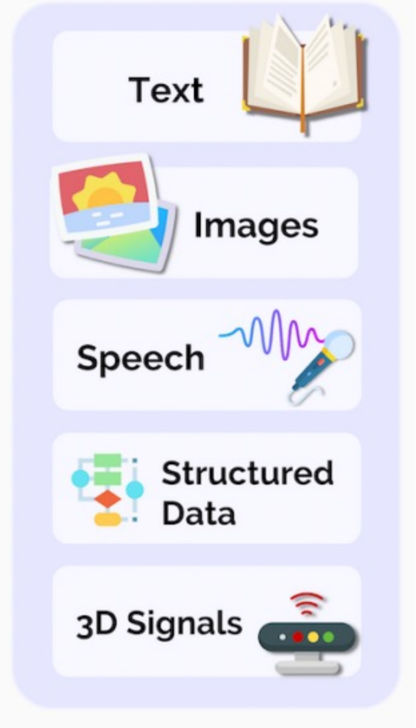
Rethinking fine-tuning to mitigate feature distortion

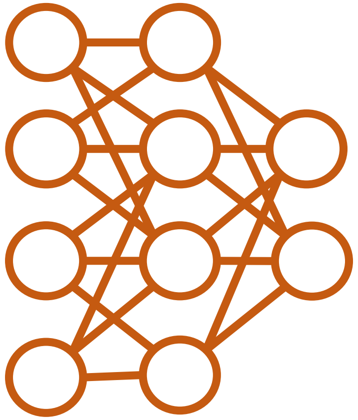
Aditi Raghunathan

Carnegie Mellon University

Large-scale pretraining



Step one: pretraining



Step two: adaptation

Diverse (typically unlabeled) data

Pretrained model

Specialize to narrow distribution

Bommasani et al. 2021

Robustness to distribution shifts

A core challenge for reliable machine learning in the wild

Train

Pedestrians using a crosswalk

Deploy

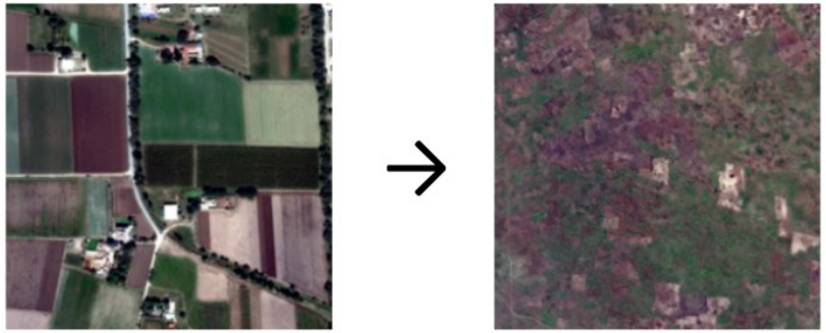
Skateboarders

Important pedestrians

Pedestrians jaywalking

Distribution shifts are everywhere

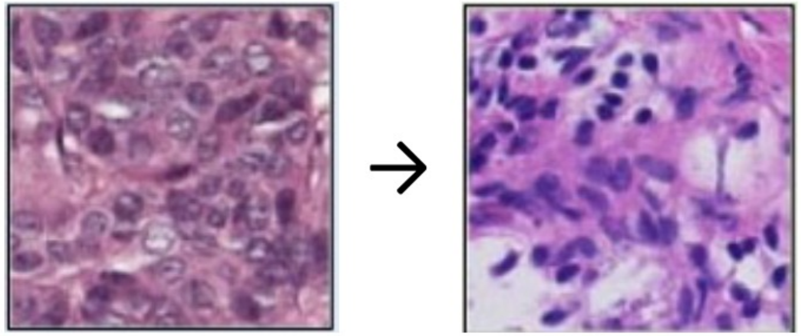
Train **Deploy**



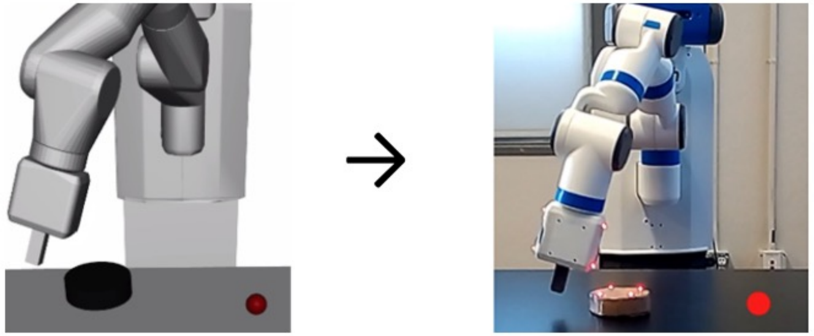
Satellite remote sensing (different regions)

Train **Deploy**

Wildlife conservation (different forests)



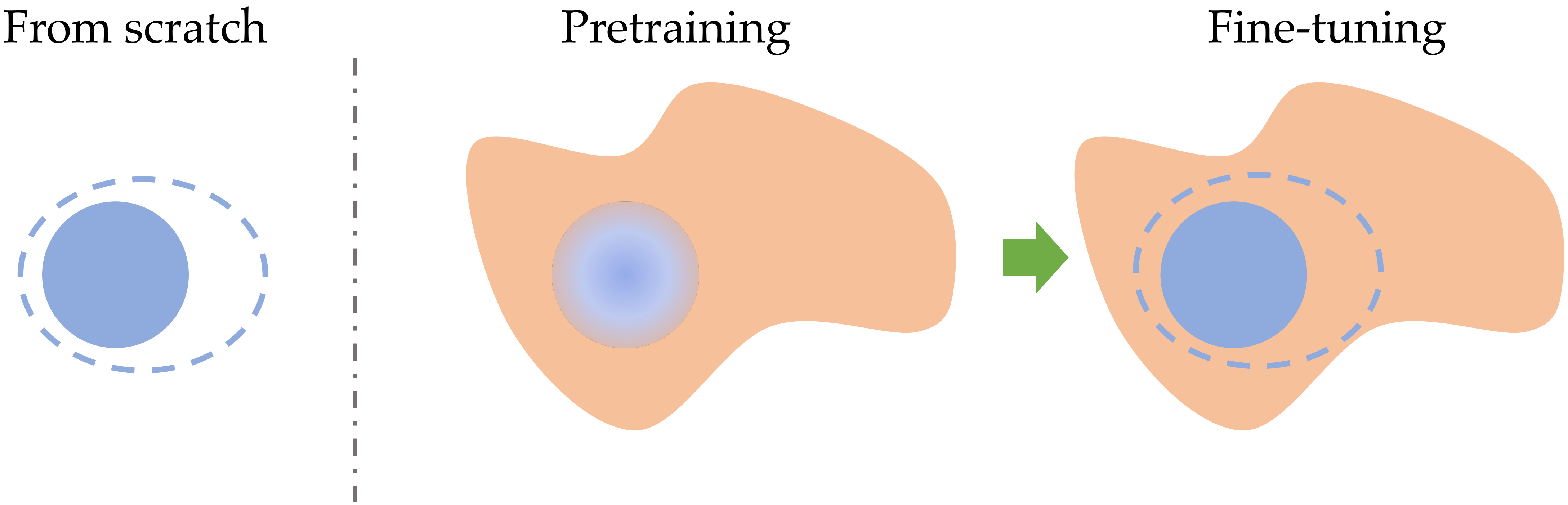
Tumor detection (new hospitals)



