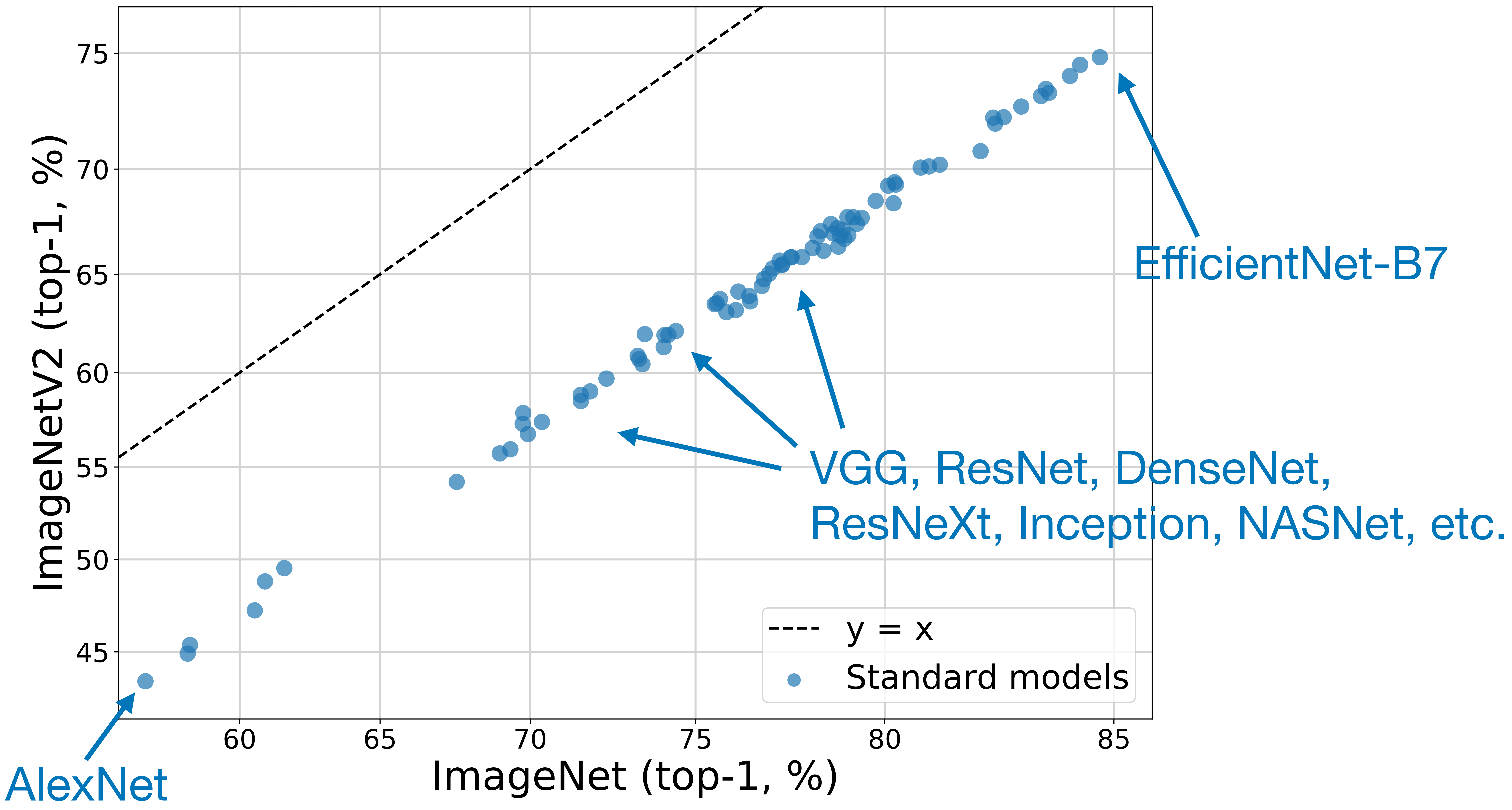


ImageNet-Sketch

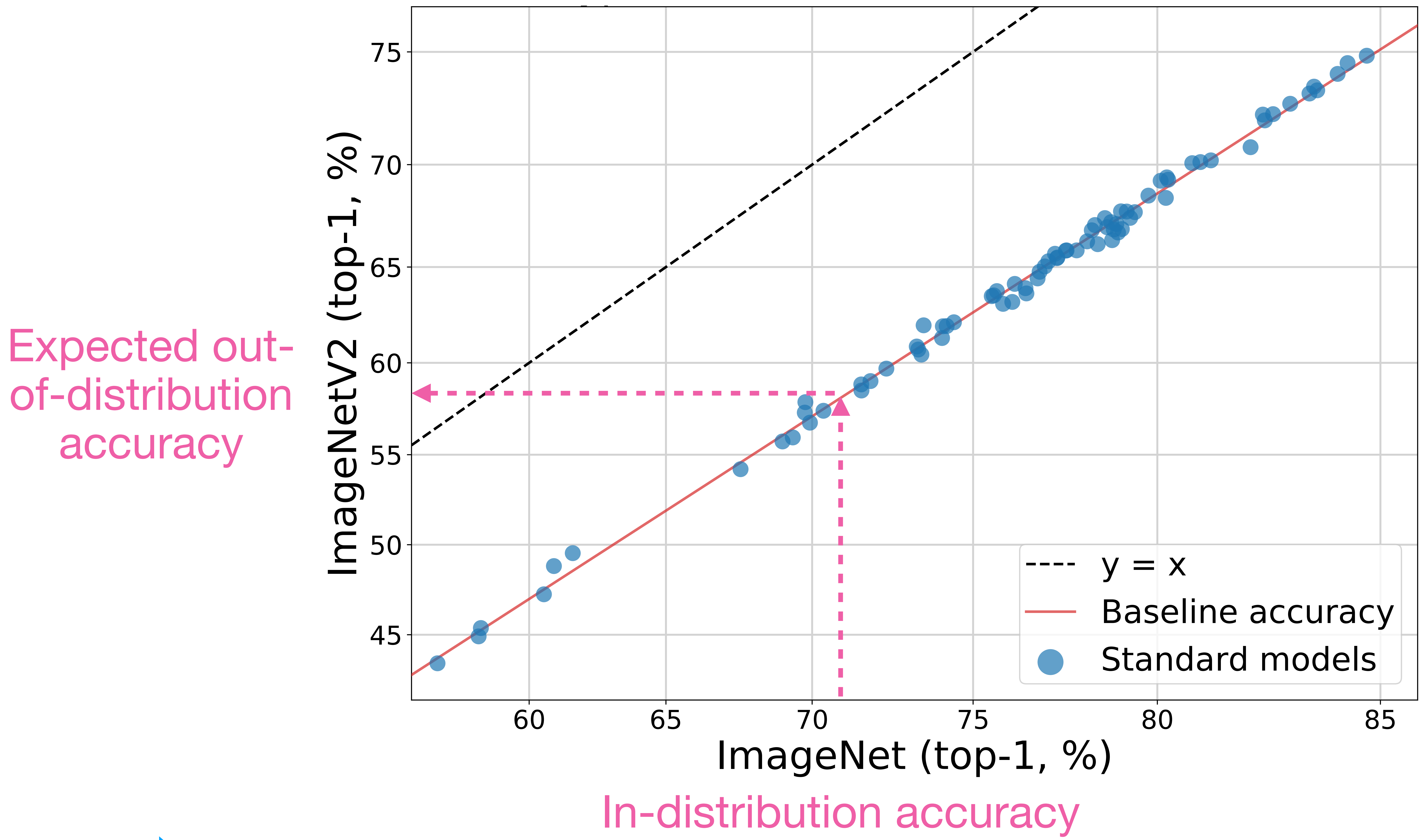
[Wang, Ge, Lipton, Xing ’19]

ImageNet-R

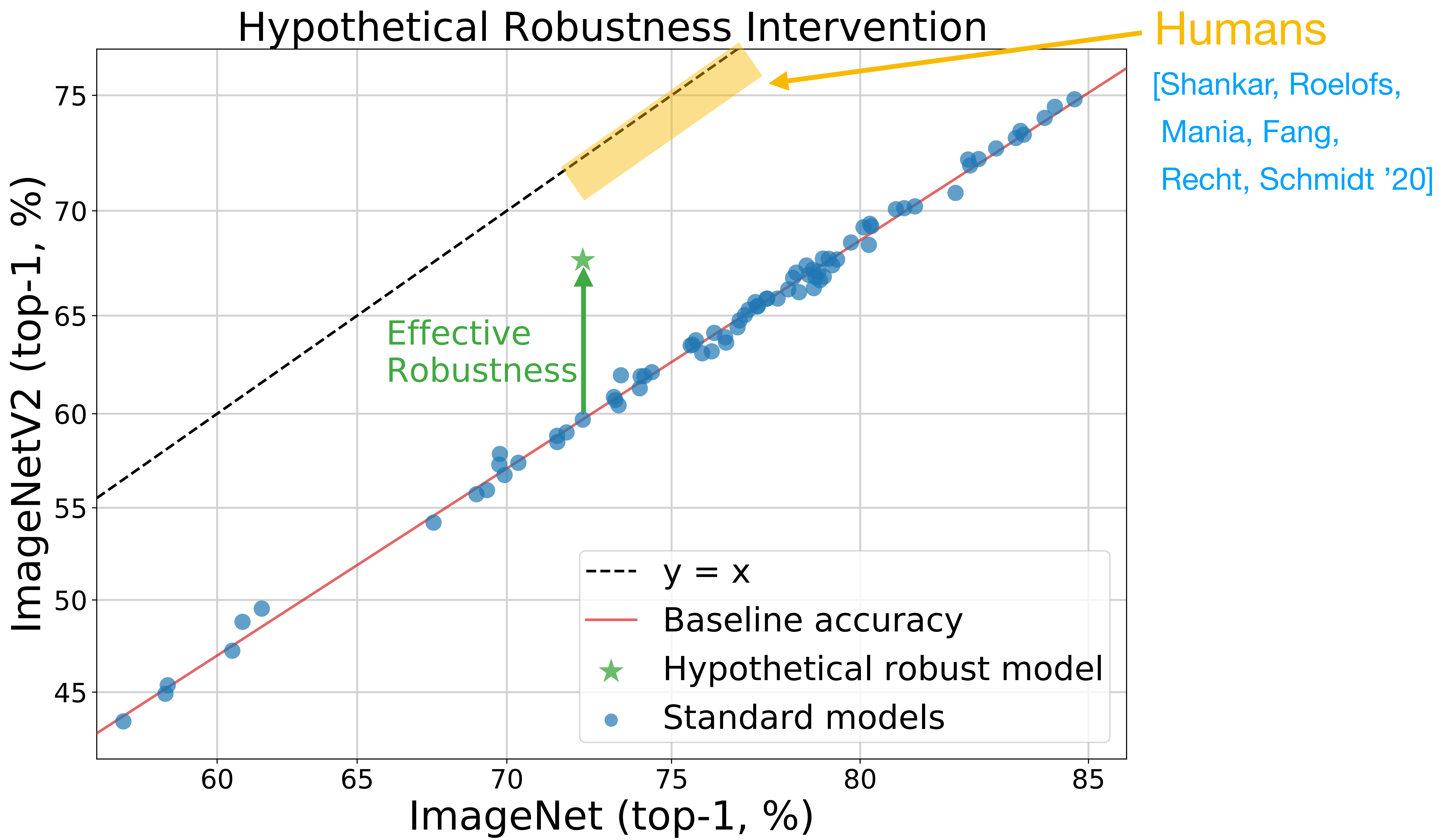
[Hendrycks, Basart, Mu, Kadavath, Wang, Dorundo, Desai, Zhu, Parajuli, Guo, Song, Steinhardt, Gilmer ’20]



[Taori, Dave, Shankar, Carlini, Recht, Schmidt '20]

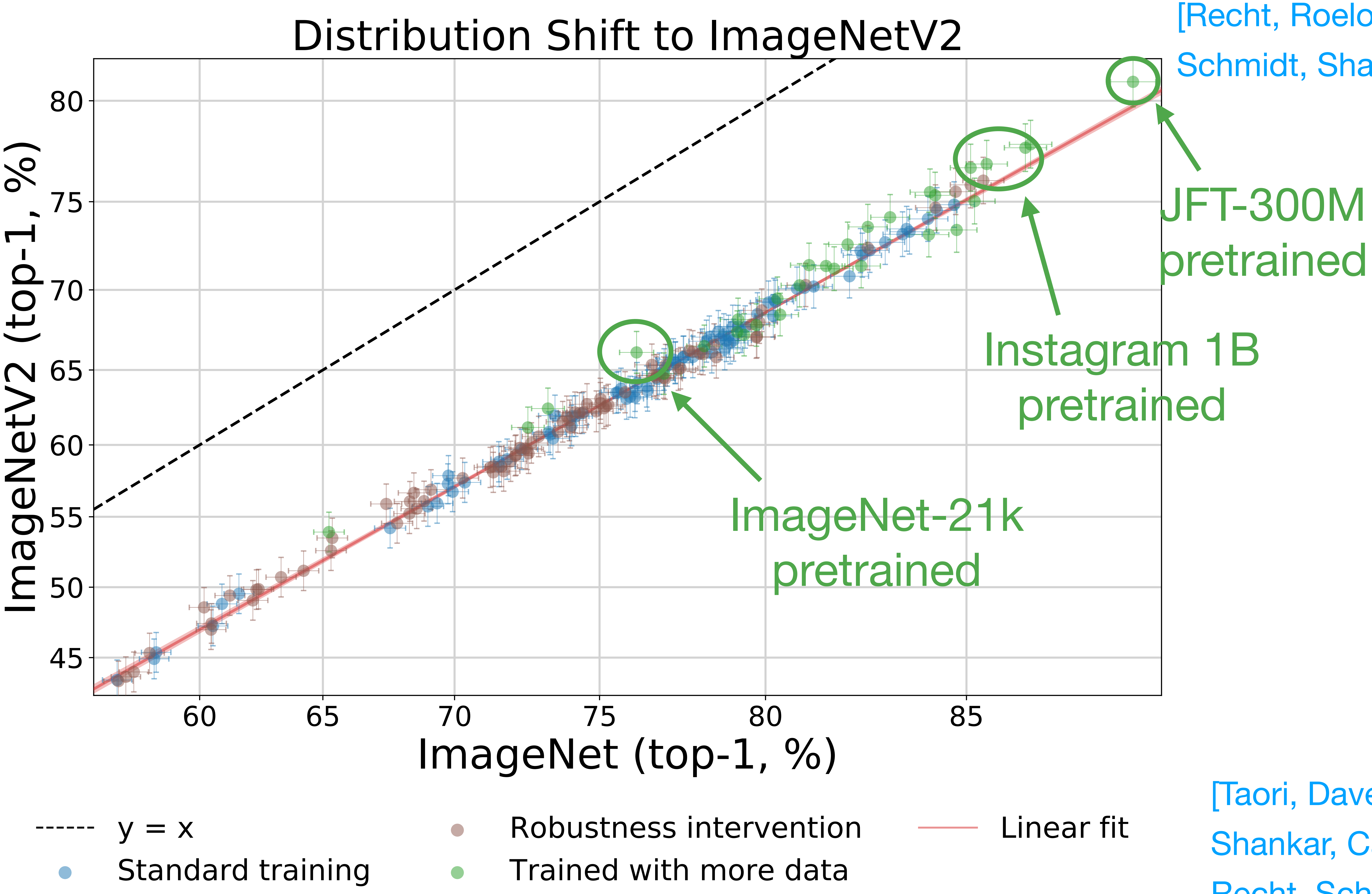


Baseline **out-of-distribution accuracy** from **in-distribution accuracy.**



Do current robustness interventions achieve effective robustness?

[Recht, Roelofs, Schmidt, Shankar '19]



[Taori, Dave, Shankar, Carlini, Recht, Schmidt '20]

Only training on (a lot) **more data** gives a small amount of effective robustness.

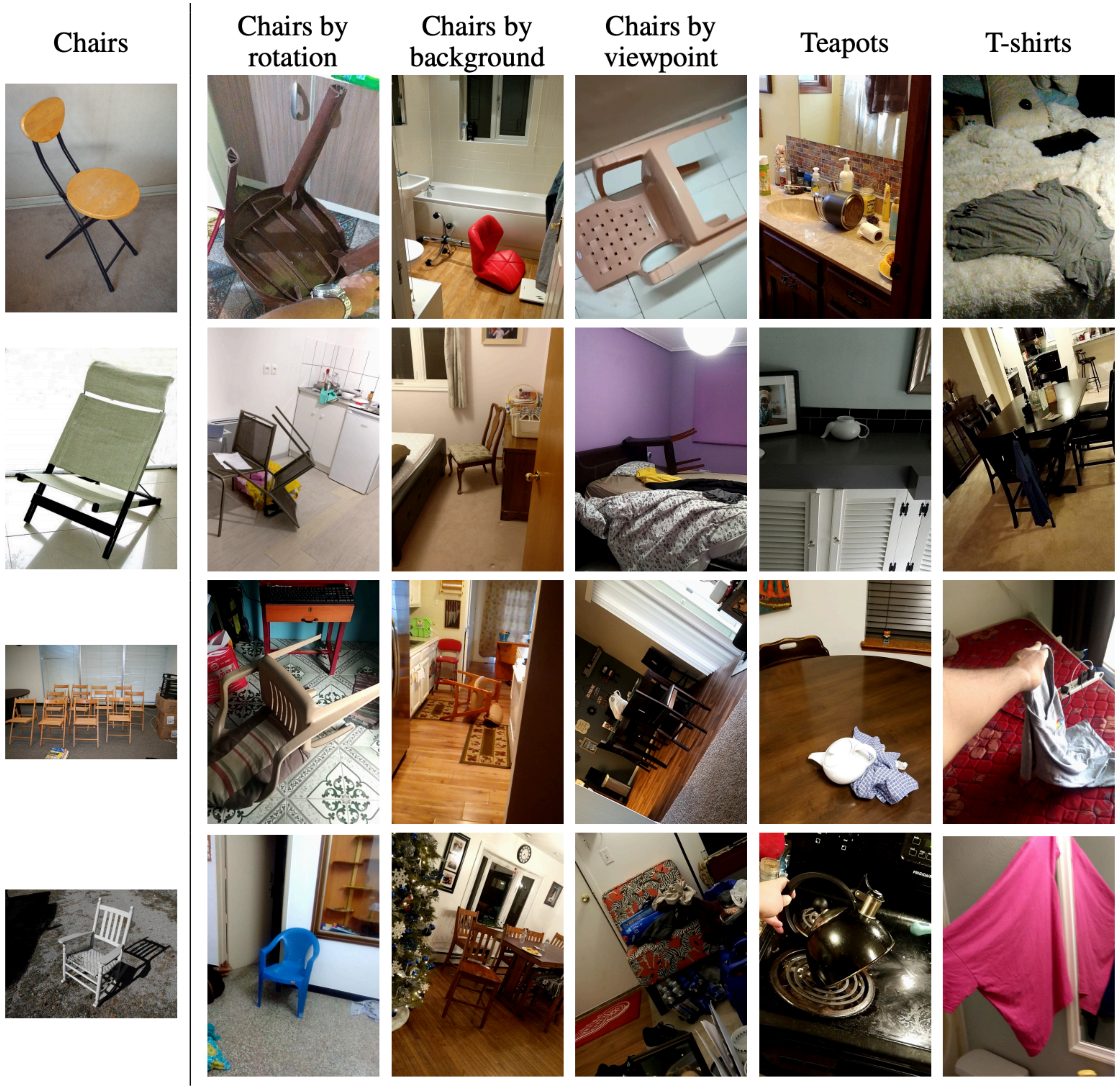
ObjectNet: Objects in Unusual Positions

ImageNet

Mainly object-centric and clean images

(collected from Flickr)

ObjectNet

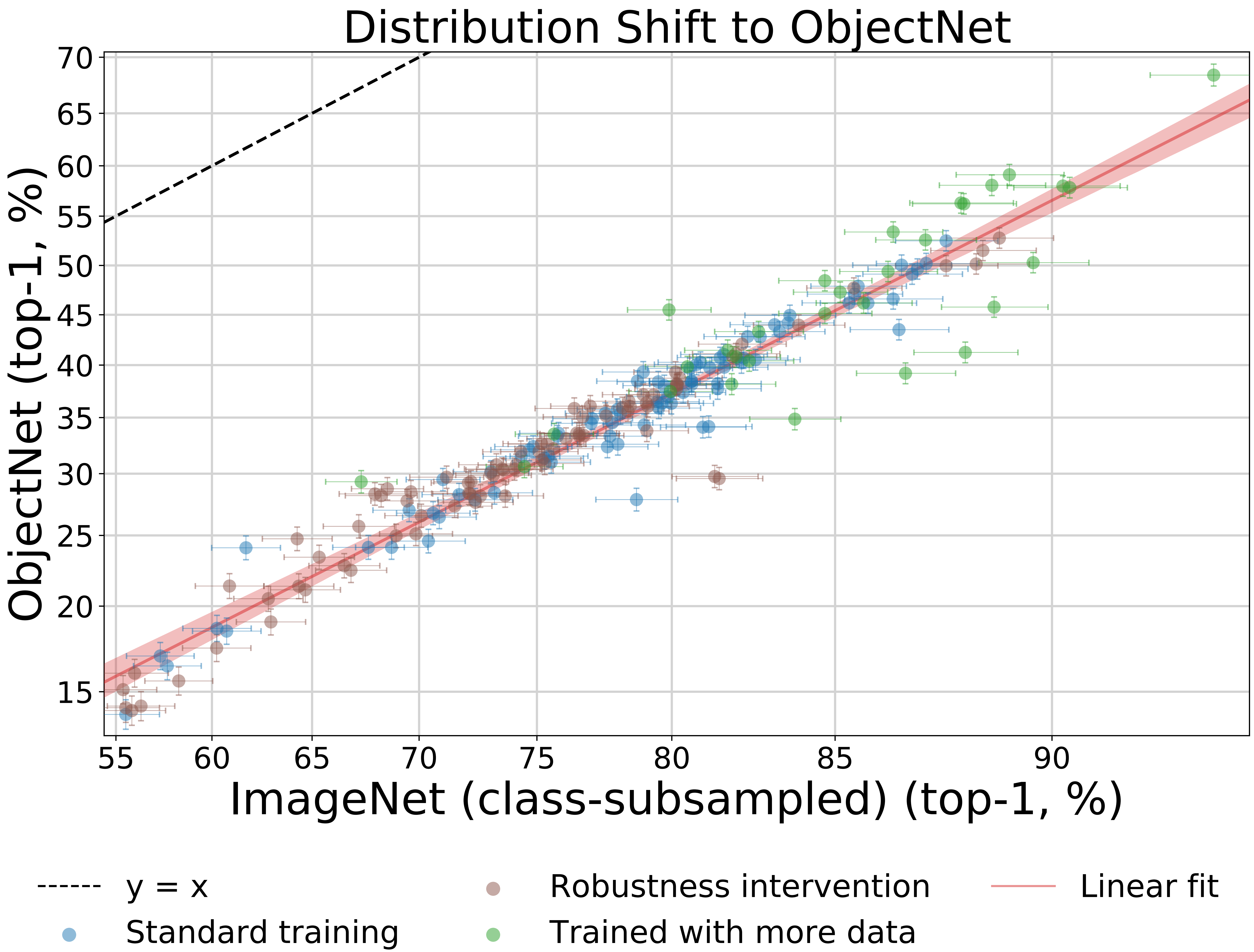


Intentionally randomized:

- object poses
- locations
- etc.