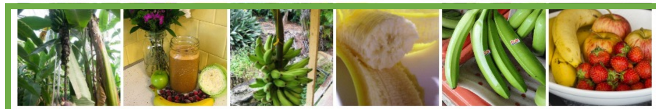
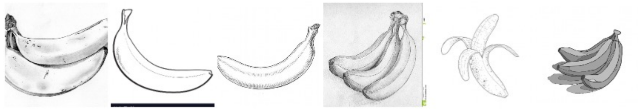
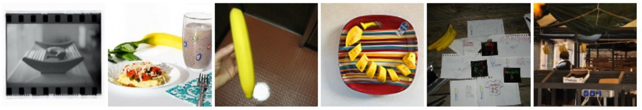
Sim-to-real

Christie et al. 2017, Beery et al. 2021, Bandi et al. 2018, Koh et al. 2021, Peng et al. 2018

The generalization challenge



The promise of large-scale pretraining

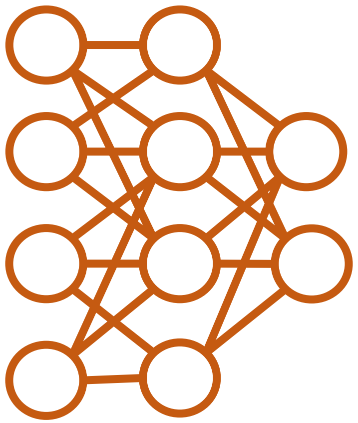
Dataset	ImageNet ResNet101	CLIP ViT-L	
 ImageNet	76.2%	76.2%	
 ImageNet V2	64.3%	70.1%	
 ImageNet Rendition	37.7%	88.9%	+51.2%
 ObjectNet	32.6%	72.3%	
 ImageNet Sketch	25.2%	60.2%	
 ImageNet Adversarial	2.7%	77.1%	+74.4%

More data generally helps

The generalization problem revisited

Diverse (typically unlabeled) data

Step one: pretraining



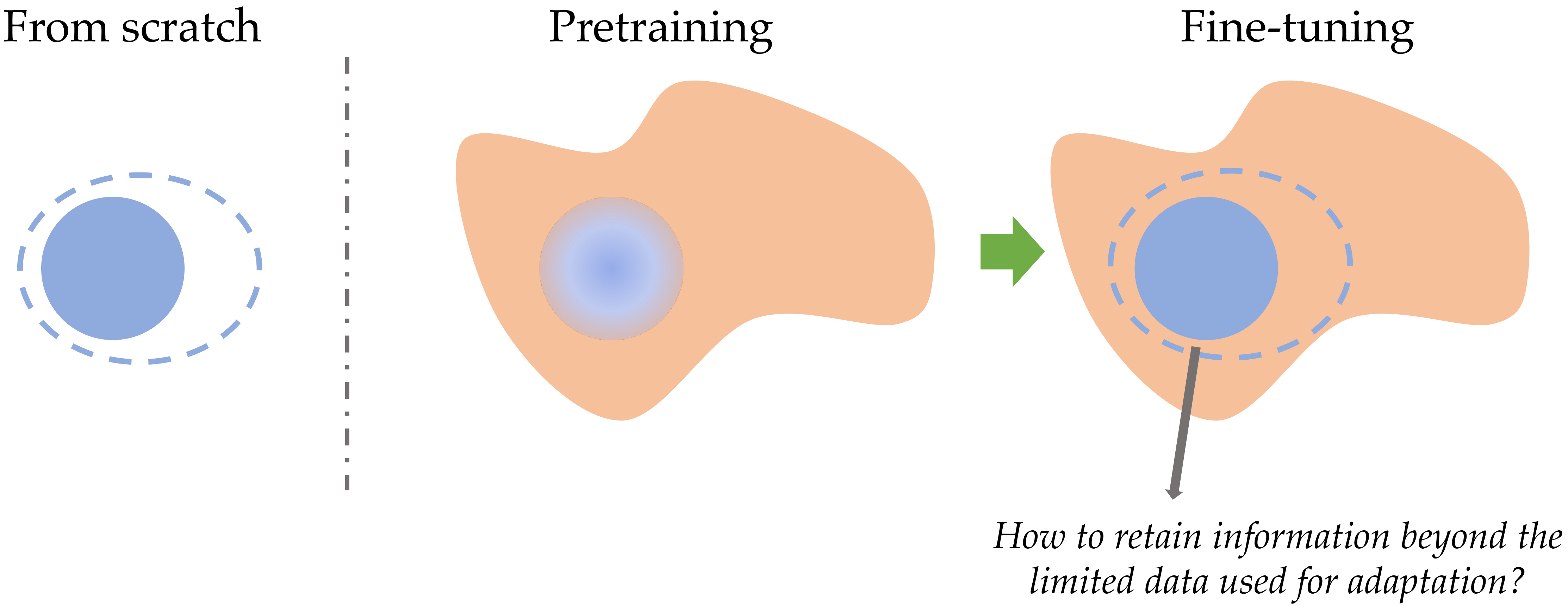
Pretrained model

Step two: adaptation

Specialize to narrow distribution

Supervision during adaptation is still coming from limited data

The generalization challenge revisited



The "art" of neural network training

- What parameters to update (model family)
- Loss function
- Optimization hyperparameters

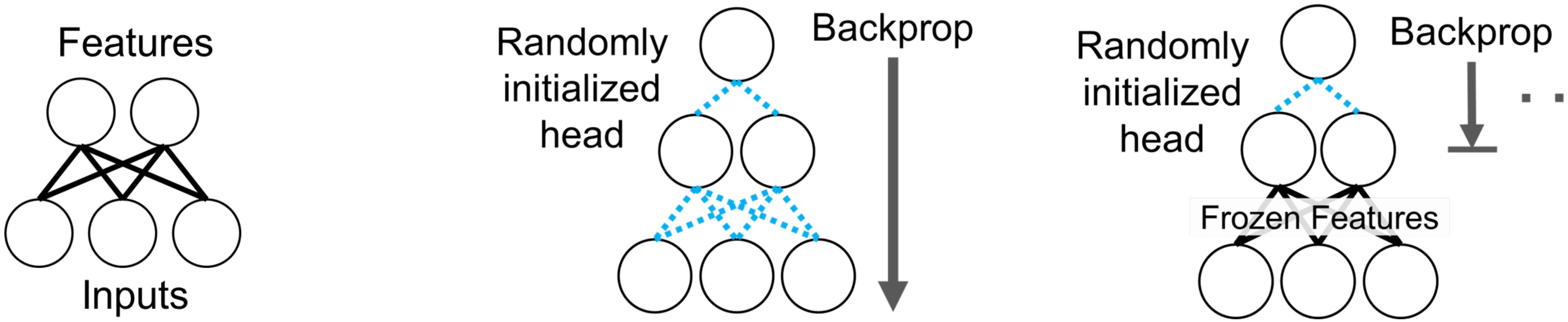
The “art” of neural network training

- **What parameters to update (model family)**
- Loss function
- Optimization hyperparameters

Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Linear probing vs (full) fine-tuning

Pretraining | **Fine-tuning** | **Linear probing**



Pop quiz!

Dataset: BREEDS Living-17

Task: classify into animal categories

Train distribution: one subset of ImageNet hierarchy tree with animal category as root

Test distribution: other subset of ImageNet hierarchy tree with animal category as root

Pretrained model: MoCo-V2, which has seen *unlabeled* ImageNet images (including various types of animals)

Train

Test

Santurkar et al. 2020

Pop quiz: living-17

Living-17	ID	OOD
Scratch	92.4%	58.2%
Linear probing	96.5%	?
Fine-tuning	97.1%	

Does linear probing do better than scratch OOD?

Pop quiz: living-17

Living-17	ID	OOD
Scratch	92.4%	58.2%
Linear probing	96.5%	82.2%
Fine-tuning	97.1%	

Does linear probing do better than scratch OOD?

Yes!

Pop quiz: living-17

Living-17	ID	OOD
Scratch	92.4%	58.2%
Linear probing	96.5%	82.2%
Fine-tuning	97.1%	?

Does fine-tuning do better than linear probing OOD?

Pop quiz: living-17

Living-17	ID	OOD
Scratch	92.4%	58.2%
Linear probing	96.5%	82.2%
Fine-tuning	97.1%	77.7%

Does fine-tuning do better than linear probing OOD?

No!

Dataset: CIFAR 10.1

Task: classify into CIFAR-10 categories

Train distribution: original CIFAR-10 dataset

Test distribution: recent near-replication of the pipeline

Pretrained model: MoCo-V2, which has seen *unlabeled* ImageNet images

Recht et al. 2019

Pop quiz: CIFAR10.1

Living-17	ID	OOD
Linear probing	91.8%	82.7
Fine-tuning	97.3%	?

Does linear probing do better than fine-tuning OOD?

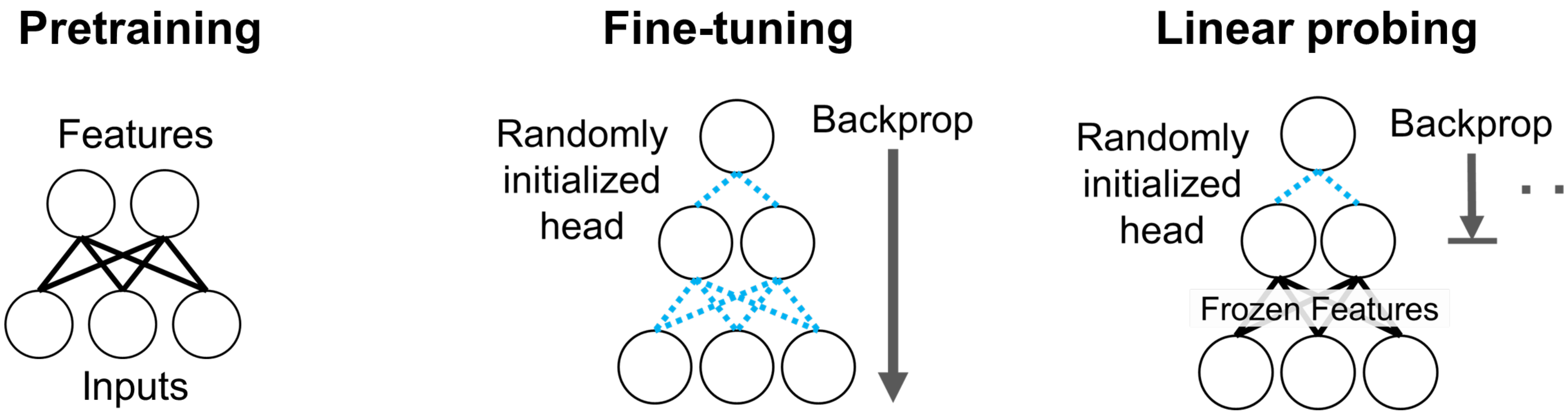
Pop quiz: CIFAR10.1

Living-17	ID	OOD
Linear probing	91.8%	82.7
Fine-tuning	97.3%	92.3%

Does linear probing do better than fine-tuning OOD?

No!

Linear probing vs fine-tuning summary



Which method does better?