(collected via specific crowd worker annotations)

[Barbu, Mayo, Alverio, Luo, Wang, Gutfreund, Tenenbaum, Katz ’19]

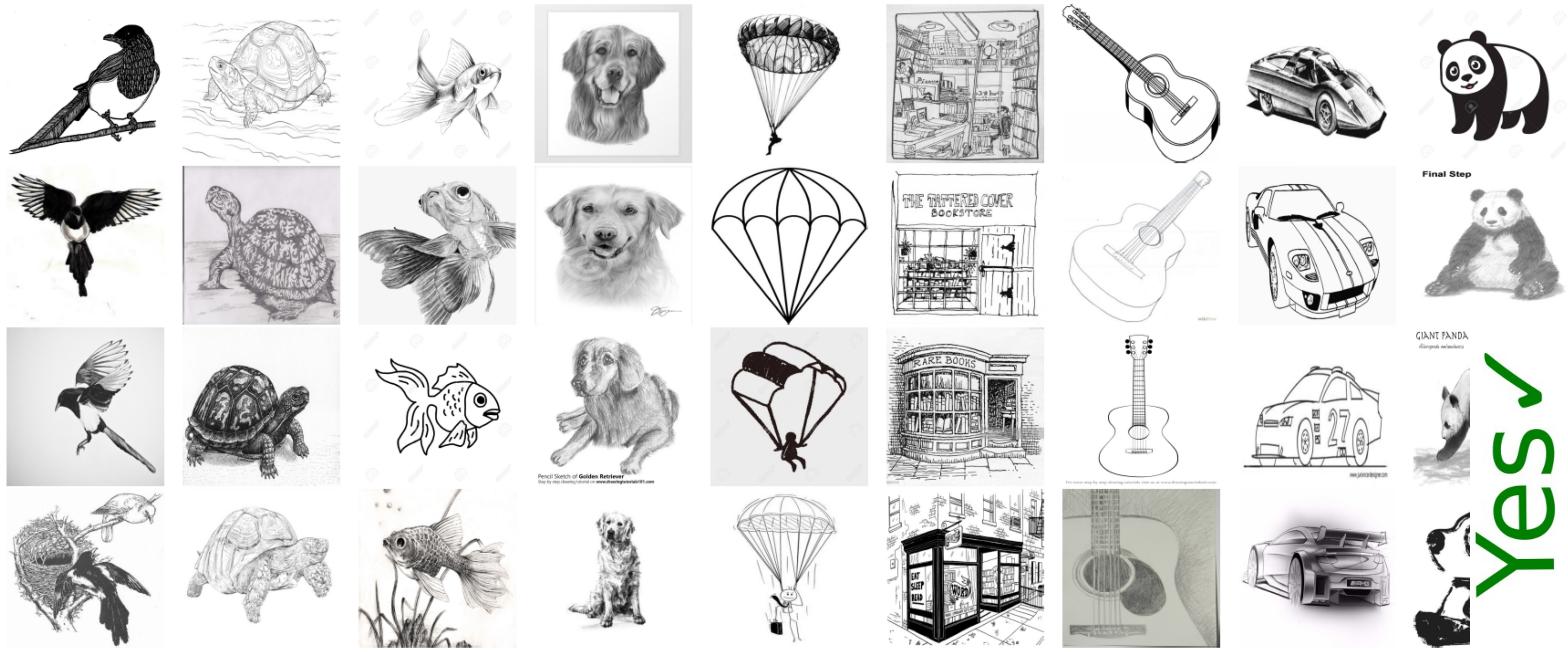


[Barbu, Mayo, Alverio, Luo, Wang, Gutfreund, Tenenbaum, Katz ’19]

Same trend: only **more data** gives effective robustness.

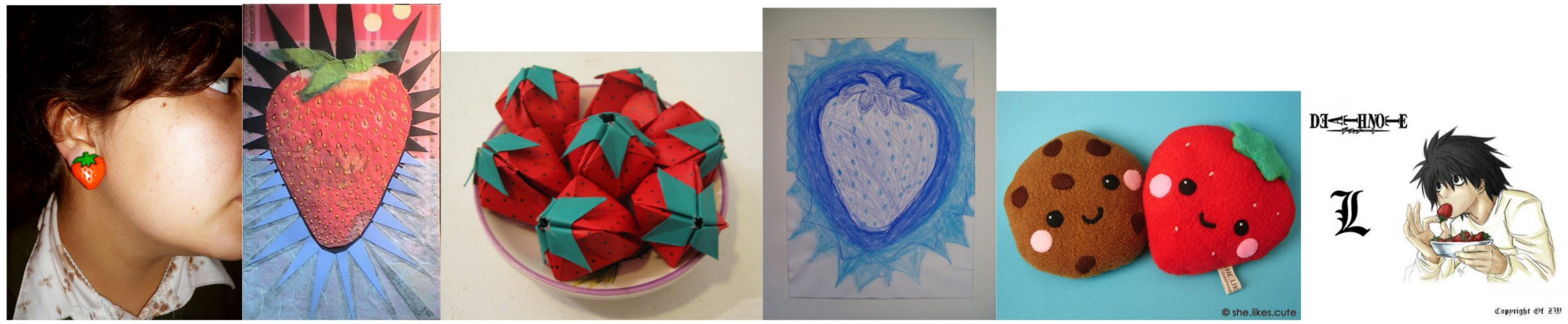
[Taori, Dave, Shankar, Carlini, Recht, Schmidt ’20]

ImageNet-Sketch & ImageNet-R



[Wang, Ge, Lipton, Xing ’19]

Yes ✓



No ×

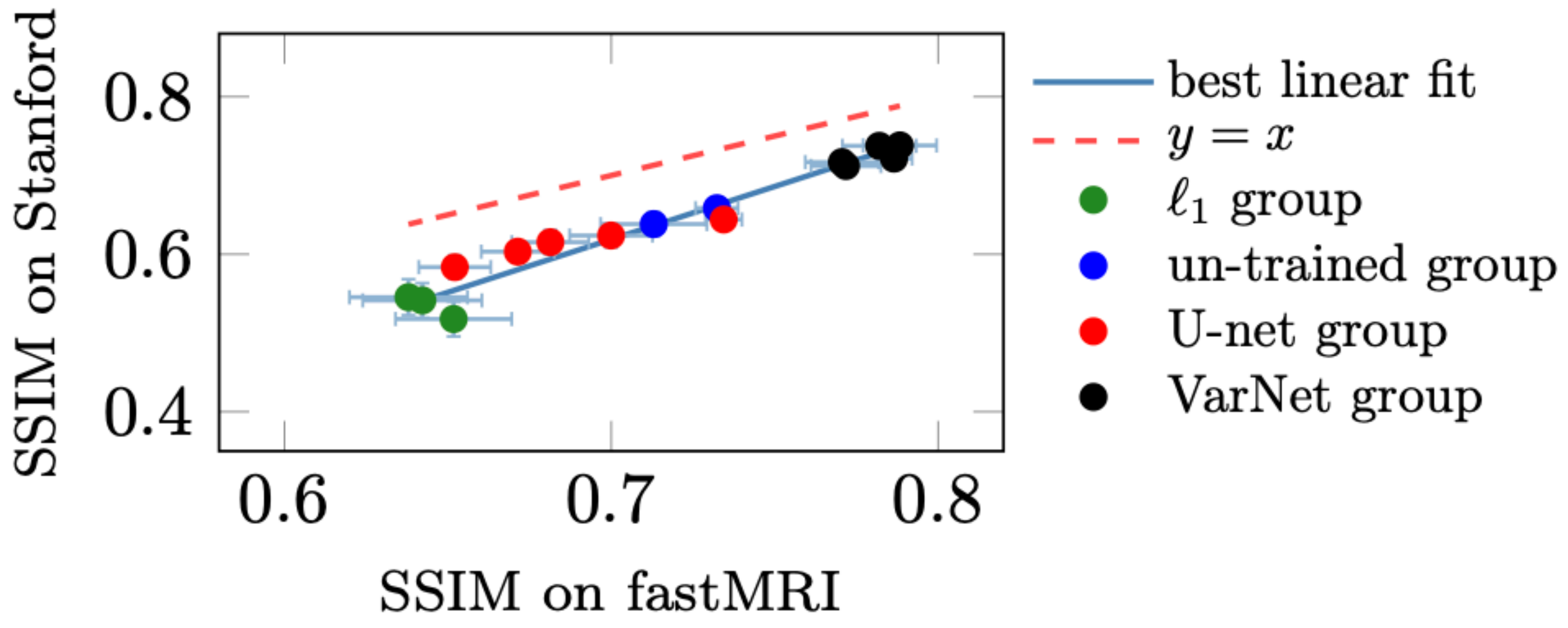
[Hendrycks, Basart, Mu, Kadavath, Wang, Dorundo, Desai, Zhu, Parajuli, Guo, Song, Steinhardt, Gilmer ’20]

Some gains from **adv. training** and data augmentation. **More data** models still best.

Beyond image classification

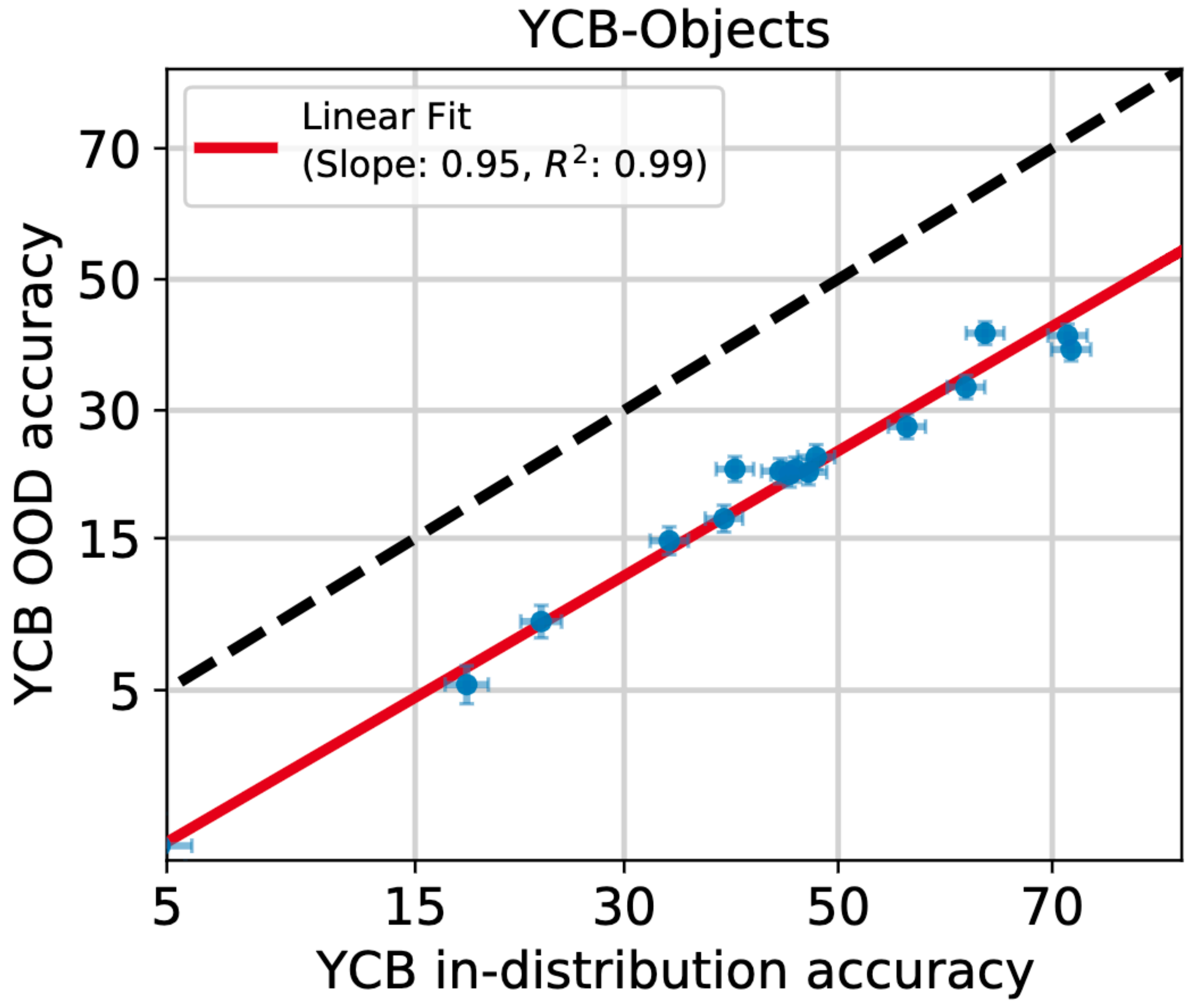
Similar phenomena appear in other computer vision problems:

MRI reconstruction



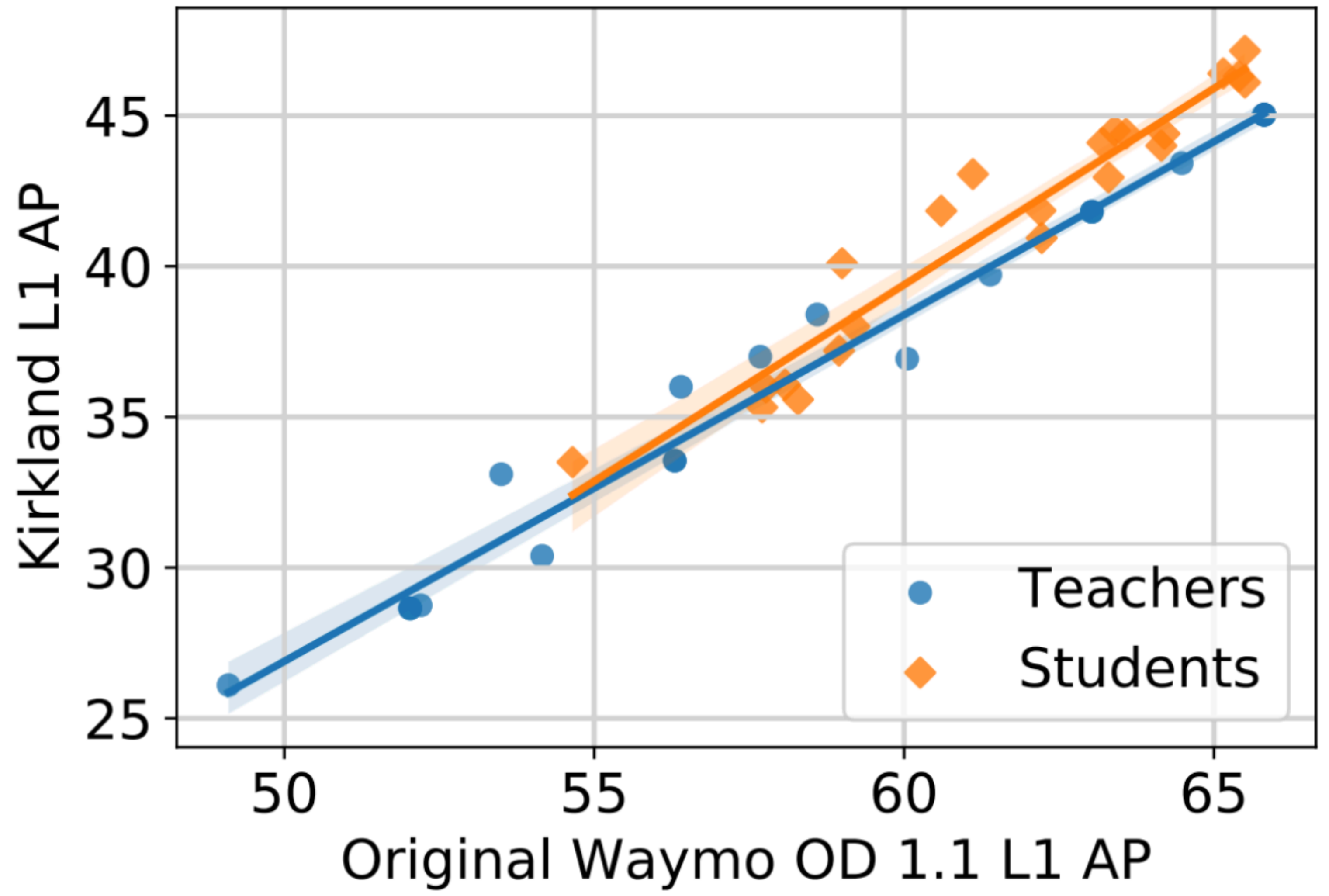
[Darestani, Chaudhari, Heckel '21]

Pose estimation



[Miller, Taori, Raghunathan, Sagawa, Koh, Shankar, Liang, Carmon, Schmidt '21]

Object detection

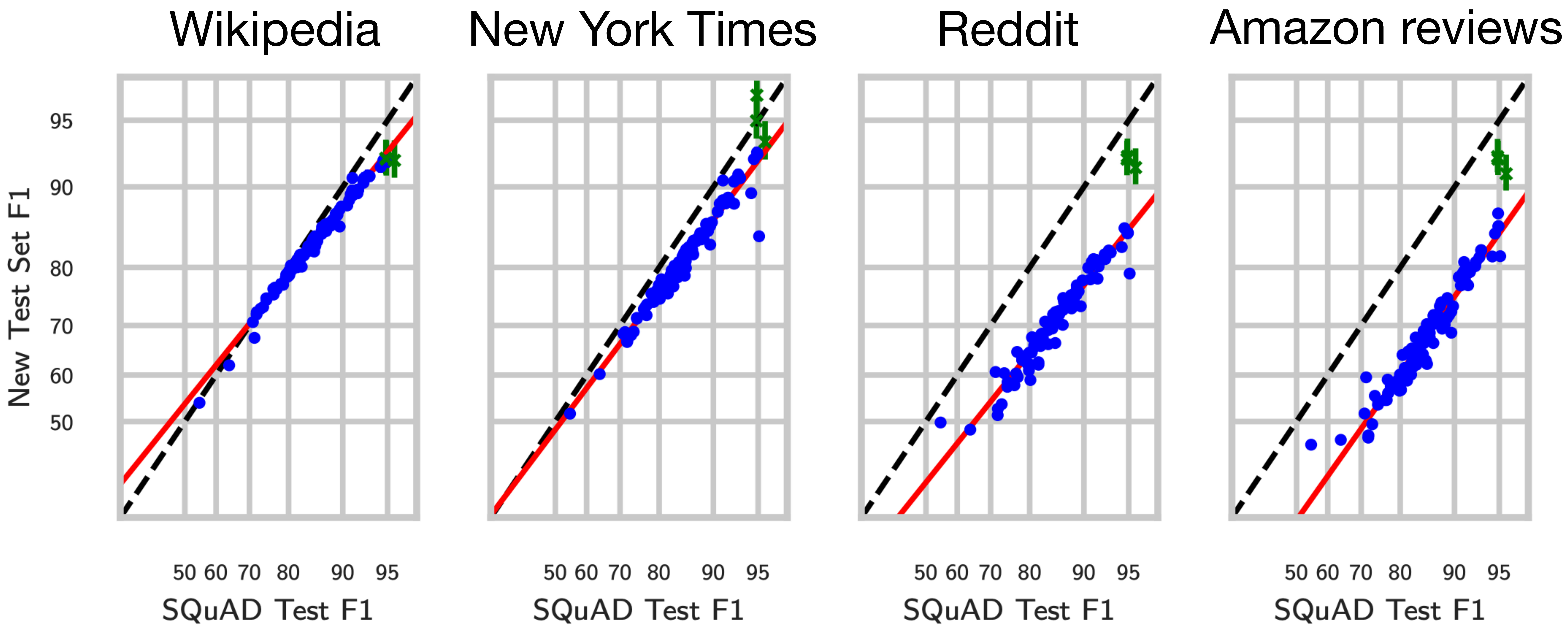


[Roelofs, Caine, Vasudevan, Ngiam, Chen, Shlens '21]

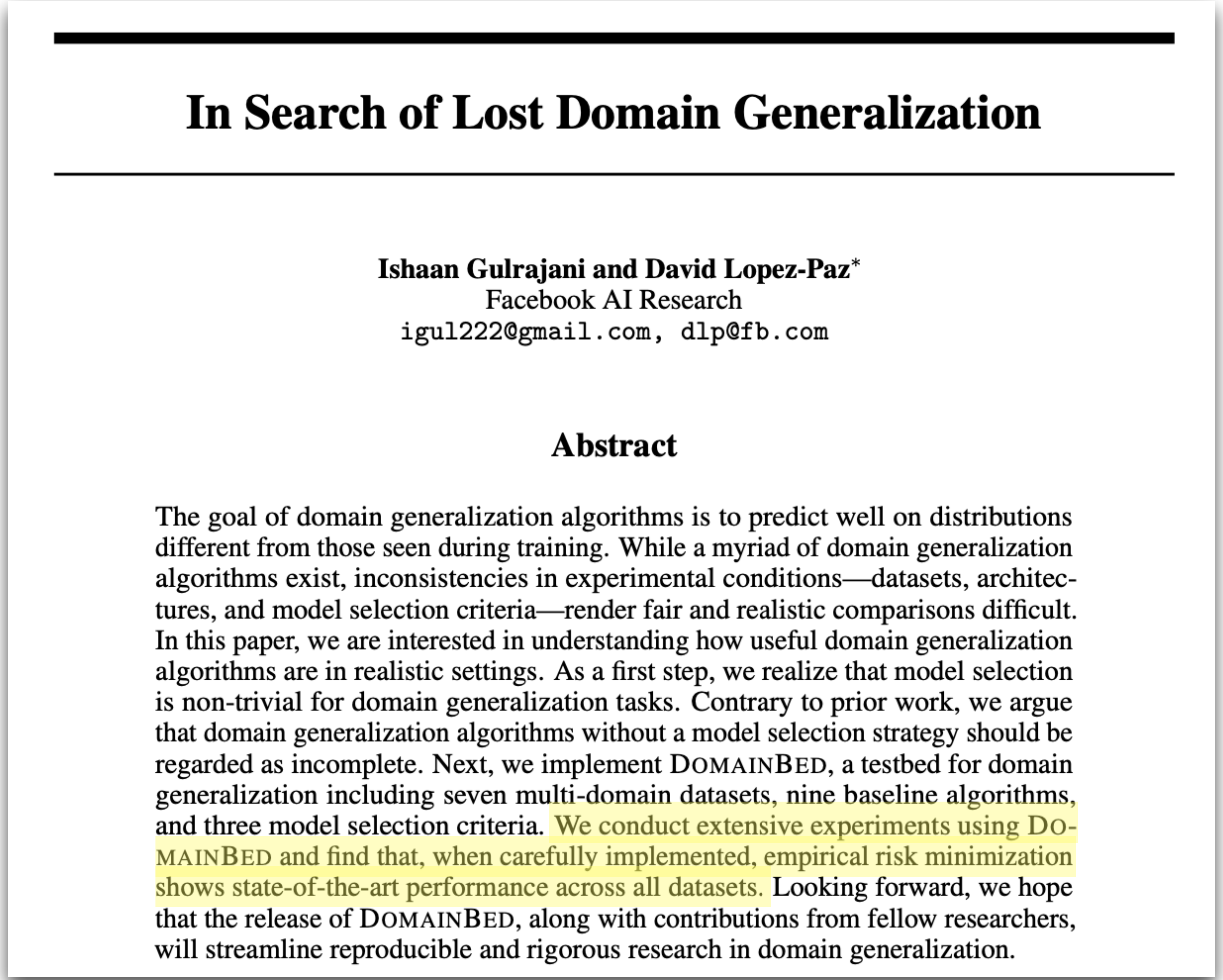
Beyond computer vision

SQuAD (Stanford Question Answering Dataset): question answering on paragraphs

➡ Similar trends in natural language processing. [Miller, Krauth, Recht, Schmidt '20]



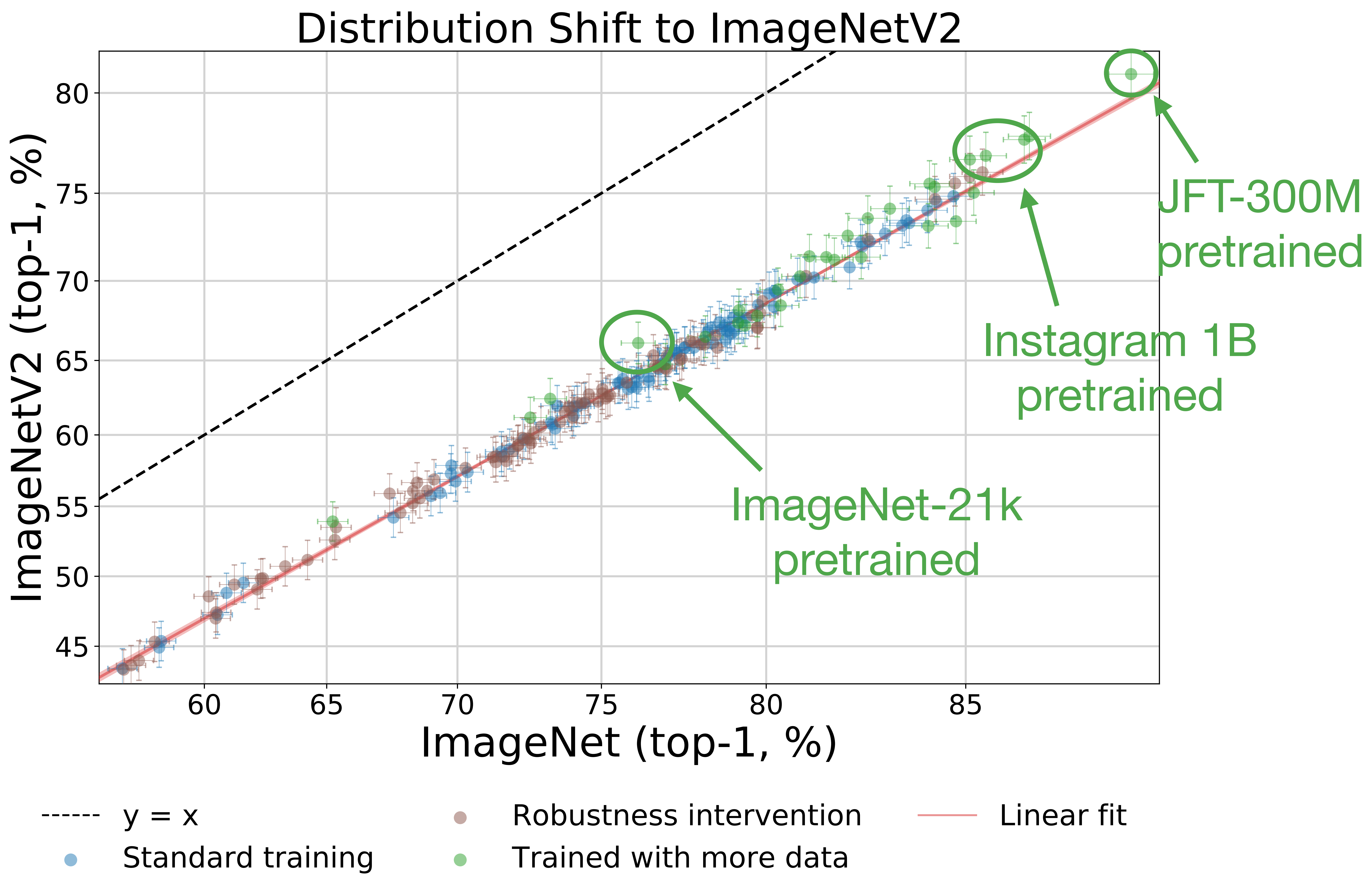
Similar story in domain generalization


In Search of Lost Domain Generalization

Ishaan Gulrajani and David Lopez-Paz*
Facebook AI Research
igul222@gmail.com, dlp@fb.com

Abstract

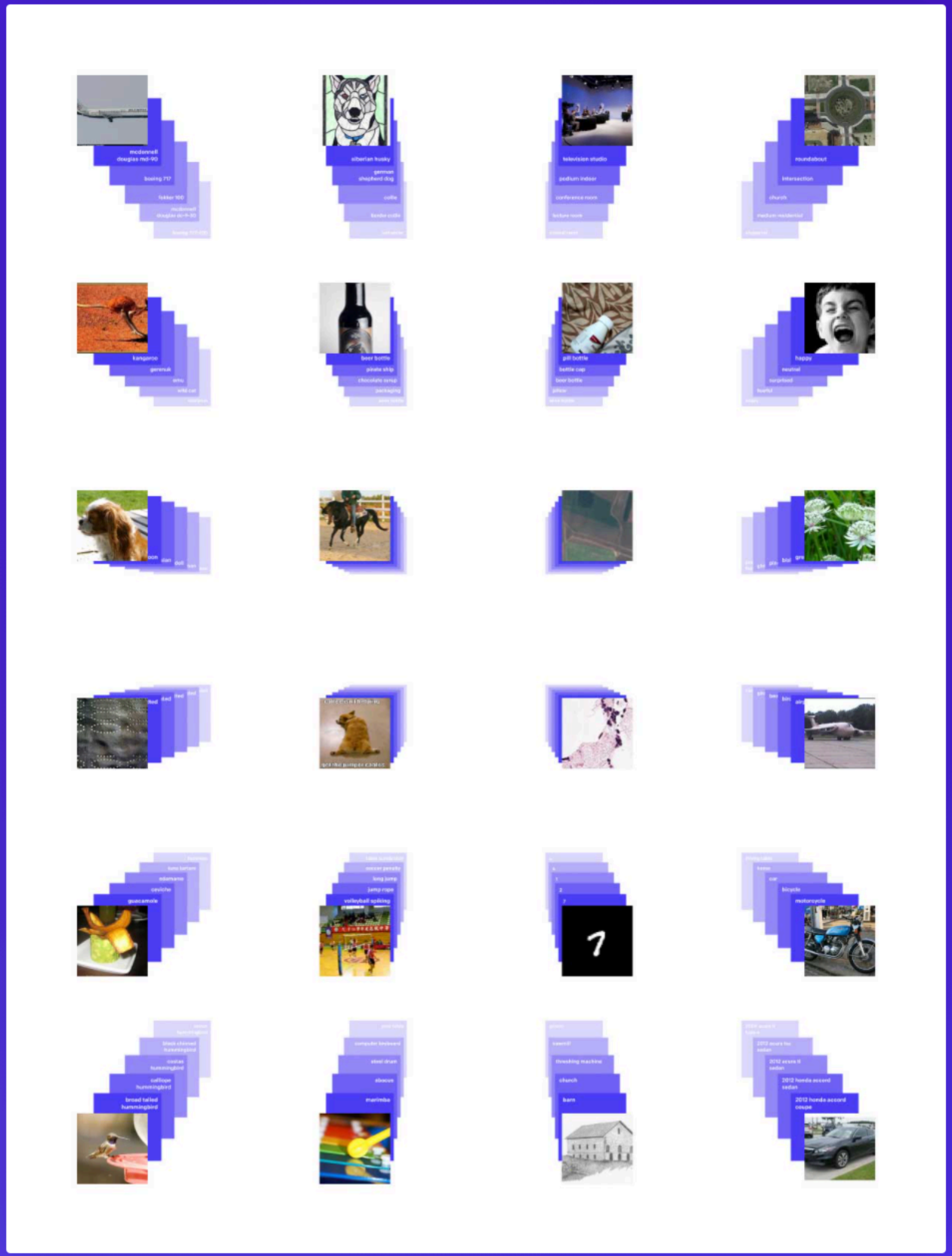
The goal of domain generalization algorithms is to predict well on distributions different from those seen during training. While a myriad of domain generalization algorithms exist, inconsistencies in experimental conditions—datasets, architectures, and model selection criteria—render fair and realistic comparisons difficult. In this paper, we are interested in understanding how useful domain generalization algorithms are in realistic settings. As a first step, we realize that model selection is non-trivial for domain generalization tasks. Contrary to prior work, we argue that domain generalization algorithms without a model selection strategy should be regarded as incomplete. Next, we implement DOMAINBED, a testbed for domain generalization including seven multi-domain datasets, nine baseline algorithms, and three model selection criteria. We conduct extensive experiments using DOMAINBED and find that, when carefully implemented, empirical risk minimization shows state-of-the-art performance across all datasets. Looking forward, we hope that the release of DOMAINBED, along with contributions from fellow researchers, will streamline reproducible and rigorous research in domain generalization.



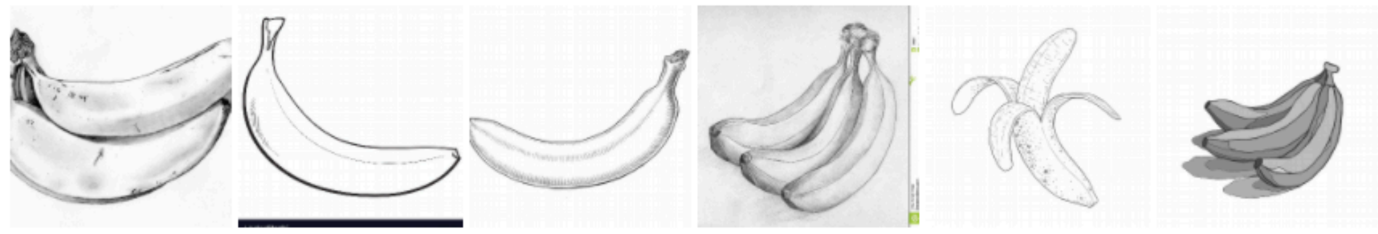
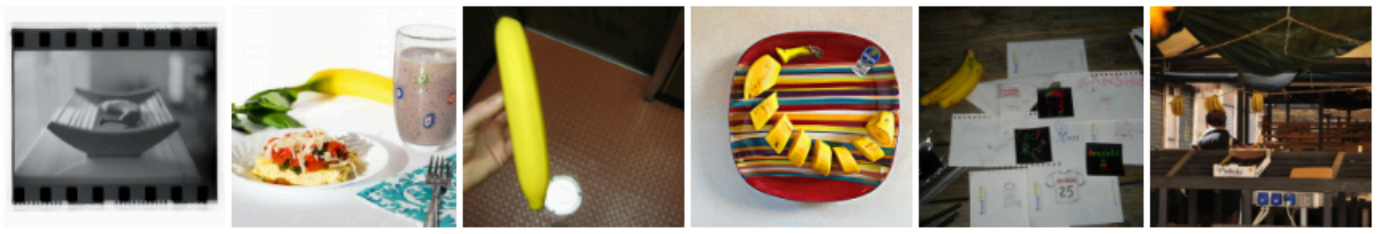
Training on (a lot) more data gives a **small** amount of effective robustness.

CLIP: Connecting Text and Images

We’re introducing a neural network called CLIP which efficiently learns visual concepts from natural language supervision. CLIP can be applied to any visual classification benchmark by simply providing the names of the visual categories to be recognized, similar to the “zero-shot” capabilities of GPT-2 and GPT-3.



January 5, 2021
15 minute read

DATASET	IMAGENET RESNET101	CLIP VIT-L	Effective robustness
 ImageNet	76.2%	76.2%	
 ImageNet V2	64.3%	70.1%	+6%
 ImageNet Rendition	37.7%	88.9%	+51%
 ObjectNet	32.6%	72.3%	+40%
 ImageNet Sketch	25.2%	60.2%	+35%
 ImageNet A	2.7%	77.1%	+74%

Very large improvements in out-of-distribution robustness.

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] **Jeffrey Wu** [*1] **Rewon Child** [1] **David Luan** [1] **Dario Amodei** [**1] **Ilya Sutsk**

The GPT-2 paper (2019)

1. Introduction

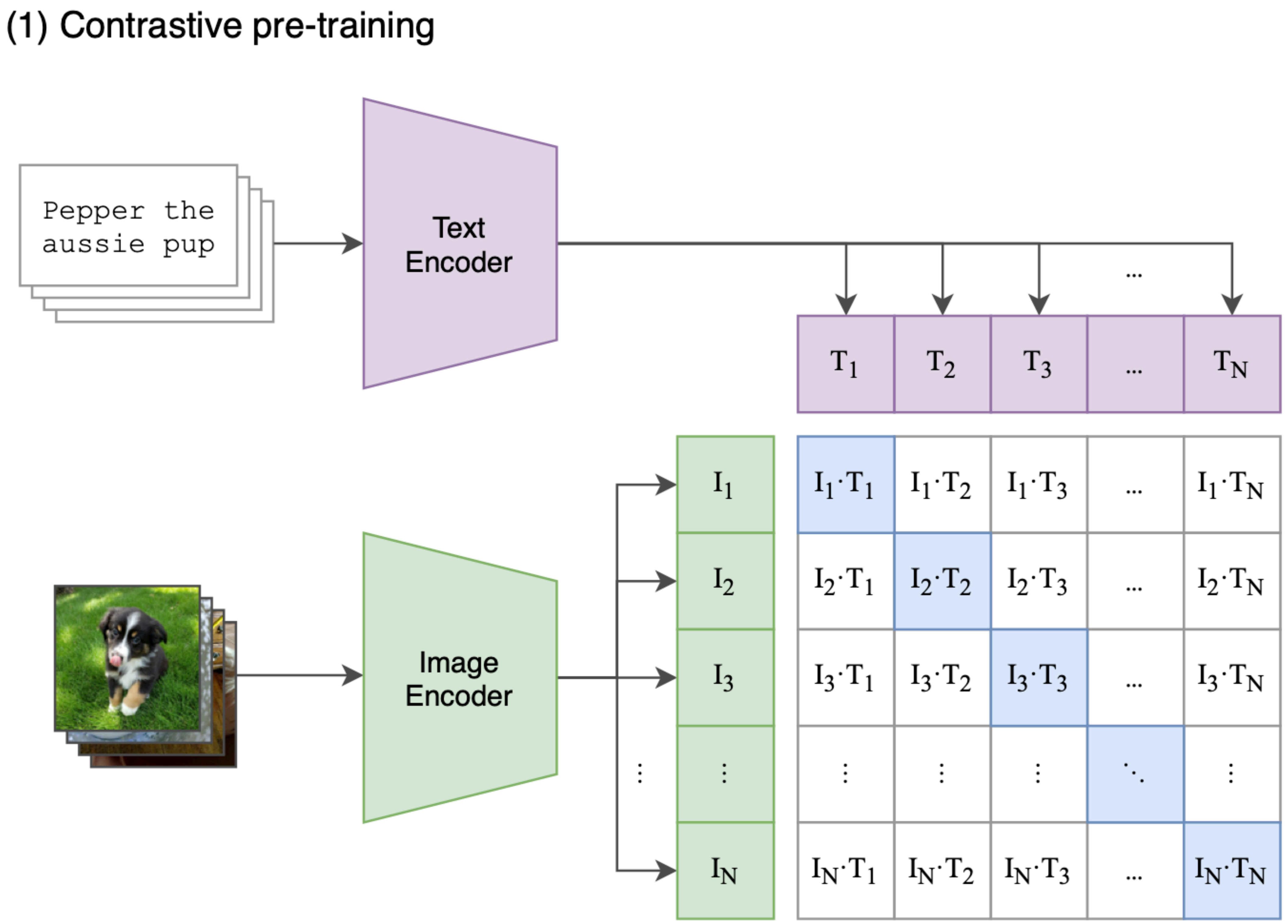
Machine learning systems now excel (in expectation) at tasks they are trained for by using a combination of large datasets, high-capacity models, and supervised learning (Krizhevsky et al., 2012) (Sutskever et al., 2014) (Amodei et al., 2016). Yet these systems are brittle and sensitive to slight changes in the data distribution (Recht et al., 2018) and task specification (Kirkpatrick et al., 2017). Current systems are better characterized as narrow experts rather than

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]

The CLIP paper (2021)

An overview of CLIP



Training data: 400 million images collected from the web (dataset internal to OpenAI).