Linear probing vs fine-tuning summary

	ID	OOD
Linear probing	82.9%	
Fine-tuning	85.1%	

Averaged over 10 datasets

Common wisdom is fine-tuning works better than linear probing

Linear probing vs fine-tuning summary

	ID	OOD
Linear probing	82.9%	66.2%
Fine-tuning	85.1%	59.3%

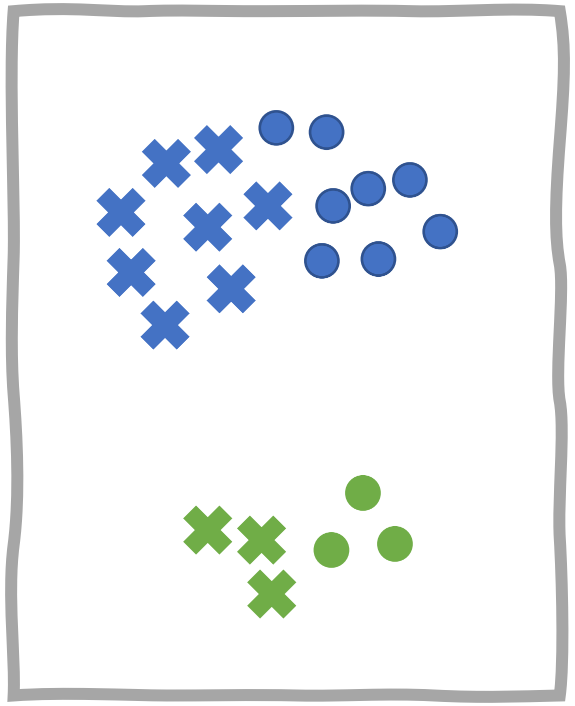
Averaged over 10 datasets

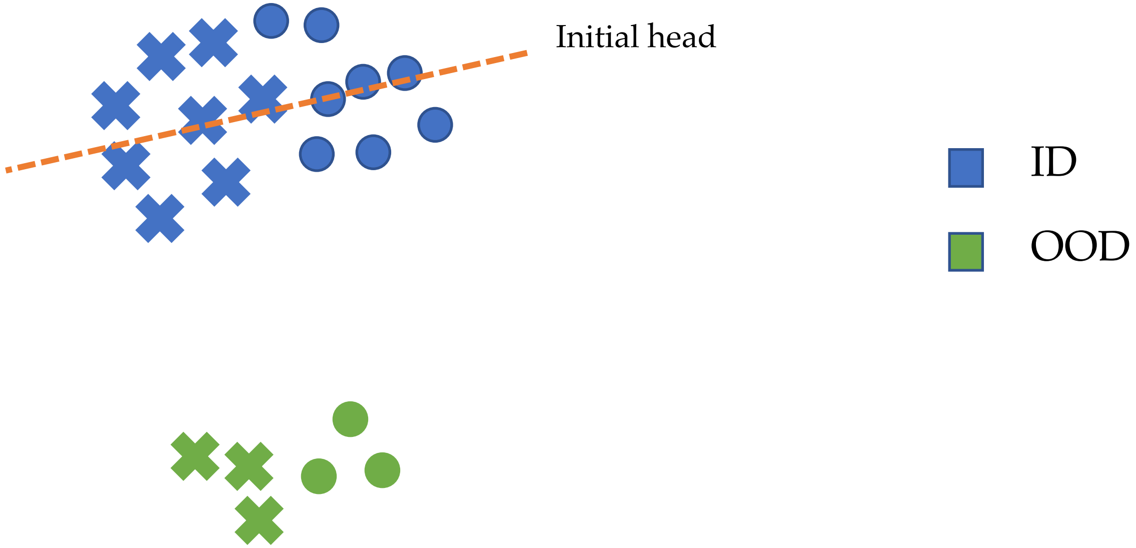
LP performs better than FT OOD on 8 out of 10 datasets

Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Intuition for theoretical result

Pretrained
Features

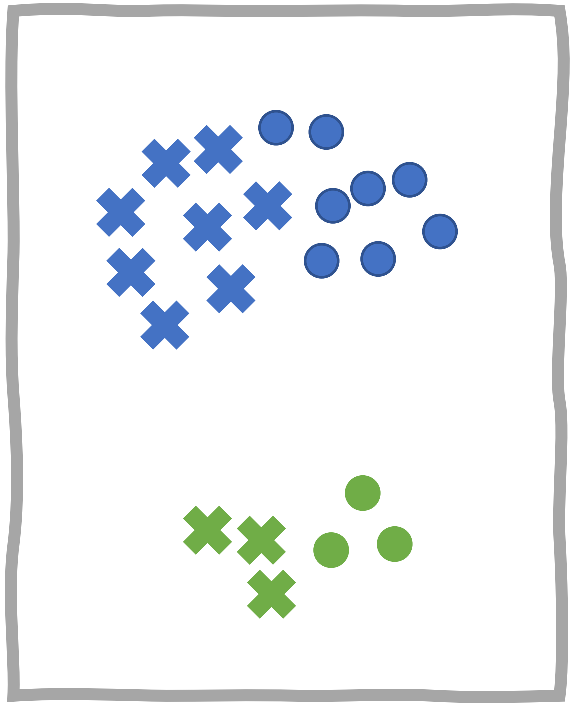




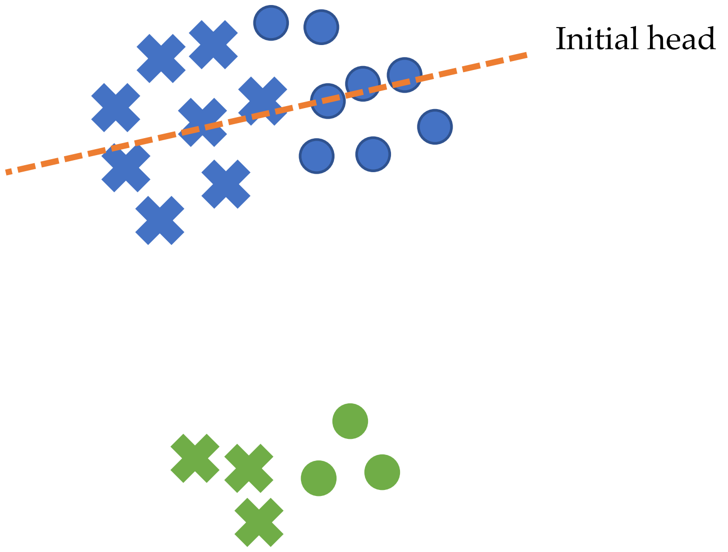
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