Compute: Trained on 250 - 600 GPUs for up to 18 days.

Model: ResNets and ViTs with up to 300M parameters.

How is CLIP Trained

The western slope of Mount Rainier in 2005

$\longrightarrow u_1$

$\longrightarrow v_1$

$\dots$

An image of an avocado armchair

$\longrightarrow u_n$

$\longrightarrow v_n$

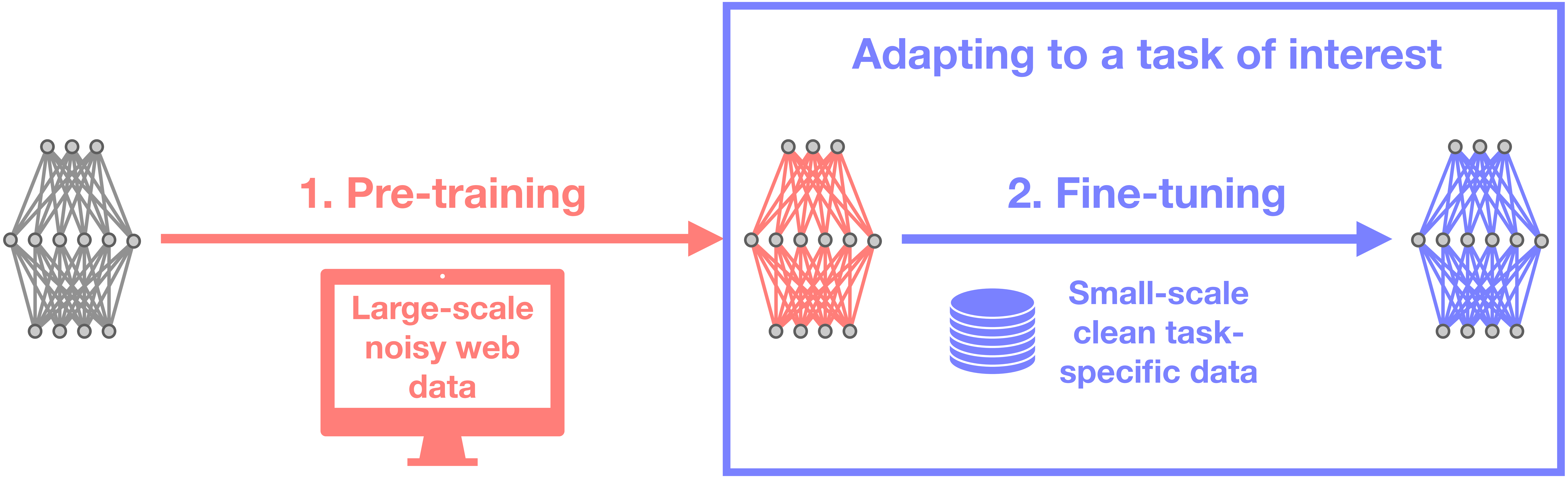
Objective: pairs should be more aligned

$$\sum_{i=1}^{n} -\log\left(\frac{e^{\langle u_i, v_i\rangle}}{\sum_{j=1}^{n} e^{\langle u_i, v_j\rangle}}\right)$$

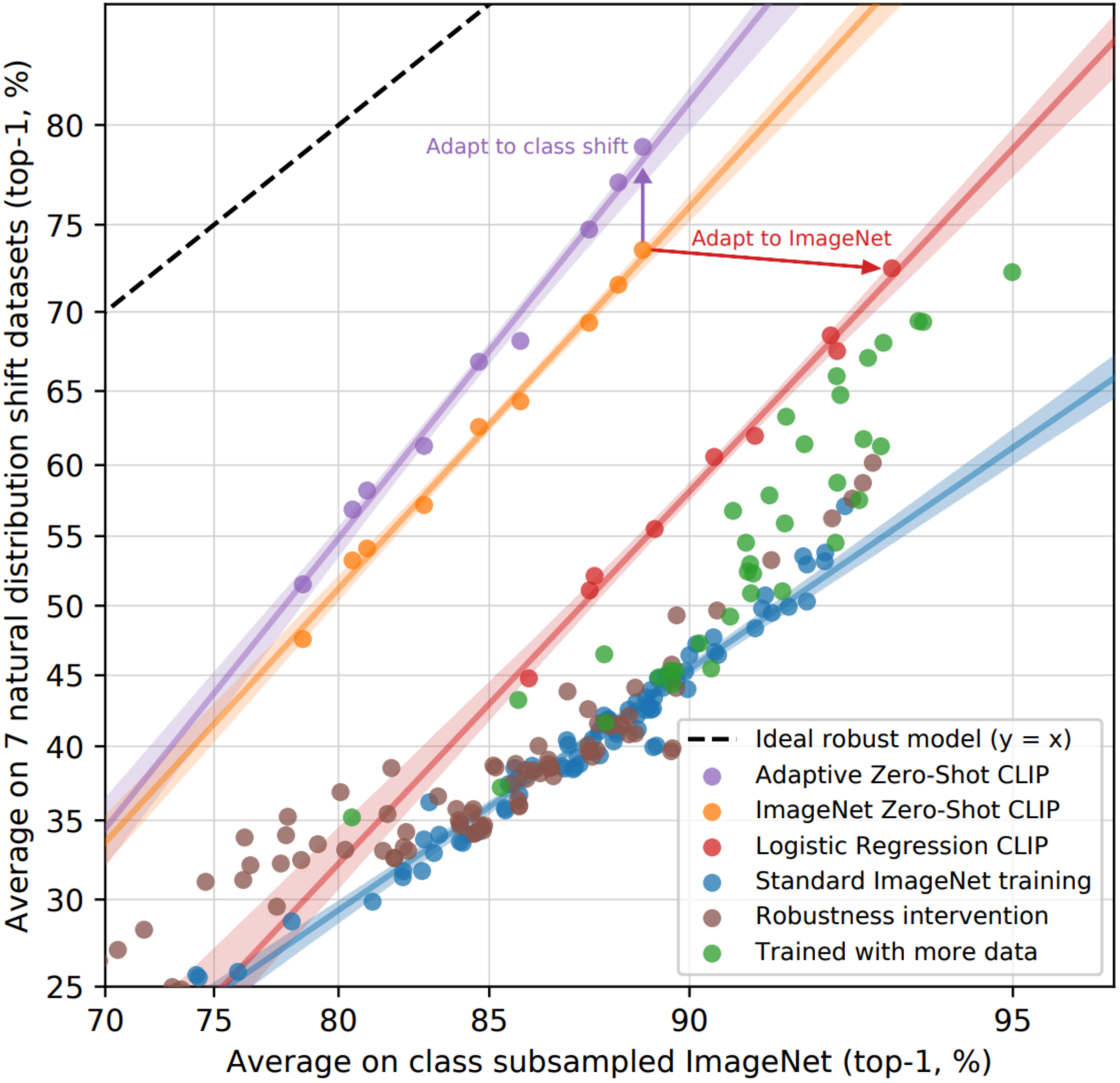
$$+\sum_{i=1}^{n} -\log\left(\frac{e^{\langle u_i, v_i\rangle}}{\sum_{j=1}^{n} e^{\langle u_j, v_i\rangle}}\right)$$

Fine-tuning vs. zero-shot inference

State-of-the-art ML models often come from a two-step process.



CLIP skips fine-tuning: directly applies to task of interest via zero-shot inference.



[Radford, Kim, Hallacy, Ramesh, Goh, Agarwal, Sastry, Askell, Mishkin, Clark, Krueger, Sutskever '21]

Large robustness gains

➡ What makes CLIP robust?

But: fine-tuning reduces robustness

➡ Can we get **both** high in-distribution **and** out-of-distribution accuracy?

Can we fine-tune CLIP without losing robustness?

Robust fine-tuning of zero-shot models

Mitchell Wortsman[*†] Gabriel Ilharco[*†] Jong Wook Kim[§] Mike Li[‡]

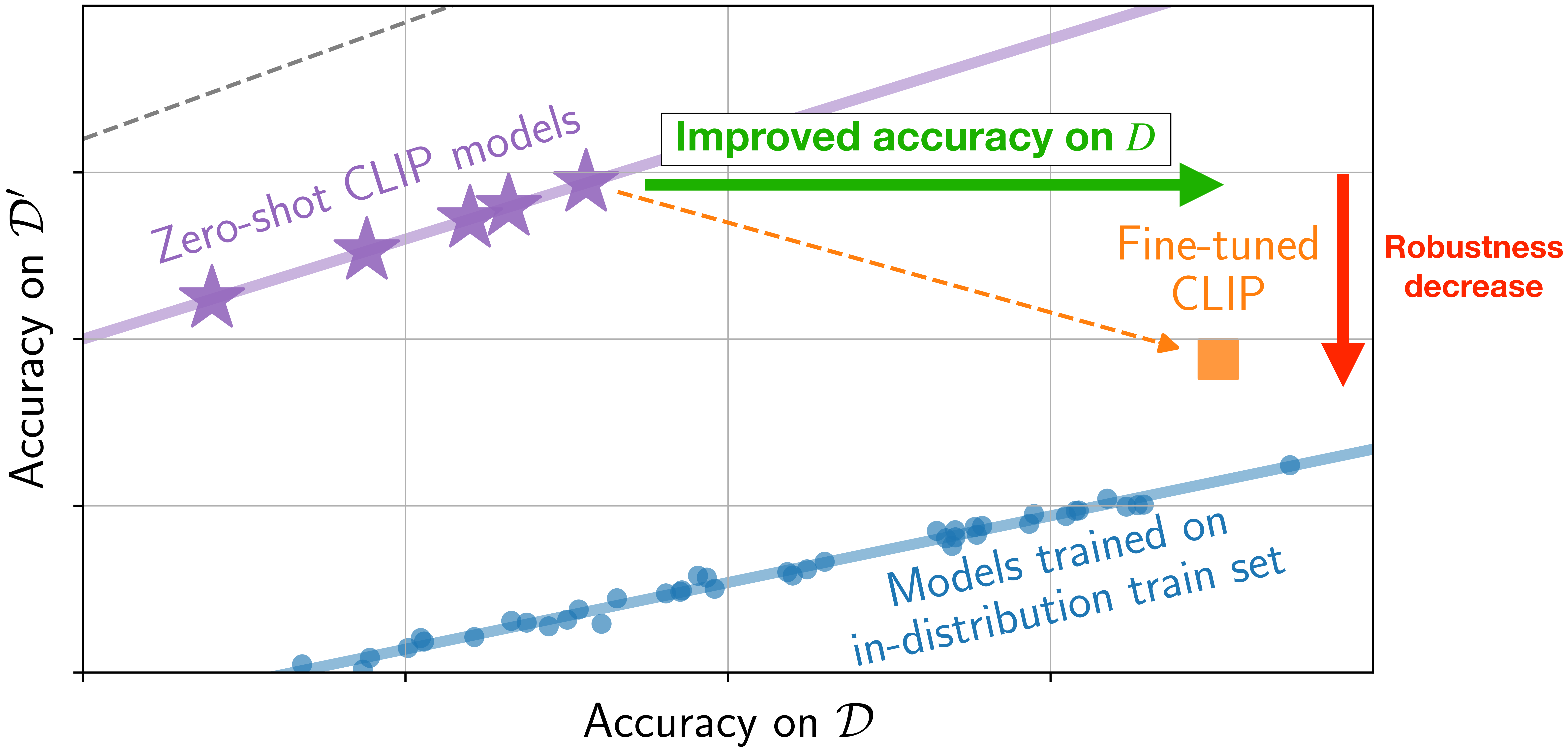
Simon Kornblith[◇] Rebecca Roelofs[◇] Raphael Gontijo-Lopes[◇]

Hannaneh Hajishirzi[†○] Ali Farhadi[*†] Hongseok Namkoong[*‡] Ludwig Schmidt[†△]

Abstract

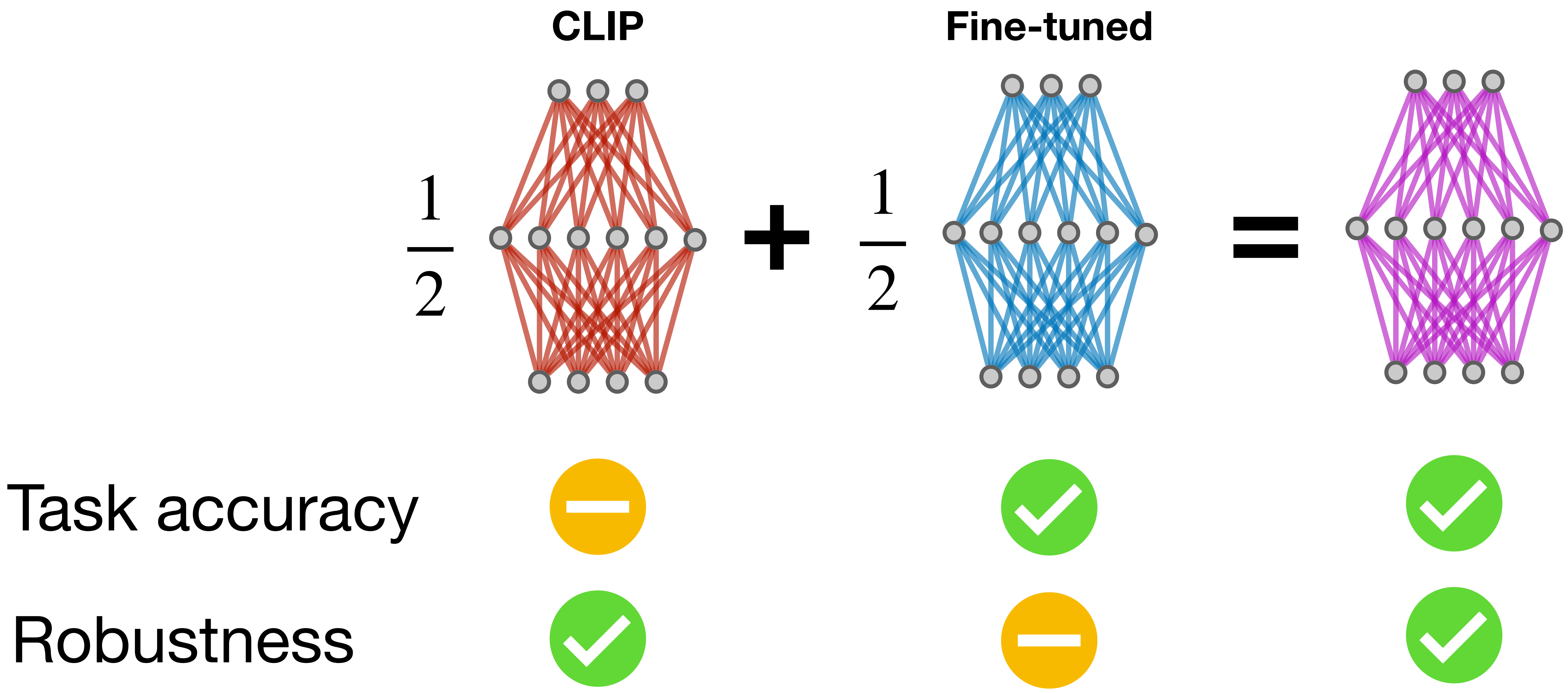
Large pre-trained models such as CLIP or ALIGN offer consistent accuracy across a range of data distributions when performing zero-shot inference (i.e., without fine-tuning on a specific dataset). Although existing fine-tuning methods substantially improve accuracy on a given target distribution, they often reduce robustness to distribution shifts. We address this tension by introducing a simple and effective method for improving robustness while fine-tuning: ensembling the weights of the zero-shot and fine-tuned models (WiSE-FT). Compared to standard fine-tuning, WiSE-FT provides large accuracy improvements under distribution shift, while preserving high accuracy on the target distribution. On ImageNet and five derived distribution shifts, WiSE-FT improves accuracy under distribution shift by 4 to 6 percentage points (pp) over prior work while increasing ImageNet accuracy by 1.6 pp. WiSE-FT achieves similarly large robustness gains (2 to 23 pp) on a diverse set of six further distribution shifts, and accuracy gains of 0.8 to 3.3 pp compared to standard fine-tuning on seven commonly used transfer learning datasets. These improvements come at no additional computational cost during fine-tuning or inference.

The problem with fine-tuning



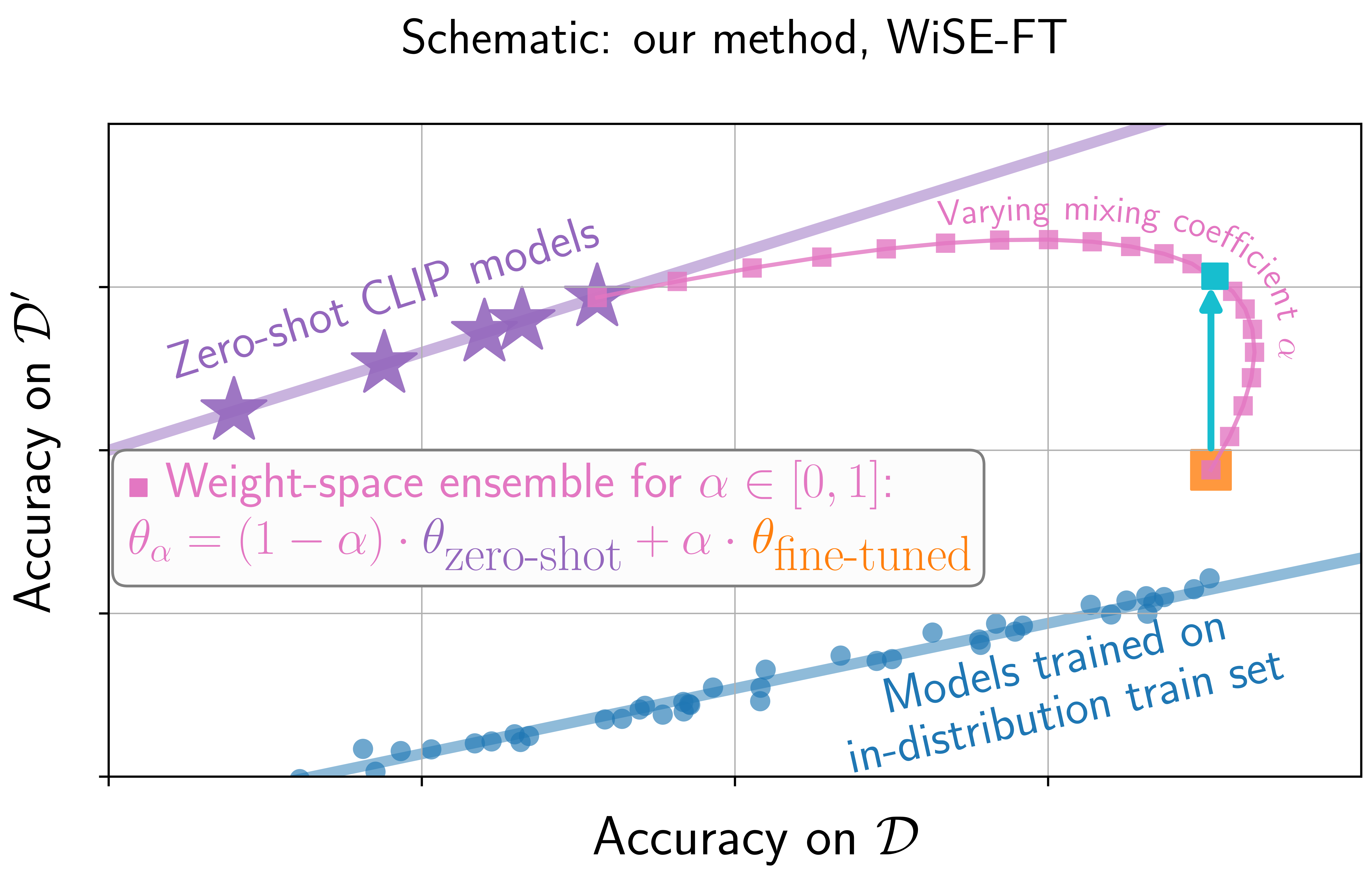
Raised as an **open problem** by researchers from OpenAI, Stanford, Google, etc.

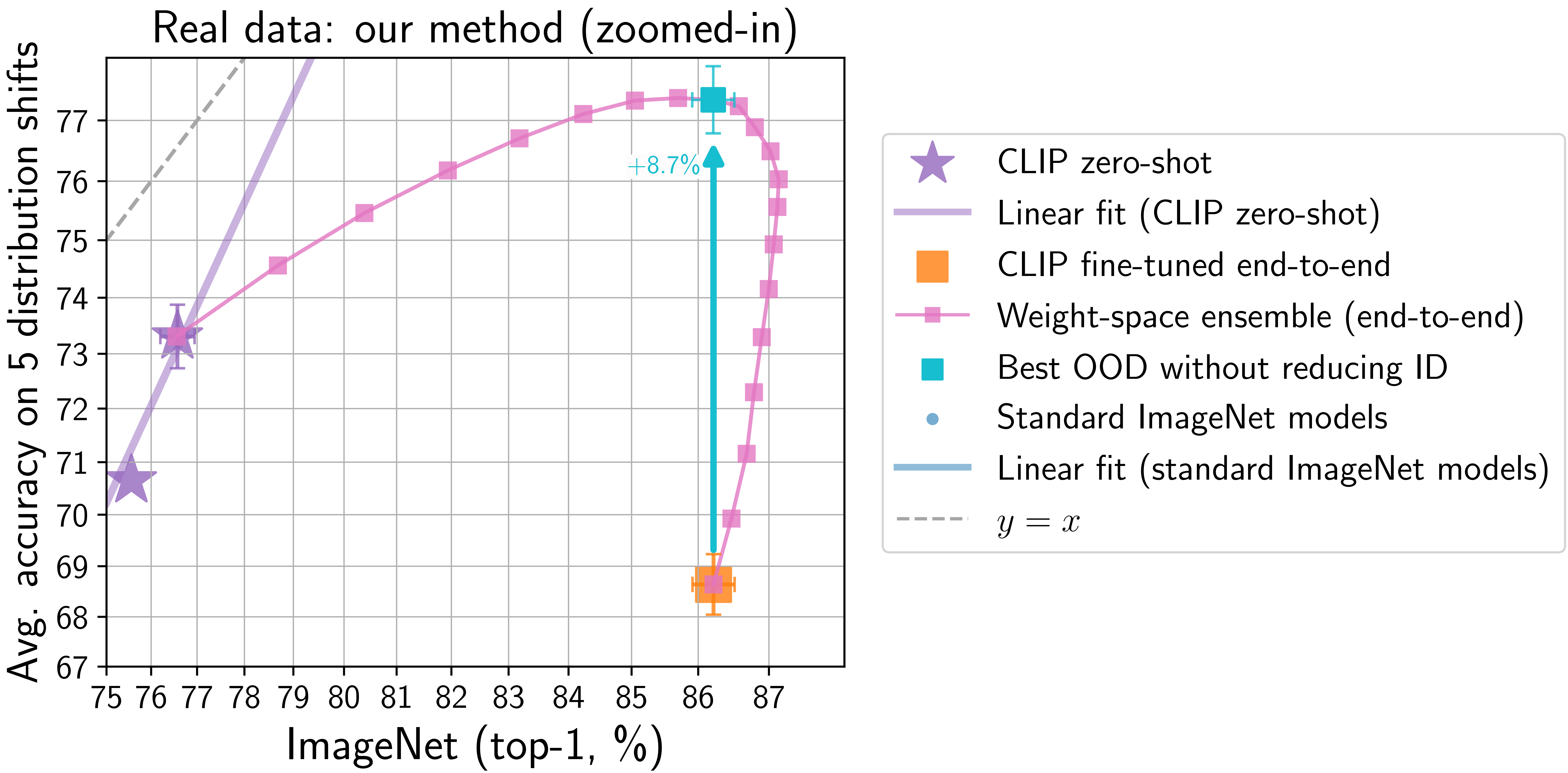
A simple but effective solution



Weight-space ensembles for fine-tuning (WiSE-FT)

Building on [Nagarajan, Kolter ’19], [Frankle, Dziugaite, Roy, Carbin ’20], [Neyshabur, Sedghi, Zhang ’20].

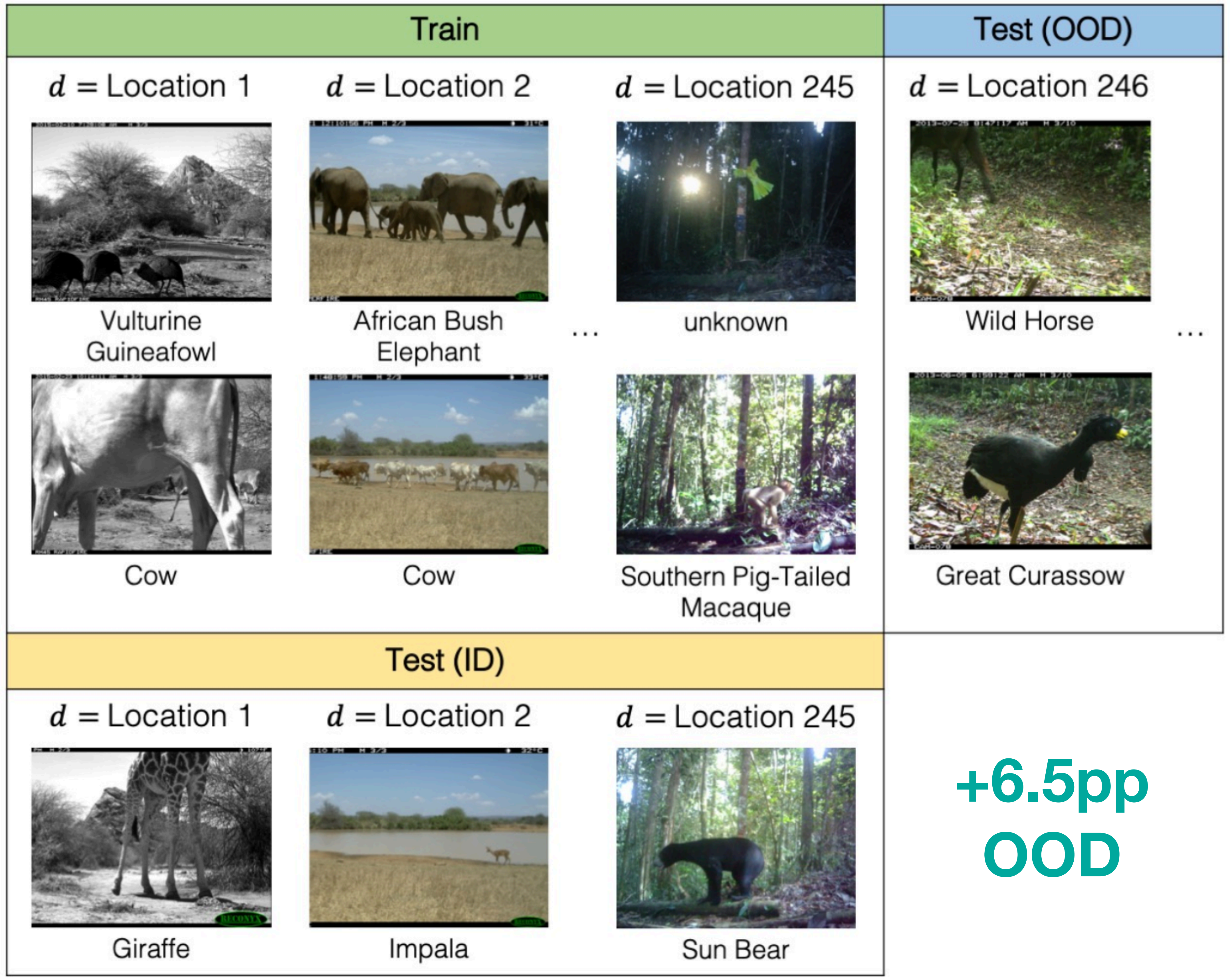

Schematic: our method, WiSE-FT
Accuracy on $\mathcal{D}'$
Accuracy on $\mathcal{D}$
Zero-shot CLIP models
Varying mixing coefficient α
Weight-space ensemble for $\alpha \in [0, 1]$:
$\theta_\alpha = (1 - \alpha) \cdot \theta_{\text{zero-shot}} + \alpha \cdot \theta_{\text{fine-tuned}}$
Models trained on
in-distribution train set


Real data: our method (zoomed-in)
Avg. accuracy on 5 distribution shifts
ImageNet (top-1, %)
+8.7%
67
68
69
70
71
72
73
74
75
76
77
75
76
77
78
79
80
81
82
83
84
85
86
87
CLIP zero-shot
Linear fit (CLIP zero-shot)
CLIP fine-tuned end-to-end
Weight-space ensemble (end-to-end)
Best OOD without reducing ID
Standard ImageNet models
Linear fit (standard ImageNet models)
$y = x$

WILDS

Koh et al., 2021

iWildCam



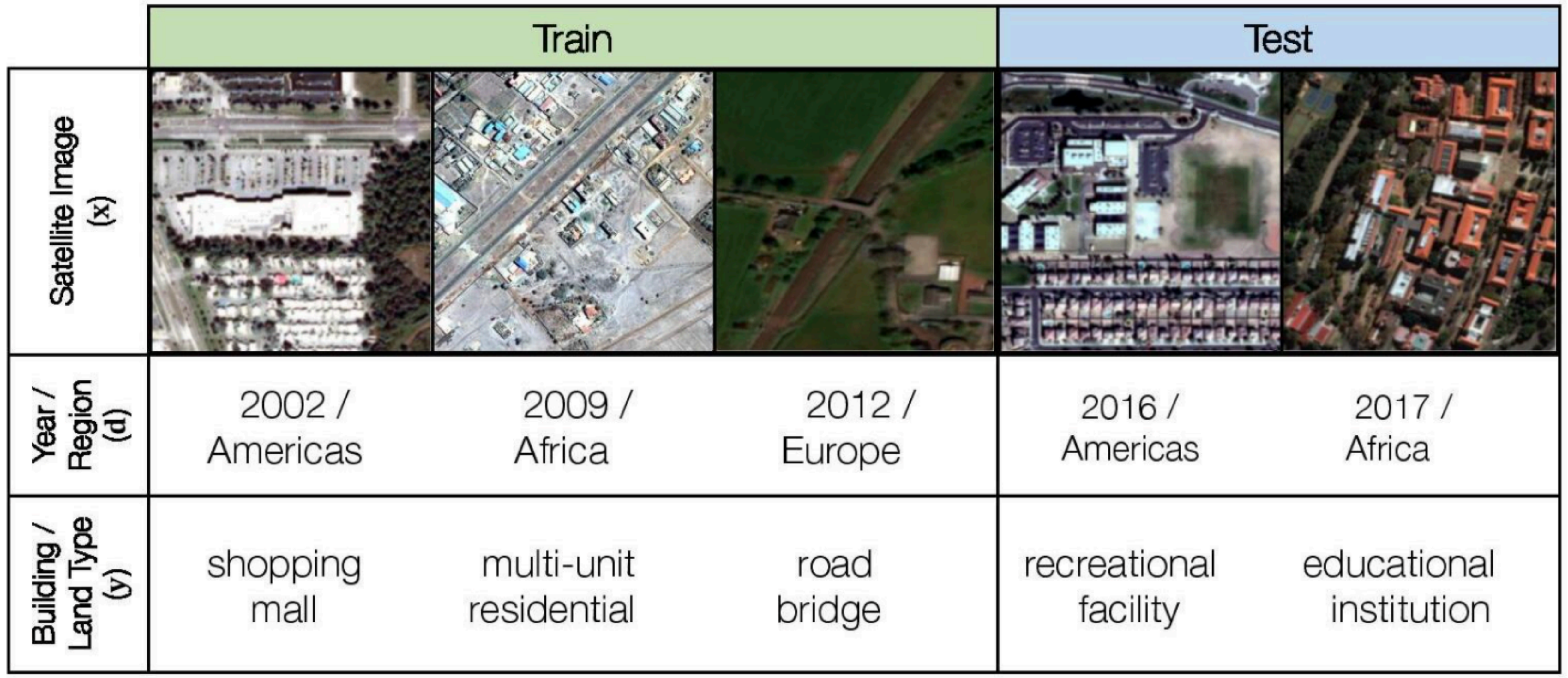
+6.5pp OOD

Beery et al., 2018

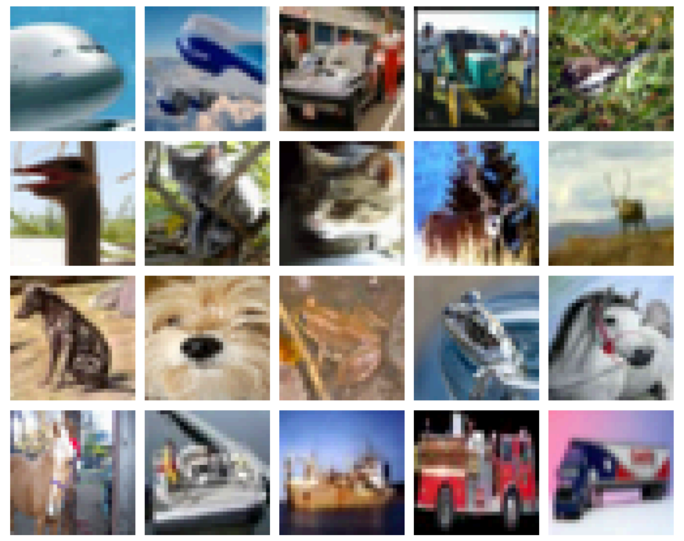
FMoW

+3.7pp OOD

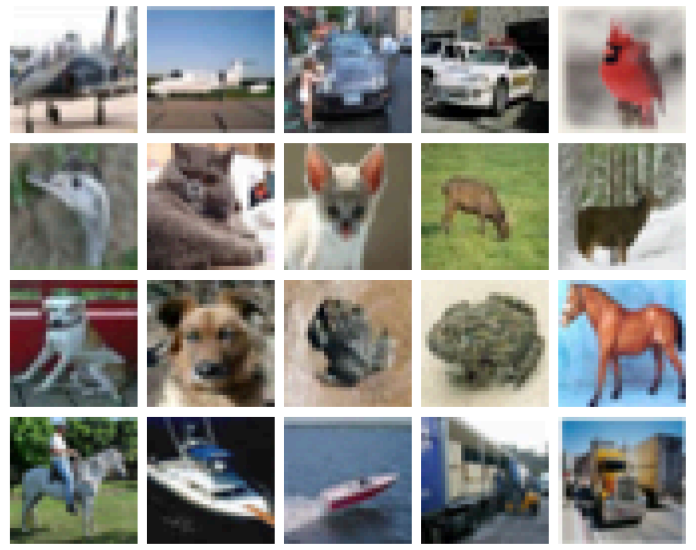
Christie et al., 2018



+2.2pp OOD



+3.0pp OOD



CIFAR-10.1.
Recht et al., 2019

CIFAR-10.2.
Lu et al., 2020

+8.3pp OOD

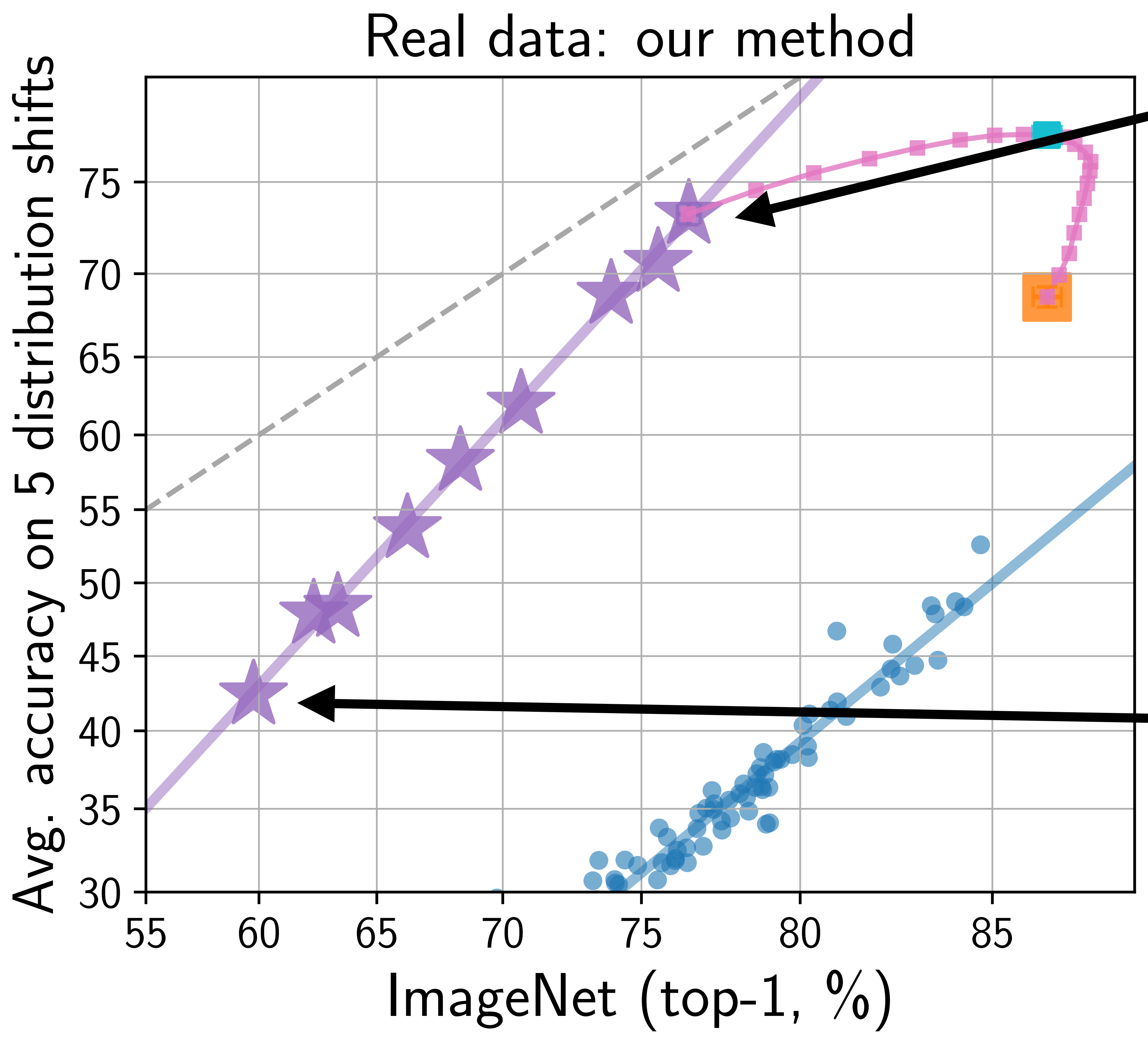
ImageNet-Vid-Robust

Shankar et al., 2019

YTBBRobust

+14.7pp OOD

Robustness gains invariant as compute scale increases



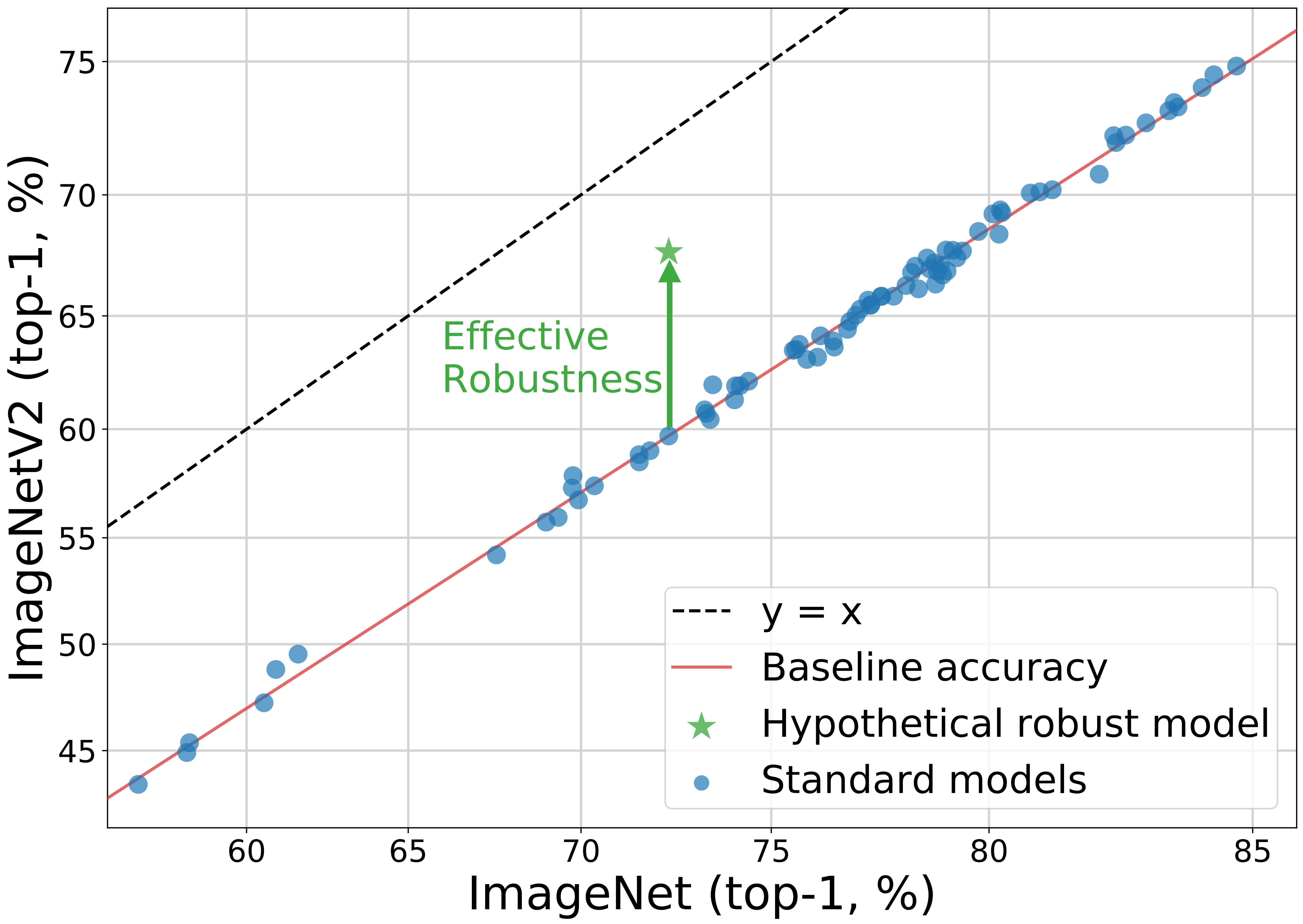
Final result (high accuracy models)