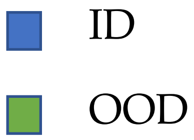
Intuition for theoretical result

Pretrained Features



Fine-tuning: features for ID examples change in sync with the linear head



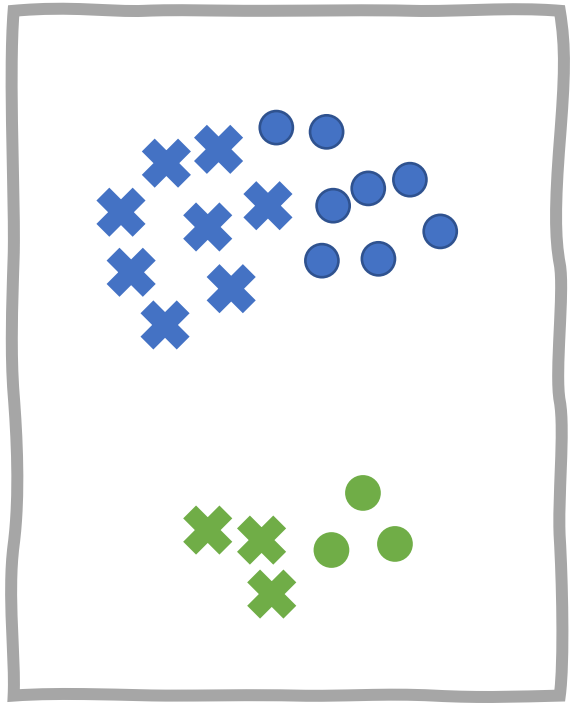


Features for OOD examples change less

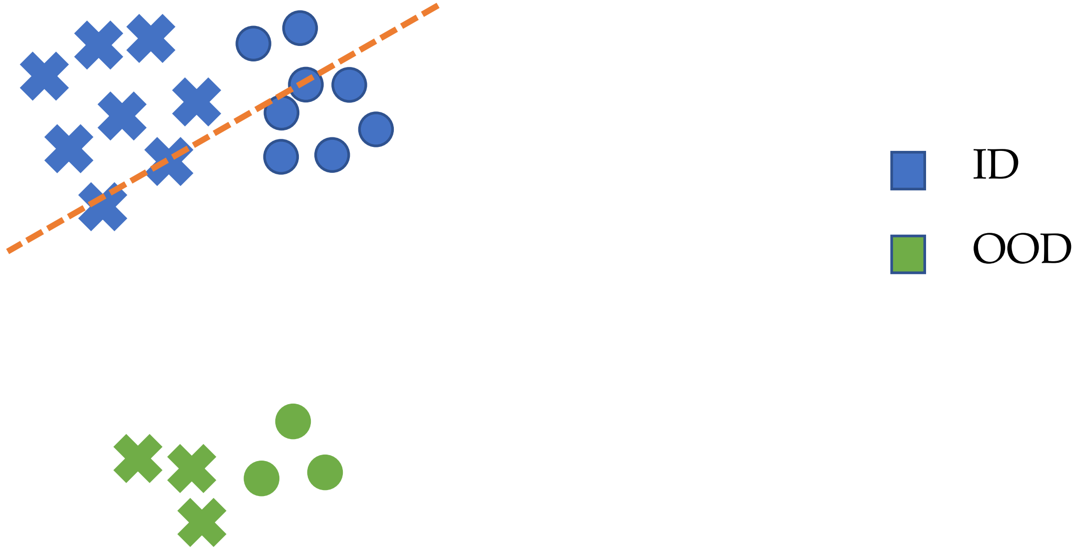
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Intuition for theoretical result

Pretrained Features



Fine-tuning: features for ID examples change in sync with the linear head

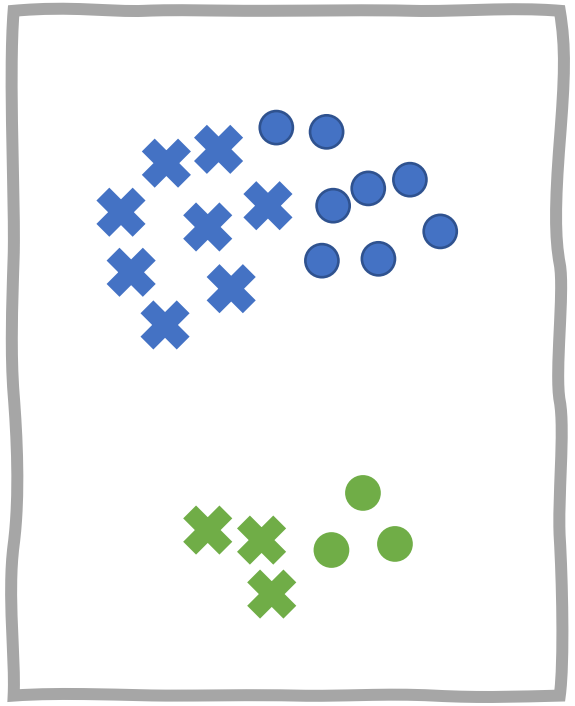


Features for OOD examples change less

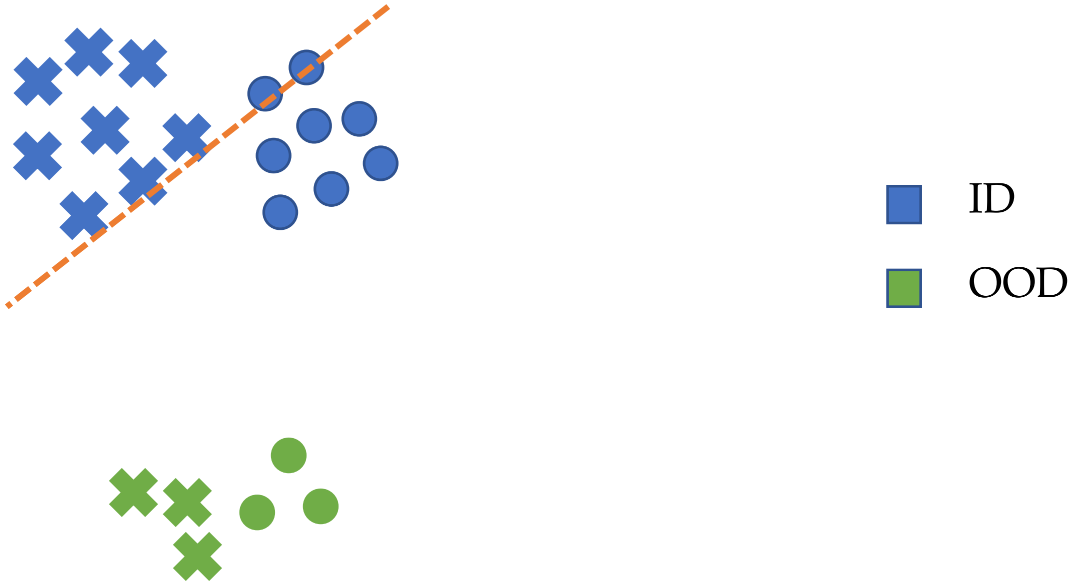
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Intuition for theoretical result

Pretrained
Features



Fine-tuning: features for ID examples change in sync with the linear head

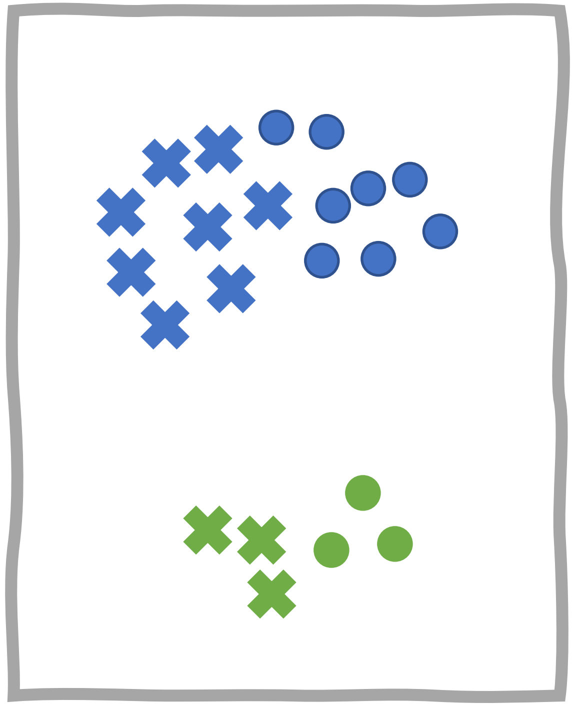


Features for OOD examples change less

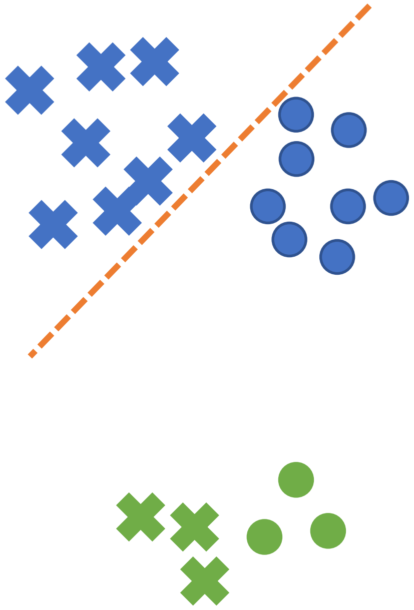
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