Reliable extrapolation via "Accuracy on the line"

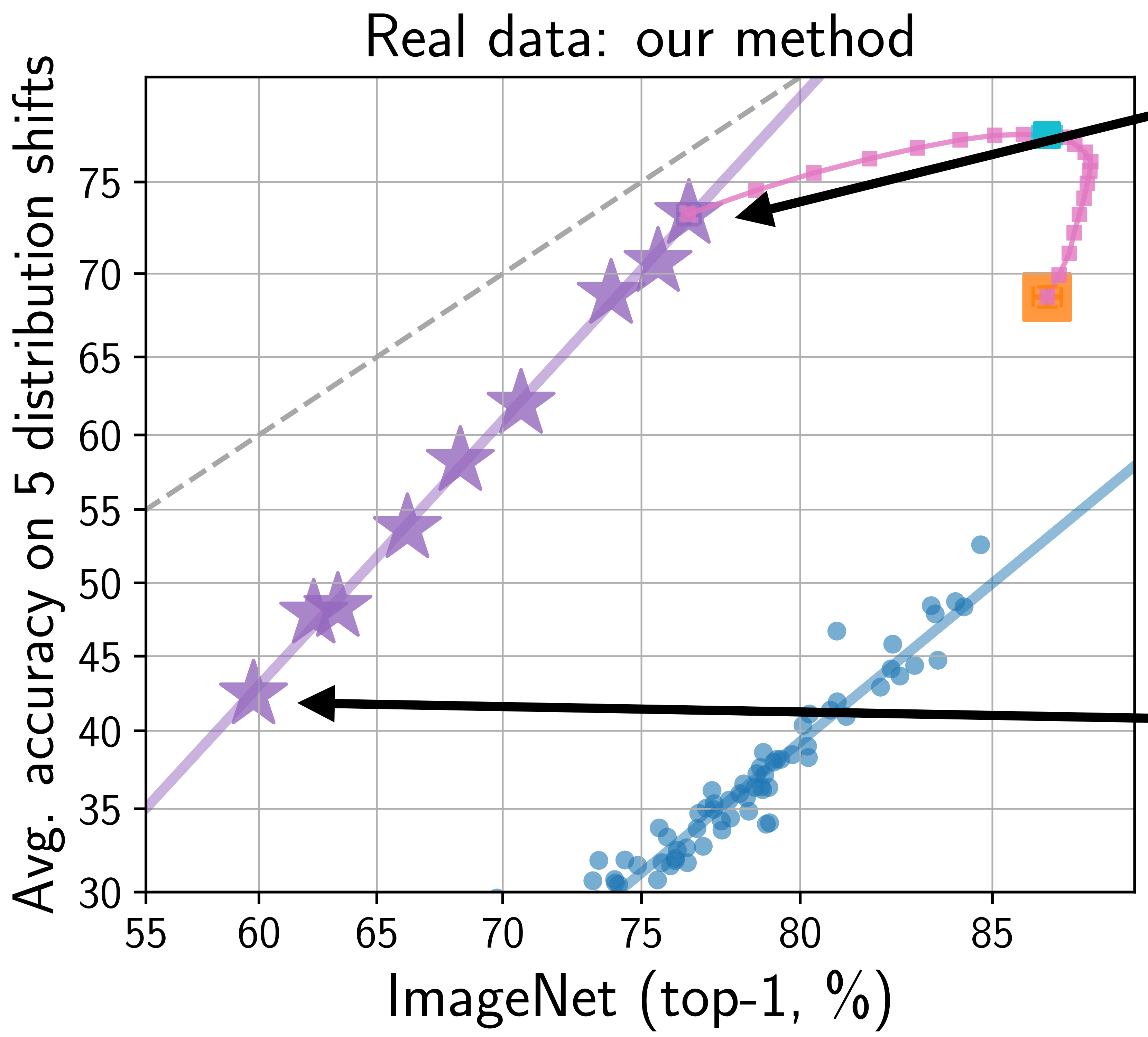
Where all the experiments happened (low accuracy models)

→ cheaper → faster iteration

All experiments measured effective robustness



Robustness gains invariant as compute scale increases



Final result (high accuracy models)

Reliable extrapolation via "Accuracy on the line"

Where all the experiments happened (low accuracy models)

→ cheaper → faster iteration

Experiment with the full-scale model at OpenAI worked on **first try**!

ID-OOD trends are a reliable scaling law for model design

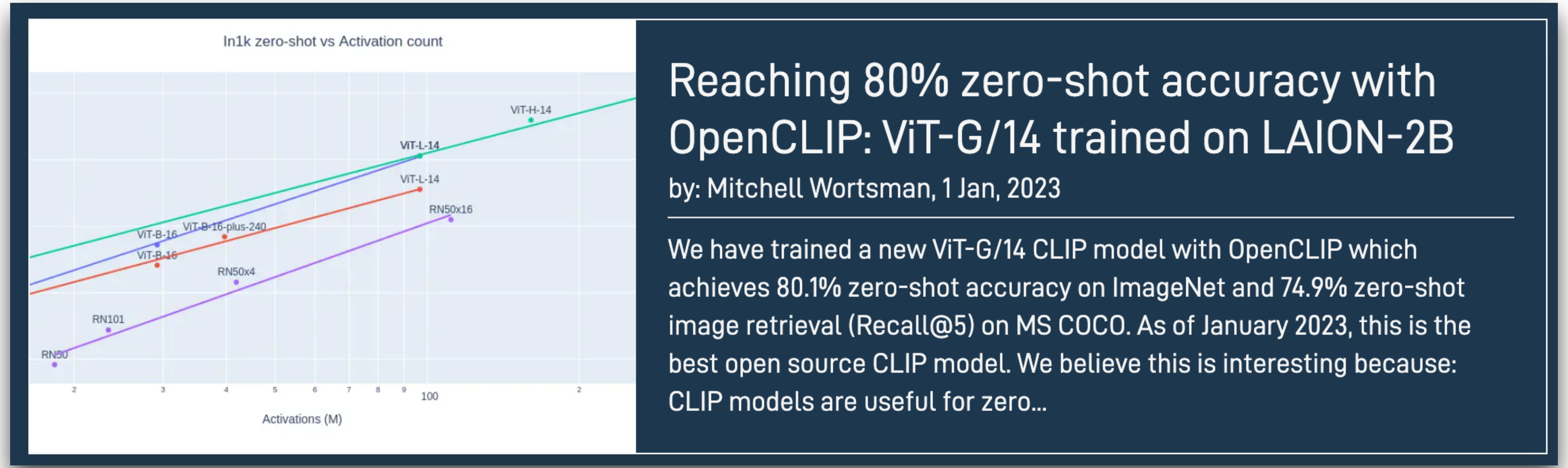
Why stop at averaging two models?

Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time

Mitchell Wortsman [1] **Gabriel Ilharco** [1] **Samir Yitzhak Gadre** [2] **Rebecca Roelofs** [3] **Raphael Gontijo-Lopes** [3]
Ari S. Morcos [4] **Hongseok Namkoong** [2] **Ali Farhadi** [1] **Yair Carmon** [*5] **Simon Kornblith** [*3] **Ludwig Schmidt** [*1]

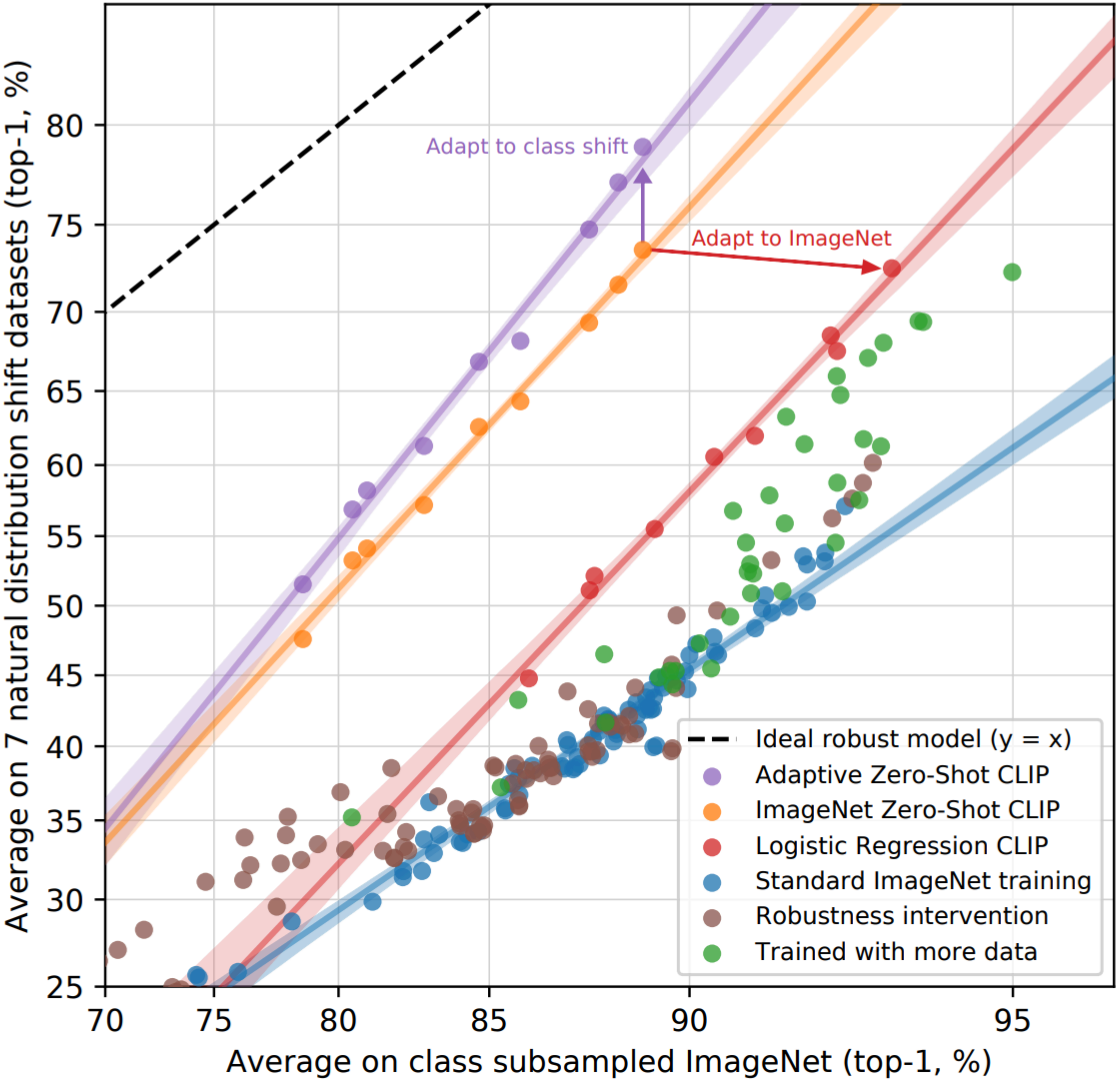
Outcome: State-of-the-art public CLIP models

To enable our research and make our models available, we built **OpenCLIP.**



- Most widely used open source CLIP implementation (4,900 GitHub stars)
- Best public CLIP models (as of 2023 better than OpenAI checkpoints)
- Template for proprietary implementations (Apple, Meta)
- More than 150,000 downloads (git clones) per day
- Top 1% of all Python packages
- OpenCLIP provides the language guidance component in Stable Diffusion

LAION



[Radford, Kim, Hallacy, Ramesh, Goh, Agarwal, Sastry, Askell, Mishkin, Clark, Krueger, Sutskever ’21]

Large robustness gains

→ What makes CLIP robust?

Data Determines Distributional Robustness in Contrastive Language Image Pre-training (CLIP)

Alex Fang[†] Gabriel Ilharco[†] Mitchell Wortsman[†] Yuhao Wan[†]

Vaishaal Shankar[◇] Achal Dave[◇] Ludwig Schmidt[†◦]

Abstract

Contrastively trained image-text models such as CLIP, ALIGN, and BASIC have demonstrated unprecedented robustness to multiple challenging natural distribution shifts. Since these image-text models differ from previous training approaches in several ways, an important question is what causes the large robustness gains. We answer this question via a systematic experimental investigation. Concretely, we study five different possible causes for the robustness gains: (i) the training set size, (ii) the training distribution, (iii) language supervision at training time, (iv) language supervision at test time, and (v) the contrastive loss function. Our experiments show that the more diverse training distribution is the main cause for the robustness gains, with the other factors contributing little to no robustness. Beyond our experimental results, we also introduce ImageNet-Captions, a version of ImageNet with original text annotations from Flickr, to enable further controlled experiments of language-image training.

1 Introduction

Large pre-trained language-image models such as CLIP [27], ALIGN [21], and BASIC [26] have recently demonstrated unprecedented robustness on a variety of natural distribution shifts. In contrast to prior models that are trained on images with class annotations, CLIP and relatives[1] are directly trained on images and their corresponding unstructured text from the web. The resulting models achieve large robustness even on challenging distribution shifts such as ImageNetV2 [28] and ObjectNet [2]. No prior algorithmic techniques had enhanced robustness on these datasets even after multiple years of intensive research in reliable machine learning [13, 35]. As CLIP also improves robustness on a wide range of other distribution shifts, an important question emerges: *What causes CLIP's unprecedented robustness?*

Hypotheses for CLIP's robustness

	CLIP	Standard ImageNet supervised learning
Language supervision	Yes	No
Training distribution	???	ImageNet
Training set size	400M	1.2M
Loss function	Contrastive	Supervised
Test-time prompting	Yes	No
Model architecture	ViTs	CNNs

Hypotheses for CLIP's robustness

	CLIP	Standard ImageNet supervised learning
~~**Language supervision**~~	~~Yes~~	~~No~~
Training distribution	???	ImageNet
~~**Training set size**~~	~~400M~~	~~1.2M~~
~~**Loss function**~~	~~Contrastive~~	~~Supervised~~
~~**Test-time prompting**~~	~~Yes~~	~~No~~
~~**Model architecture**~~	~~ViTs~~	~~CNNs~~

What is the path to reliable generalization?

ML = **algorithms** + **data**

- Optimization procedures
- Model architectures
- Loss functions
- … (thousands of papers)

?

DeepMind

28-04-2022

Flamingo: a Visual Language Model for Few-Shot Learning

Jean-Baptiste Alayrac[*,‡], Jeff Donahue[*], Pauline Luc[*], Antoine Miech[*], Iain Barr[†], Yana Hasson[†], Karel Lenc[†], Arthur Mensch[†], Katie Millican[†], Malcolm Reynolds[†], Roman Ring[†], Eliza Rutherford[†], Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, Karen Simonyan[*,‡]

[*]Equal contributions, ordered alphabetically, [†]Equal contributions, ordered alphabetically, [‡]Equal senior contributions

Building models that can be rapidly adapted to numerous tasks using only a handful of annotated examples is an open challenge for multimodal machine learning research. We introduce Flamingo, a family of Visual Language Models (VLM) with this ability. Flamingo models include key architectural innovations to: (i) bridge powerful pretrained vision-only and language-only models, (ii) handle sequences of arbitrarily interleaved visual and textual data, and (iii) seamlessly ingest images or videos as inputs. Thanks to their flexibility, Flamingo models can be trained on large-scale multimodal web corpora containing arbitrarily interleaved text and images, which is key to endow them with in-context few-shot learning capabilities. We perform a thorough evaluation of the proposed Flamingo models, exploring and measuring their ability to rapidly adapt to a variety of image and video understanding benchmarks. These include open-ended tasks such as visual question-answering, where the model is prompted with a question which it has to answer, captioning tasks, which evaluate the ability to describe a scene or an event, and close-ended tasks such as multiple choice visual question-answering. For tasks lying anywhere on this spectrum, we demonstrate that a *single* Flamingo model can achieve a new state of the art for few-shot learning, simply by prompting the model with task-specific examples. On many of these benchmarks, *Flamingo* actually surpasses the performance of models that are fine-tuned on thousands of times more task-specific data.

Main innovation in GPT-3: "In-context learning"

The three settings we explore for in-context learning

Zero-shot

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

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

One-shot

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

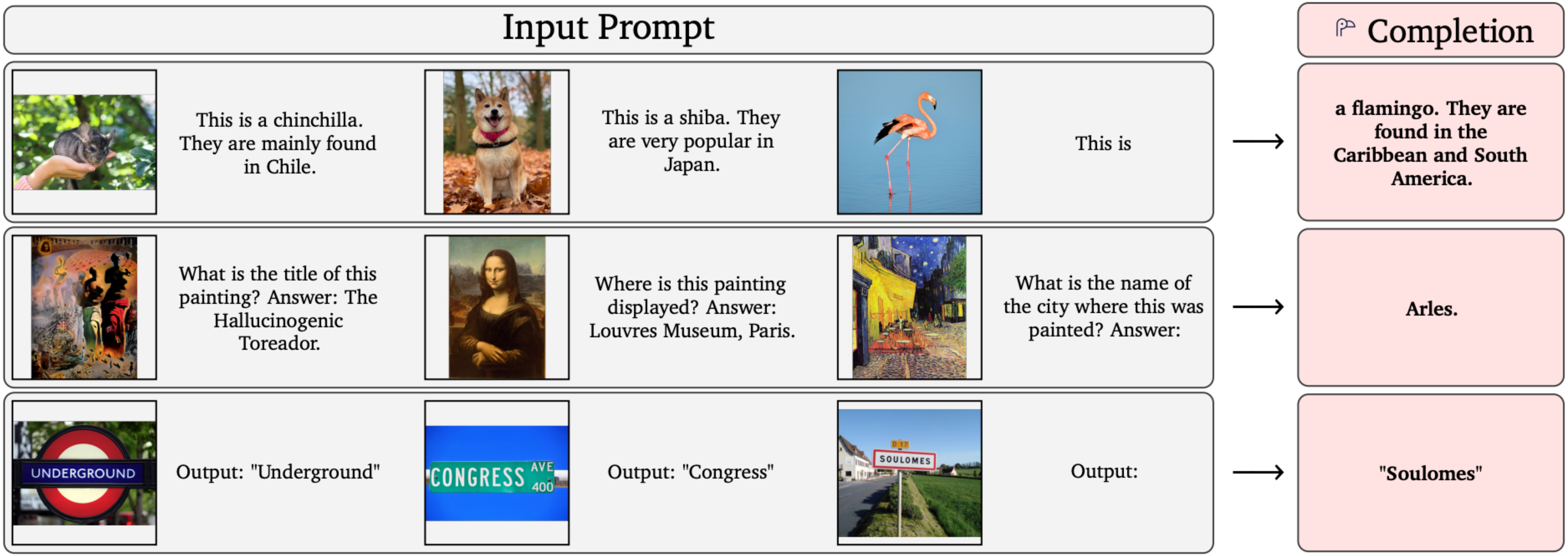
```
Translate English to French:     ← task description
sea otter => loutre de mer       ← example
cheese =>                        ← prompt
```

Few-shot

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

```
Translate English to French:     ← task description
sea otter => loutre de mer       ← examples
peppermint => menthe poivrée     ←
plush girafe => girafe peluche   ←
cheese =>                        ← prompt
```

How can we achieve in-context learning for multimodal tasks?


Input Prompt
Completion
This is a chinchilla. They are mainly found in Chile.
This is a shiba. They are very popular in Japan.
This is
a flamingo. They are found in the Caribbean and South America.
What is the title of this painting? Answer: The Hallucinogenic Toreador.
Where is this painting displayed? Answer: Louvres Museum, Paris.
What is the name of the city where this was painted? Answer:
Arles.
UNDERGROUND
Output: "Underground"
CONGRESS AVE 400
Output: "Congress"
D 17
SOULOMES
Output:
"Soulomes"

2 + 1
2+1=3
5 + 6
5+6=11
3 x 6
3x6=18
Output: A propaganda poster depicting a cat dressed as French emperor Napoleon holding a piece of cheese.
Output: A pink room with a flamingo pool float.
Output:
A portrait of Salvador Dali with a robot head.
Les sanglots longs des violons de l'automne blessent mon coeur d'une langueur monotone.
Pour qui sont ces serpents qui sifflent sur vos têtes?
Je suis un cœur qui bat pour vous.


pandas: 3
dogs: 2
giraffes: 4
I like reading
, my favourite play is Hamlet. I also like
, my favorite book is
Dreams from my Father.
What happens to the man after hitting the ball? Answer:
he falls down.

This is a picture of two teddy bears on the moon.

What are they doing?

They are having a conversation.

What object are they using?

It looks like a computer.

Is this surprising?