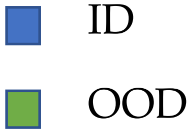
Intuition for theoretical result

Pretrained
Features



Fine-tuning: features for ID examples change in sync with the linear head



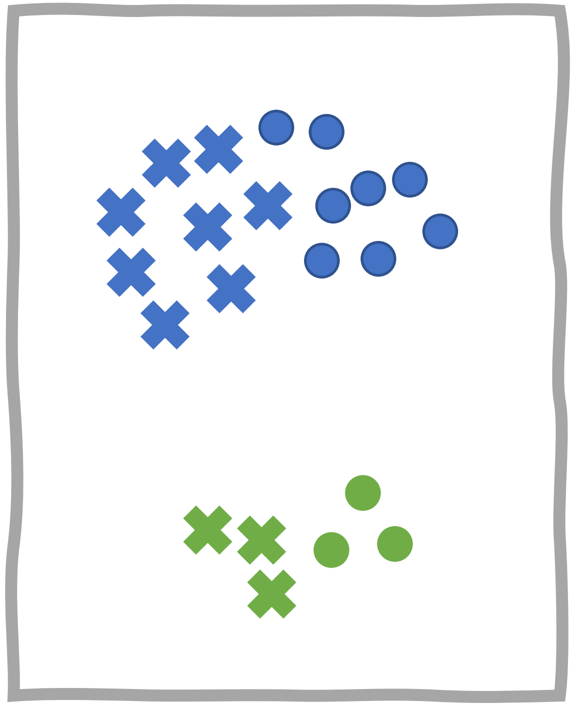


Features for OOD examples change less

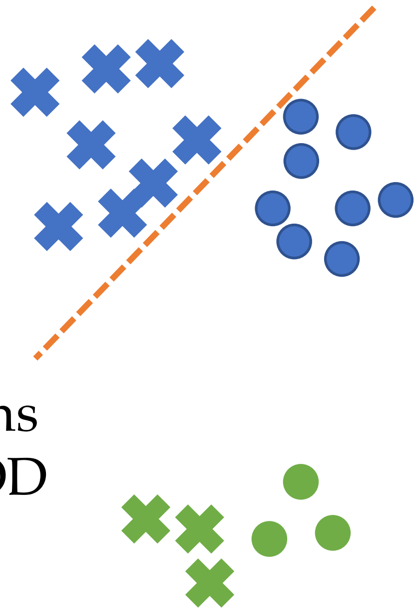
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Intuition for theoretical result

Pretrained Features

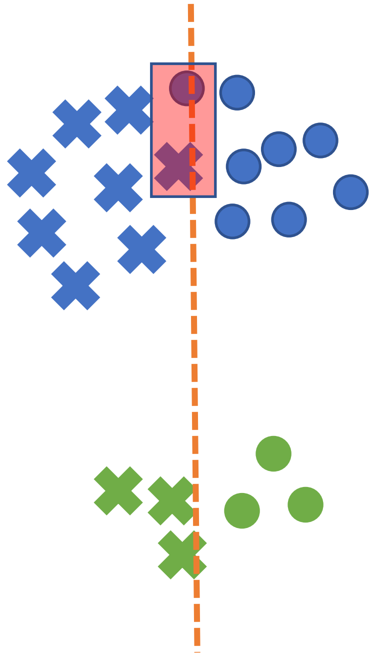


Fine-tuning: features for ID examples change in sync with the linear head



Head performs poorly on OOD examples

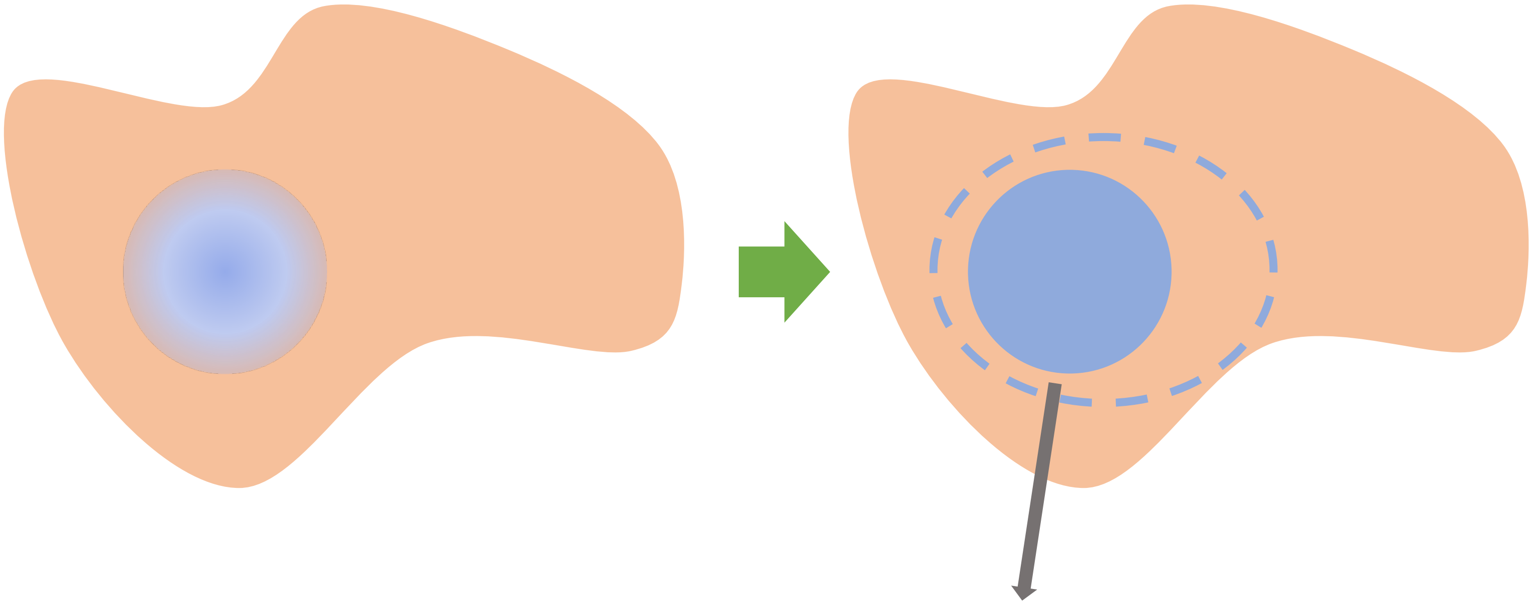
Linear probing: freezes pretrained features



Head is decent on OOD examples

Key takeaway

A larger change in parameters can **distort** pretrained features



How to retain information beyond the limited data used for adaptation?

Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Best of both worlds

Why does FT do better ID?

Training data may not be linearly separable in the space of pre-trained features i.e. imperfect pre-trained features

Why does FT do worse OOD?

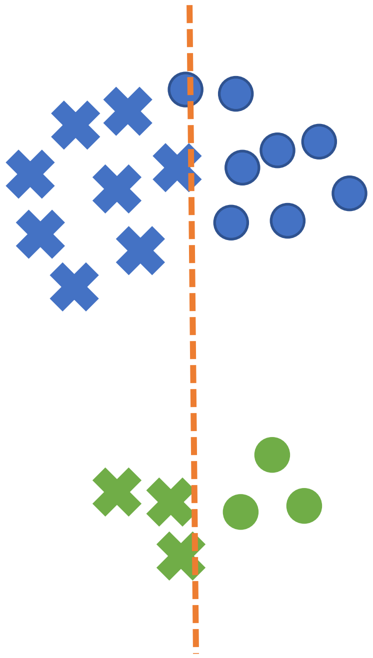
Features can change a lot to accommodate a randomly initialized head

Can we refine features without distorting them too much?

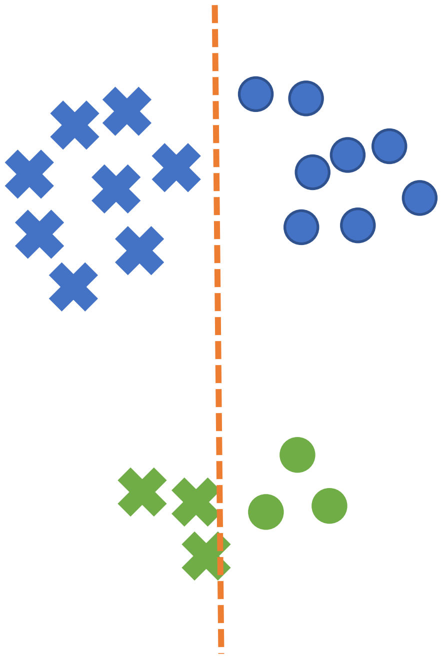
Method to achieve best of both worlds

Idea: modify pre-trained features **only as necessary**

Step 1: Linear probe



Step 2: Fine-tune



Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Method to achieve best of both worlds

Idea: modify pre-trained features **only as necessary**

Step 1: Linear probe

Step 2: Fine-tune

LP-FT method

Can prove that LP-FT dominates both LP and FT under the simple setting of perfect features

Improving fine-tuning

	ID	OOD
Linear probing	82.9%	66.2%
Fine-tuning	85.1%	59.3%
LP-FT	**85.7%**	**68.9%**

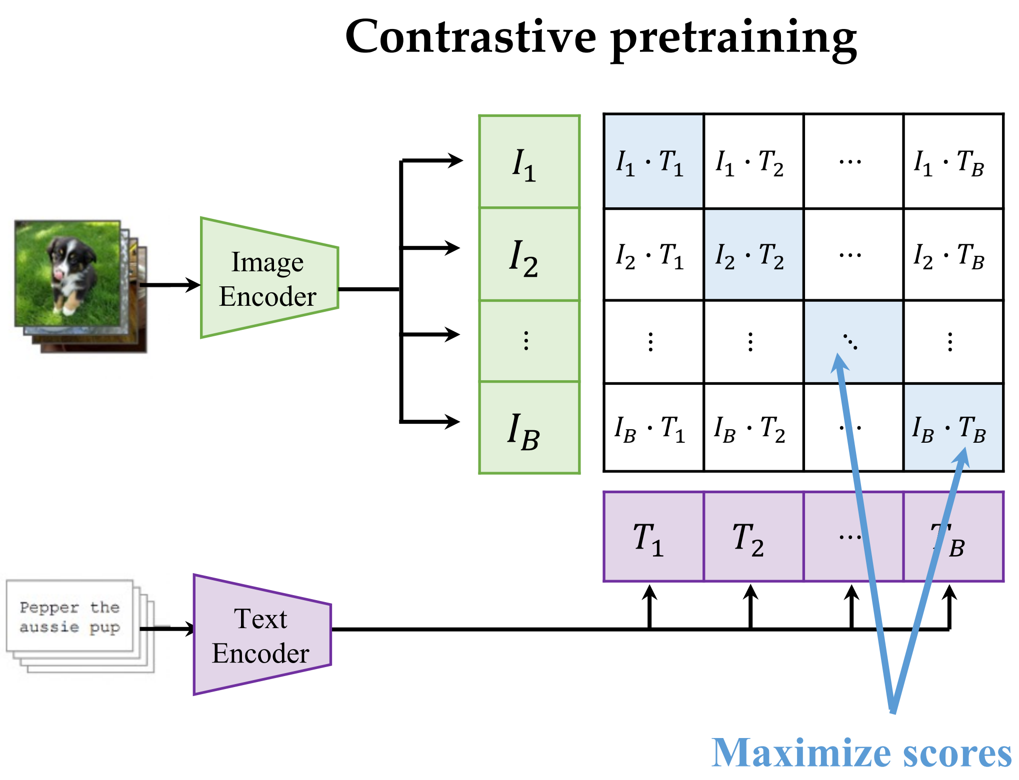
+10% over fine-tuning!

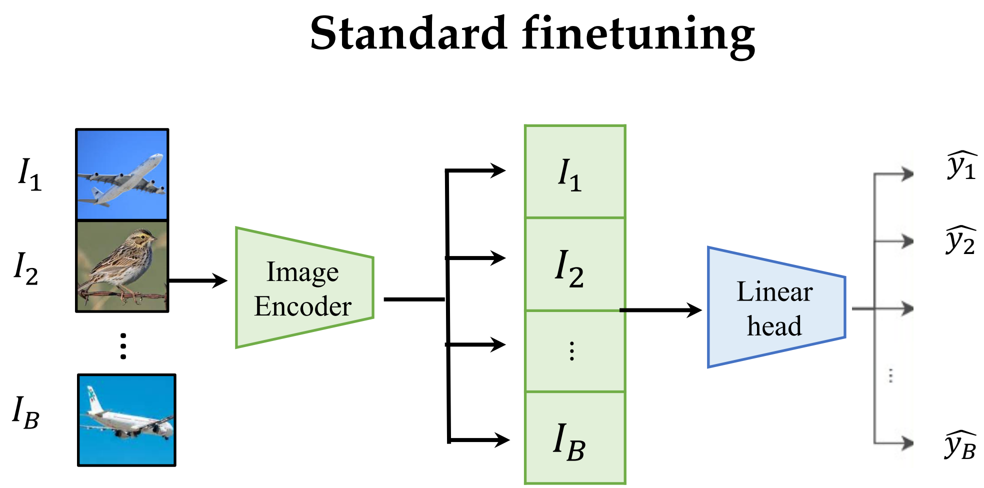
LP-FT obtains better than the best of both worlds

The "art" of neural network training

- What parameters to update (model family)
- **Loss function**
- Optimization hyperparameters