Yes, it is surprising.

Why is this picture surprising to you?

I think it is surprising because teddy bears are not usually found on the moon.

What is the common thing about these three images?

They are all flamingos.

What is the difference between these three images?

The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.

This is an apple with a sticker on it.

What does the sticker say?

The sticker says "iPod".

Where is the photo taken?

It looks like it's taken in a backyard.

Do you think it is printed or handwritten?

It looks like it's handwritten.

What color is the sticker?

It's white.

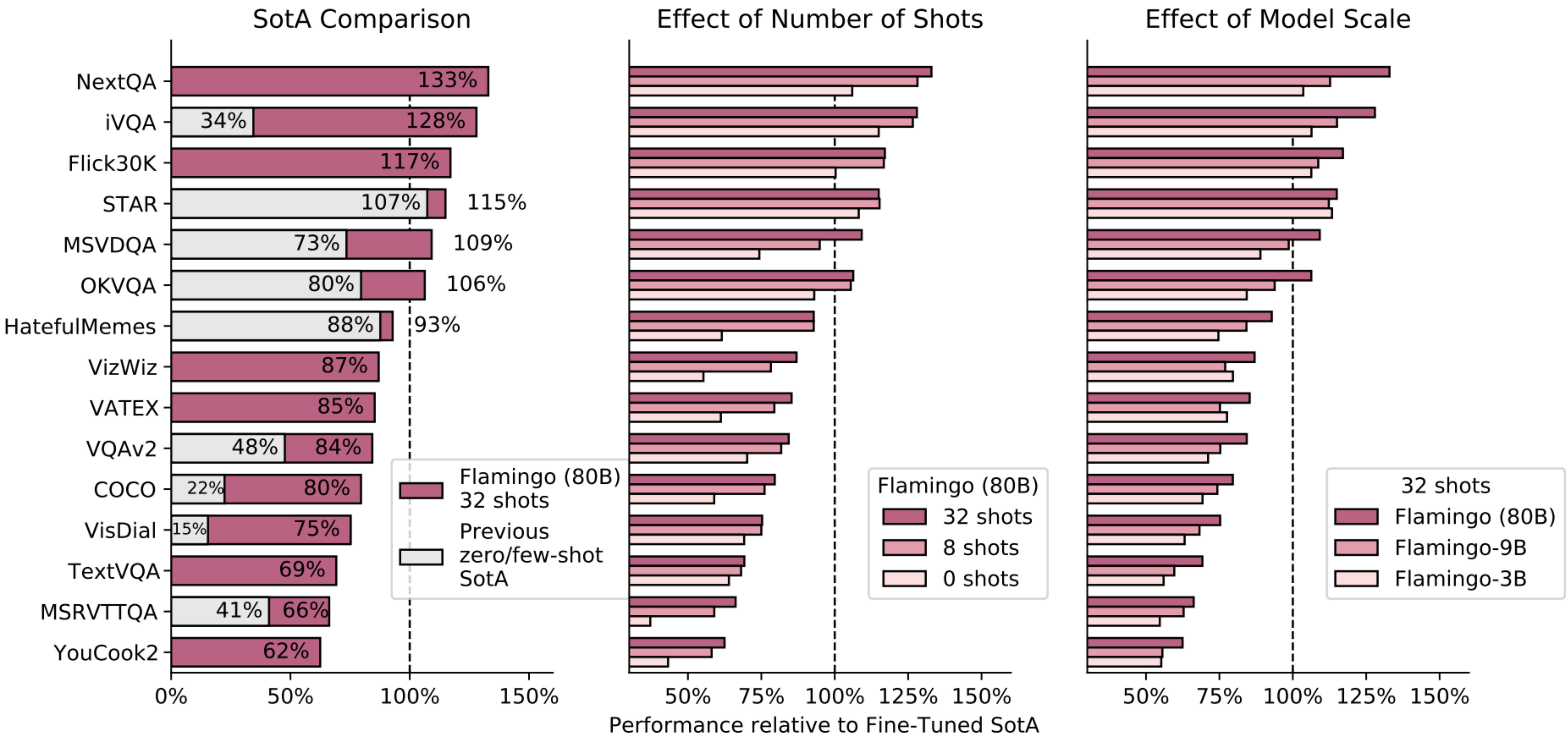


Figure 2 | **Overview of the results of the Flamingo models.** *Left*: Our largest model, dubbed *Flamingo*, outperforms state-of-the-art fine-tuned models on six out of the 16 tasks we consider despite not using any fine-tuning at all. For all 16 tasks where published few-shot results are available, *Flamingo* outperforms them by a large margin and sets the new few-shot state of the art. *Center*: *Flamingo* performance improves with the number of shots. *Right*: The performance of the Flamingo models increases with the model scale. *Note:* We omit RareAct, our 16th benchmark, as it is a zero-shot benchmark with no available fine-tuning results.

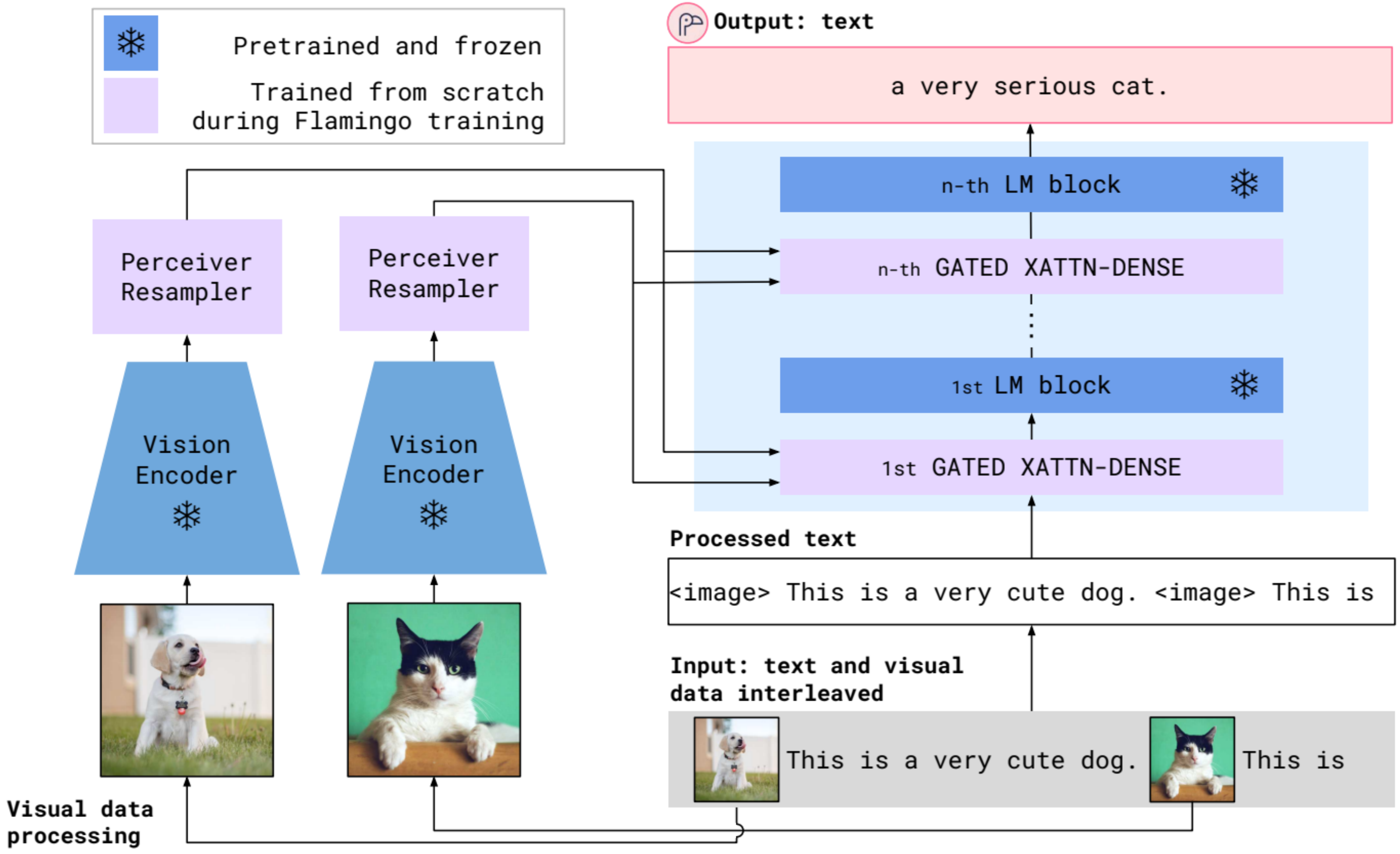
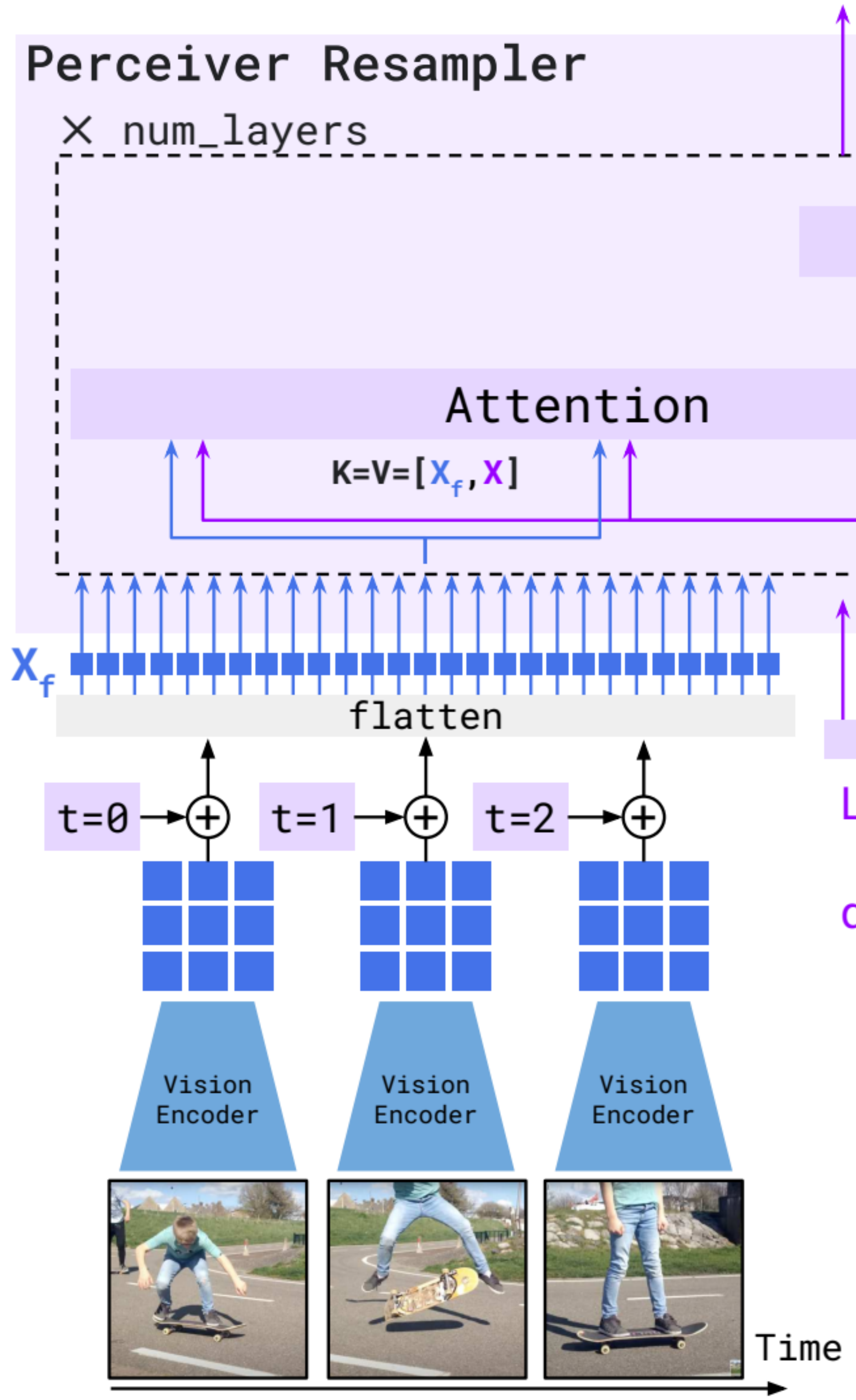


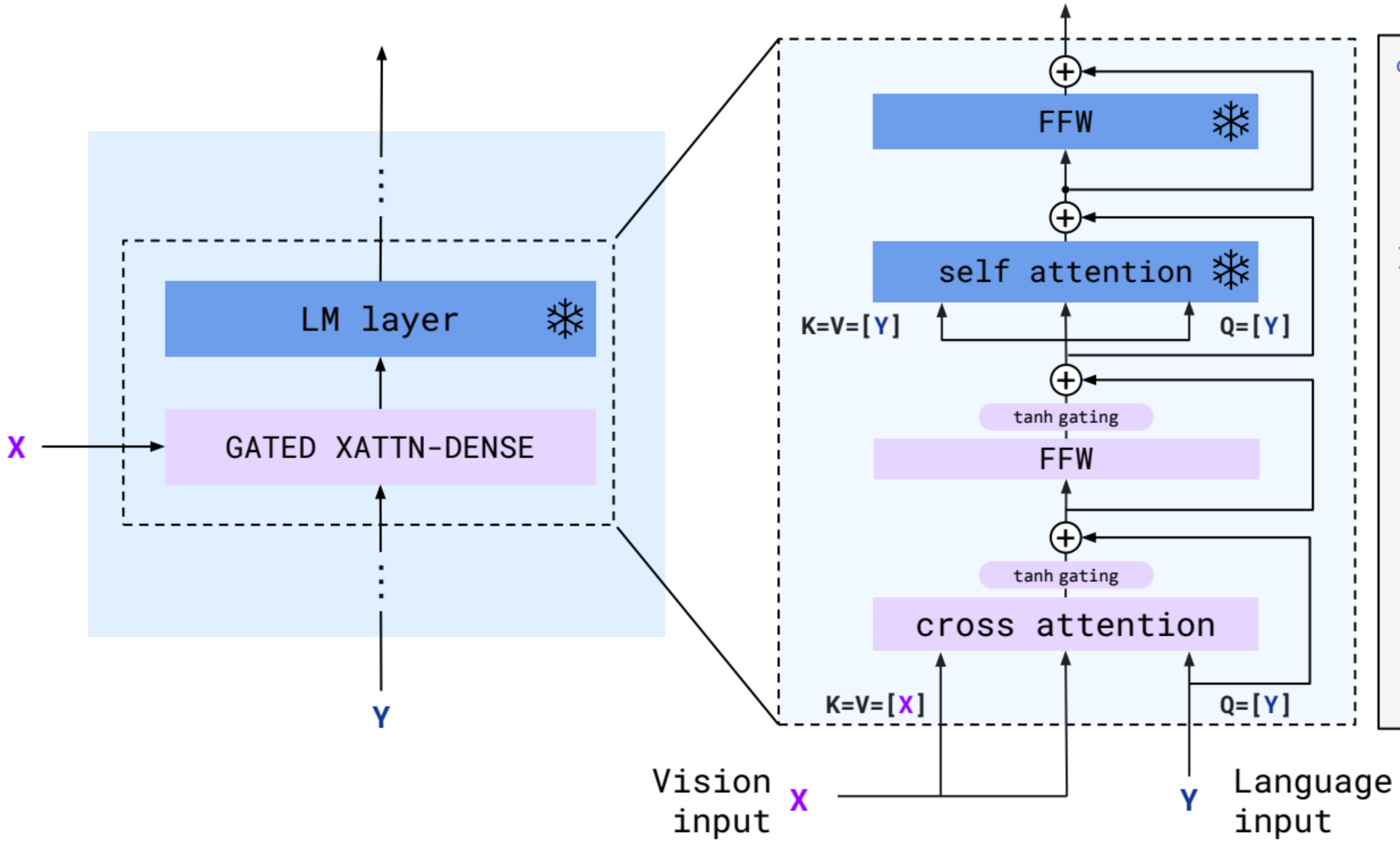
Figure 3 | **Overview of the Flamingo model.** The Flamingo models are a family of visual language model (VLM) that can take as input visual data interleaved with text and can produce free-form text as output. Key to its performance are novel architectural components and pretraining strategies described in Section 3.




```python
def perceiver_resampler(
    x_f,  # The [T, S, d] visual features (T=time, S=space)
    time_embeddings,  # The [T, 1, d] time pos embeddings.
    x,  # R learned latents of shape [R, d]
    num_layers,  # Number of layers
):
  """The Perceiver Resampler model."""

  # Add the time position embeddings and flatten.
  x_f = x_f + time_embeddings
  x_f = flatten(x_f)  # [T, S, d] -> [T * S, d]
  # Apply the Perceiver Resampler layers.
  for i in range(num_layers):
    # Attention.
    x = x + attention_i(q=x, kv=concat([x_f, x]))
    # Feed forward.
    x = x + ffw_i(x)
  return x
```