The loss function

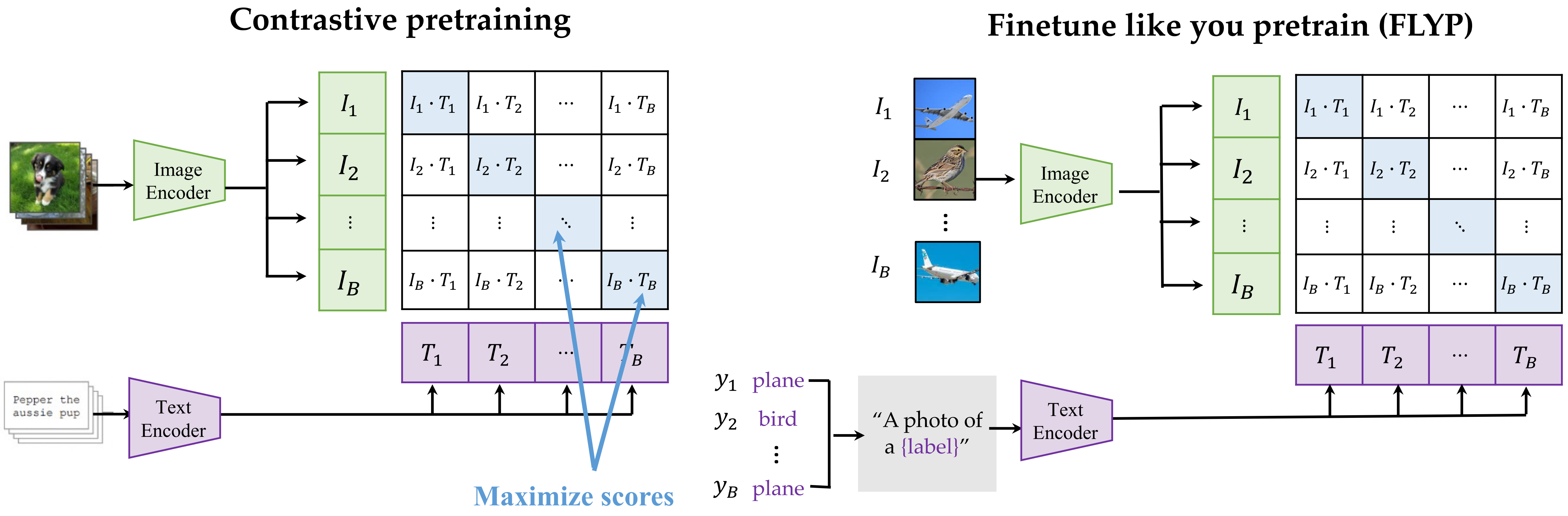




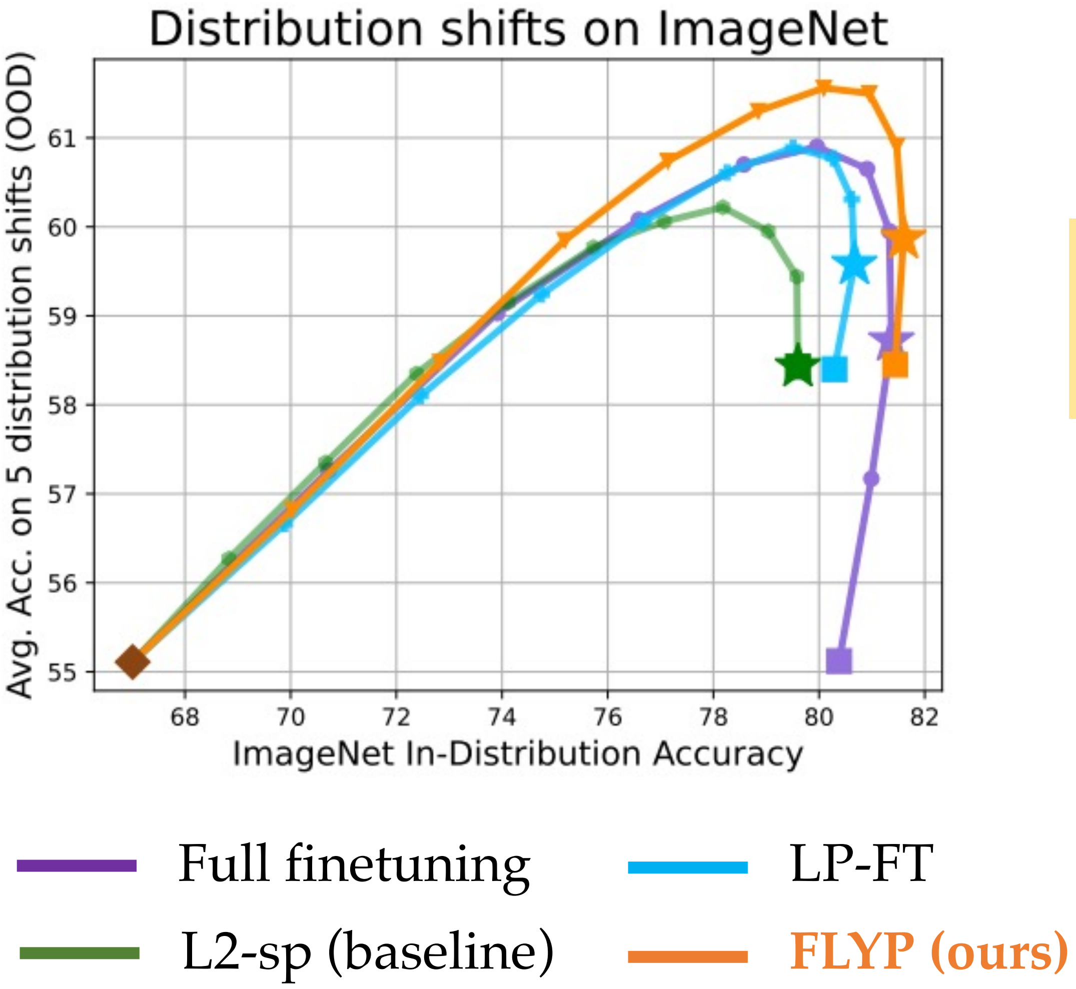
Can we reduce distortion?

Goyal, Kumar, Garg, Kolter, **Raghunathan**. *Finetune like you pretrain: improved finetuning of zero-shot vision models. CVPR* 2023.

Revisiting the fine-tuning loss function



Fine-tune like you pretrain



Same pretraining loss can reduce distortion and improve robustness

Fine-tune like you pretrain

Also see gains in few-shot learning

	PatchCamelyon	SST2
Zero shot	56.5%	60.5%
FT	63.1%	61.1%
LP-FT	62.7%	60.9%
FLYP	**66.9%**	**61.3%**

Summary

- Pretrained models give large improvements in accuracy, but how we fine-tune them is key
- **General principle: minimize distortion while fine-tuning**
- Two simple ways to do that
  - LP-FT (only change features once the head is trained)
  - FLYP (keep the fine-tuning loss identical to the pretraining loss)

Thanks!


Ananya Kumar


Robbie Jones


Tengyu Ma


Percy Liang


Sachin Goyal


Sankalp Garg


Zico Kolter

Apple

Google

Schmidt Futures

Open Philanthropy