Figure 4 | **The Perceiver Resampler** module maps a *variable size* grid of spatio-temporal visual features coming out of the Vision Encoder to a *fixed* number of output tokens (five in the figure), independently of the input image resolution or the number of input video frames. This transformer has a set of learned latent vectors as queries, and the keys and values are a concatenation of the spatio-temporal visual features with the learned latent vectors. More details can be found in Section 3.1.1.




```python
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

Figure 5 | **GATED XATTN-DENSE layers.** We insert new cross-attention layers, whose keys and values are obtained from the vision features while using language queries, followed by dense feed forward layers in between existing pretrained and frozen LM layers in order to condition the LM on visual inputs. These layers are *gated* so that the LM is kept intact at initialization for improved stability and performance.

	Requires model sharding	Frozen		Trainable		Total count
		Language	Vision	GATED XATTN-DENSE	Resampler	
Flamingo-3B	✗	1.4B	435M	1.2B (every)	194M	**3.2B**
Flamingo-9B	✗	7.1B	435M	1.6B (every 4th)	194M	**9.3B**
Flamingo	✓	70B	435M	10B (every 7th)	194M	**80B**

Table 1 | **Parameter counts for Flamingo models.** We focus on increasing the parameter count of the frozen LM and the trainable vision-text GATED XATTN-DENSE modules while maintaining the frozen vision encoder and trainable Resampler to a fixed and small size across the different models. The frequency of the GATED XATTN-DENSE with respect to the original language model blocks is given in parenthesis.

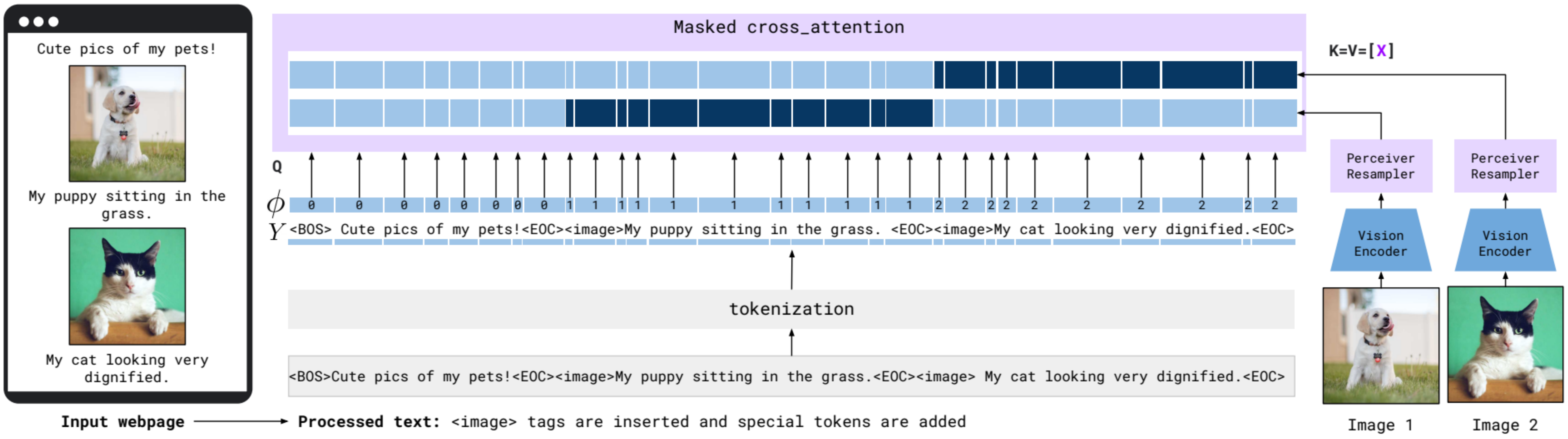


Figure 6 | **Interleaved visual data and text support.** Given text interleaved with images/videos, e.g. coming from a webpage, we first process the text by inserting `<image>` tags at the location of the visual data in the text as well as special tokens (`<BOS>` for "begining of sentence" or `<EOC>` for "end of chunk"). The images are processed independently by the Vision Encoder and Perceiver Resampler to extract visual tokens. Following our modeling choice motivated in Section 3.1.3, each text token only cross-attends to the visual tokens corresponding to the last preceding image. The function ϕ illustrated above indicates for each token what is the index of the last preceding image (and 0 if there are no preceding images). In practice, this selective cross-attention is achieved via a masked cross attention mechanism – illustrated here with the dark blue entries (non masked) and light blue entries (masked).

Figure 7 | **Training datasets.** Mixture of training datasets of different nature. N corresponds to the number of visual inputs for a single example. For paired image (or video) and text datasets, $N = 1$. T is the number of video frames with $T = 1$ being the special case of images. H, W, C are height, width and color channels.

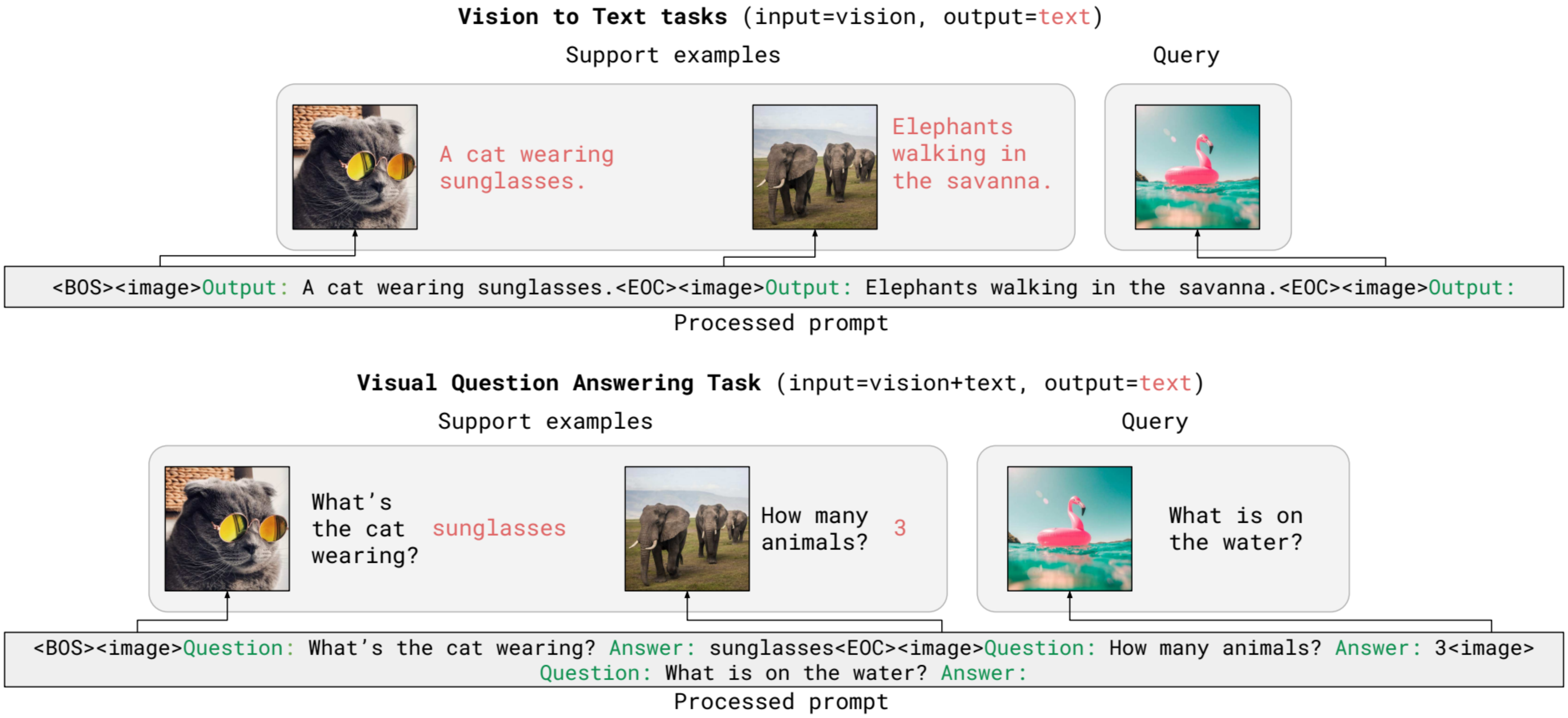


Figure 8 | **Few-shot interleaved prompt generation.** Given some task-specific few-shot examples (a.k.a. support examples) and a query for which Flamingo models have to make a prediction, we build the prompt by interleaving the image before each corresponding text. We introduce some formatting to do this, e.g. we prepend `"Output:"` to the expected response for all vision to text tasks or use a formatting prompt `"Question: {question} Answer: {answer}"` for visual question answering tasks.

Method	FT	Shot	OKVQA	VQAv2	COCO	MSVDQA	VATEX	VizWiz	Flick30K	MSRVTTQA	iVQA	YouCook2	STAR	VisDial	TextVQA	NextQA	HatefulMemes	RareAct
Zero/Few shot SOTA	✗	(X)	[39] 43.3 (16)	[124] 38.2 (4)	[134] 32.2 (0)	[64] 35.2 (0)	-	-	-	[64] 19.2 (0)	[145] 12.2 (0)	-	[153] 39.4 (0)	[87] 11.6 (0)	-	-	[94] 66.1 (0)	[94] 40.7 (0)
Flamingo-3B	✗	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
	✗	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
	✗	8	44.6	55.4	90.6	37.0	54.5	38.4	71.7	19.6	36.8	68.0	40.6	47.6	32.4	23.9	54.7	-
	✗	16	45.6	56.7	95.4	40.2	57.1	43.3	73.4	23.4	37.4	73.2	40.1	47.5	31.8	25.2	55.3	-
	✗	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	OOC	30.6	26.1	56.3	-
Flamingo-9B	✗	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
	✗	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	42.8	50.4	33.6	24.7	62.7	-
	✗	8	50.0	58.0	99.0	40.8	55.2	39.4	73.4	23.9	40.0	75.0	**43.4**	51.2	33.6	25.8	63.9	-
	✗	16	50.8	59.4	102.2	44.5	58.5	43.0	72.7	27.6	41.5	77.2	42.4	51.3	33.5	27.6	64.5	-
	✗	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	OOC	32.6	28.4	63.5	-
Flamingo	✗	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	**60.8**
	✗	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	55.6	36.5	30.8	68.6	-
	✗	8	57.5	65.6	108.8	45.5	60.6	44.8	78.2	27.6	44.8	80.7	42.3	56.4	37.3	32.3	**70.0**	-
	✗	16	57.8	66.8	110.5	48.4	62.8	48.4	**78.9**	30.0	45.2	84.2	41.1	**56.8**	37.6	32.9	**70.0**	-
	✗	32	**57.8**	**67.6**	**113.8**	**52.3**	**65.1**	**49.8**	75.4	**31.0**	**45.3**	**86.8**	42.2	OOC	**37.9**	**33.5**	**70.0**	-
Pretrained FT SOTA	✔	(X)	54.4 [39] (10K)	80.2 [150] (444K)	143.3 [134] (500K)	47.9 [32] (27K)	76.3 [165] (500K)	57.2 [70] (20K)	67.4 [162] (30K)	46.8 [57] (130K)	35.4 [145] (6K)	138.7 [142] (10K)	36.7 [138] (46K)	75.2 [87] (123K)	54.7 [147] (20K)	25.2 [139] (38K)	75.4 [60] (9K)	-

Table 3 | **Comparison to the state of the art on multimodal benchmarks.** A *single* Flamingo model reaches state-of-the-art on a wide array of image and video tasks with in-context learning from as few as 4 examples per task, beating previous zero-shot or few-shot method by a large margin. More importantly, using only 32 examples and without adapting any model weight, Flamingo *outperforms* the current best methods on 7 tasks, that are fine-tuned on thousands of annotated examples. Best few-shot numbers are in **bold**. Best numbers overall are underlined. See also Figure 2 that illustrate the table. OOC: out-of-context, which happens when the few-shot prompt is longer than the maximum sequence length the model has been trained on.

Microsoft COCO Captions: Data Collection and Evaluation Server

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam
Saurabh Gupta, Piotr Dollár, C. Lawrence Zitnick

Abstract—In this paper we describe the Microsoft COCO Caption dataset and evaluation server. When completed, the dataset will contain over one and a half million captions describing over 330,000 images. For the training and validation images, five independent human generated captions will be provided. To ensure consistency in evaluation of automatic caption generation algorithms, an evaluation server is used. The evaluation server receives candidate captions and scores them using several popular metrics, including BLEU, METEOR, ROUGE and CIDEr. Instructions for using the evaluation server are provided.

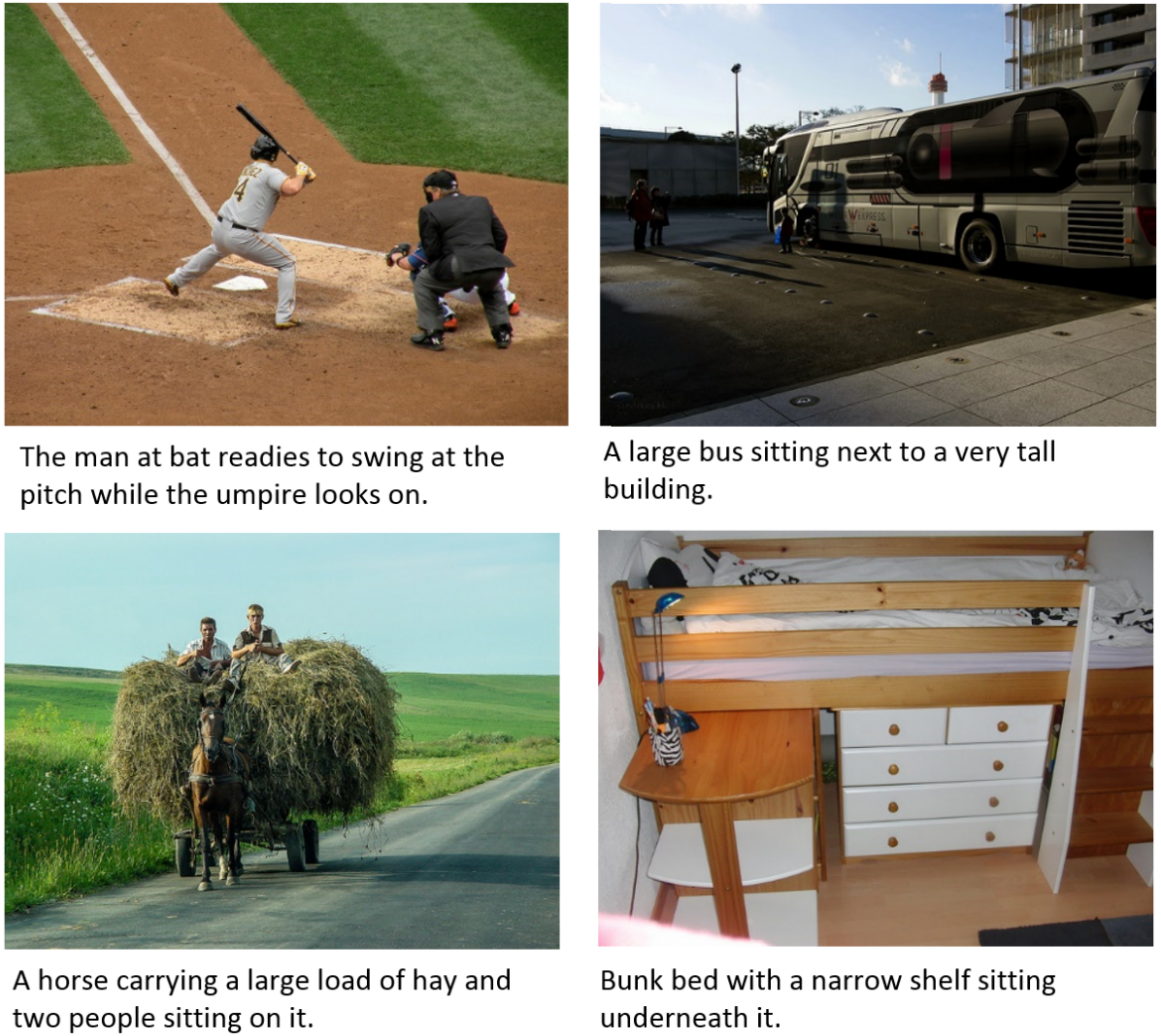


Fig. 1: Example images and captions from the Microsoft COCO Caption dataset.

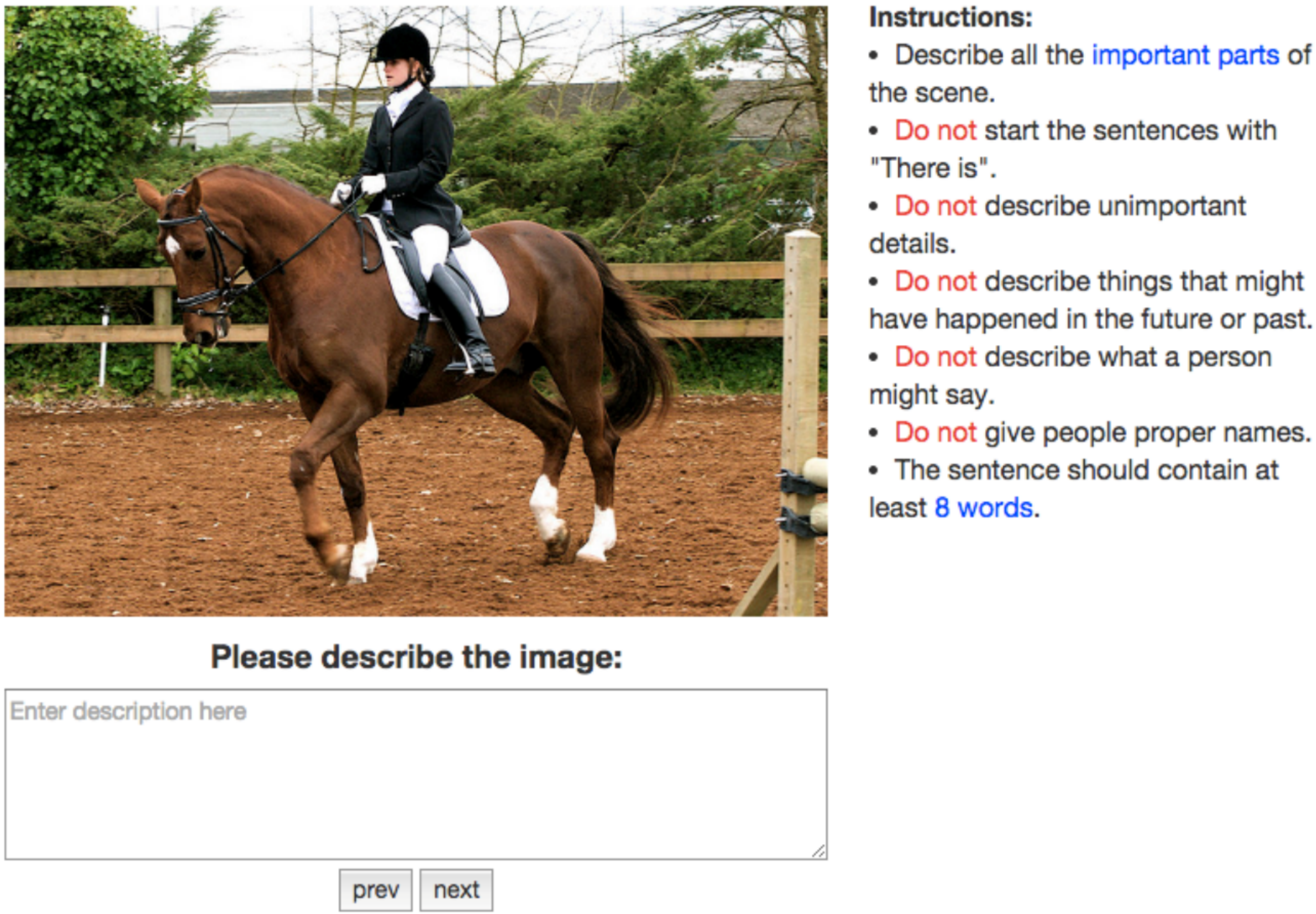


Fig. 2: Example user interface for the caption gathering task.

Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering

Yash Goyal*[1] Tejas Khot*[1] Douglas Summers-Stay[2] Dhruv Batra[3] Devi Parikh[3]
[1]Virginia Tech [2]Army Research Laboratory [3]Georgia Institute of Technology
[1]{ygoyal, tjskhot}@vt.edu [2]douglas.a.summers-stay.civ@mail.mil [3]{dbatra, parikh}@gatech.edu

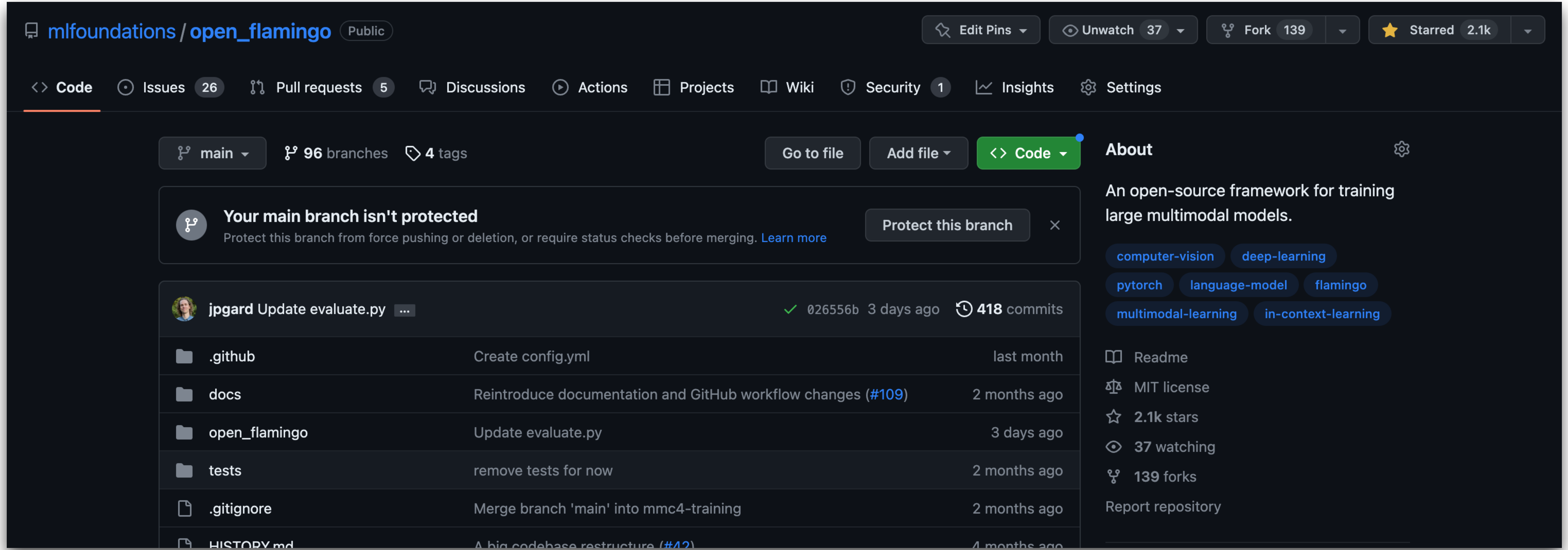
Abstract

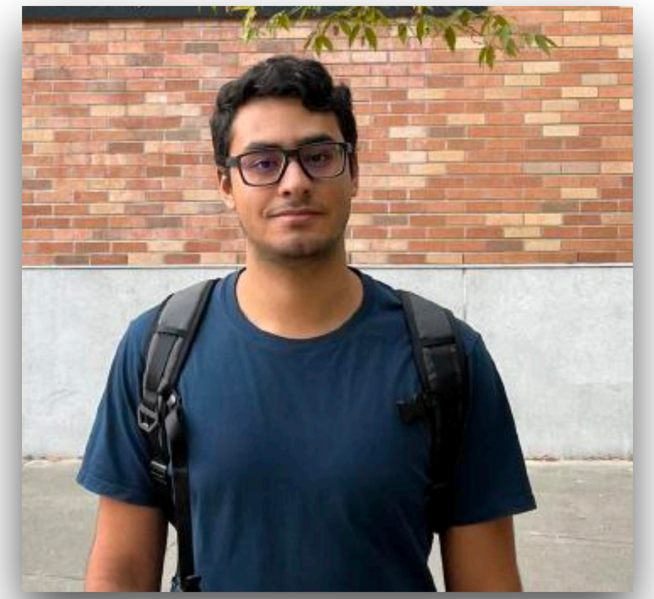
Problems at the intersection of vision and language are of significant importance both as challenging research questions and for the rich set of applications they enable. However, inherent structure in our world and bias in our language tend to be a simpler signal for learning than visual modalities, resulting in models that ignore visual information, leading to an inflated sense of their capability.

We propose to counter these language priors for the task of Visual Question Answering (VQA) and make vision (the V in VQA) matter! Specifically, we balance *the popular VQA dataset [3] by collecting complementary images such that every question in our balanced dataset is associated with*

Figure 1: Examples from our balanced VQA dataset.

Figure 2: Random examples from our proposed balanced VQA dataset. Each question has two similar images with different answers to the question.





Anas Awadalla

Irena Gao