Estimating and improving the robustness of ML under **natural distribution shifts**

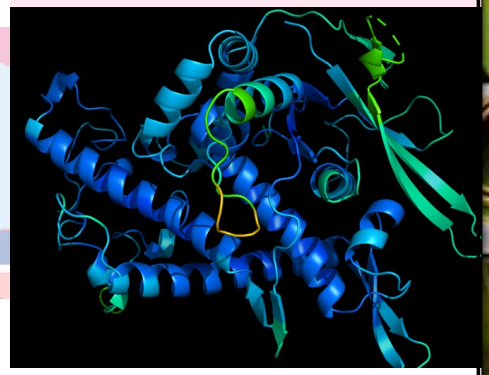
Aditi Raghunathan

Carnegie Mellon University

Google's AlphaGo Defeats Chinese Go Master in Win for A.I.

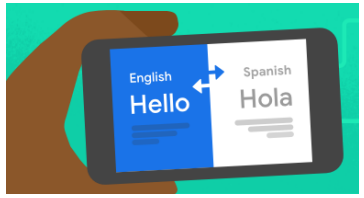
OpenAI
GPT-3

AlphaFold: a solution to a 50-year-old grand challenge in biology



ML is widely used

Translation

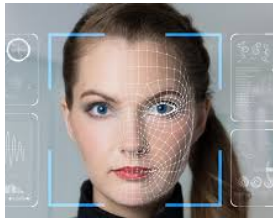


Spam detection

Voice assistant

Stock market

Facial recognition



Hiring systems

Autonomous driving

2015

Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software

2018

Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian

2020

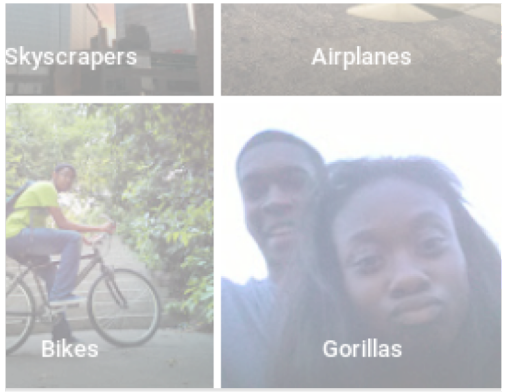
Another Arrest, and Jail Time, Due to a Bad Facial Recognition Match

A New Jersey man was accused of shoplifting and trying to hit an officer with a car. He is the third known Black man to be wrongfully arrested based on face recognition.

AI mistakes referee's bald head for football — hilarity ensued

2015

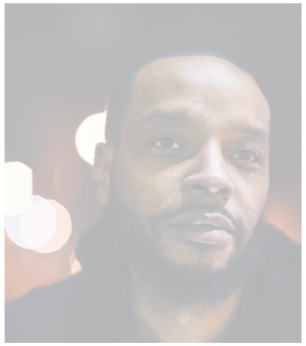
Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software



2018

Self-driving Uber kills Arizona woman in olving pedestrian

2020



Another Arres Bad Facial Re

A New Jersey man was accused of shoplifting and trying to hit an officer with a car. He is the third known Black man to be wrongfully arrested based on face recognition.

Current ML is brittle

AI mistakes referee's bald head for football — hilarity ensued

On a quest to improve robustness

Classical ML pipeline

Insufficient!

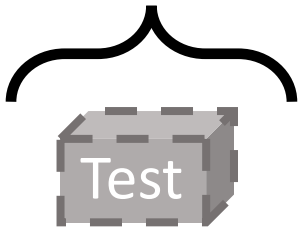
Evaluate only on **examples** from **P**

Most benchmarks and theory study this setting

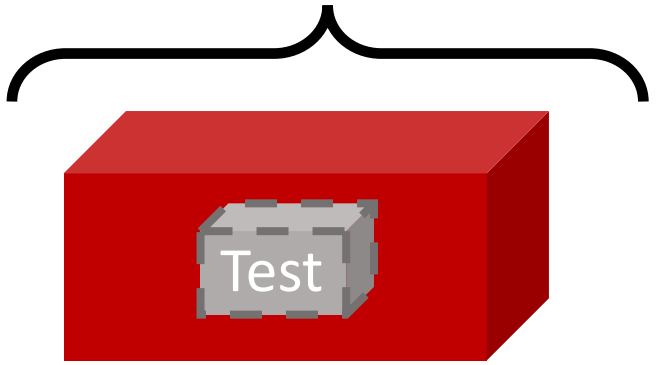
Classical ML pipeline

Insufficient!

Evaluate only on **examples** from **P**



What we really care about!



What do we care about?

Driving in California

Driving in Pittsburgh

Work reliably in **all** weather conditions

What do we care about?

Pedestrians using a crosswalk

Pedestrians jaywalking

Skateboarders

Important pedestrians

Work reliably with **all** pedestrian activity

What do we care about?

Work well in **all** locations

What do we care about?

Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software

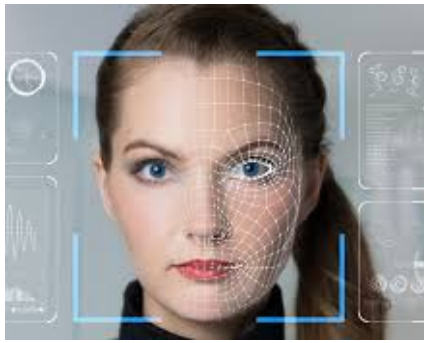
Another Arrest, and Jail Time, Due to a Bad Facial Recognition Match

A New Jersey man was accused of shoplifting and trying to hit an officer with a car. He is the third known Black man to be wrongfully arrested based on face recognition.

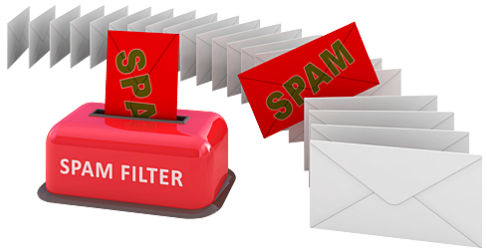
Work well across **all** subpopulations

What do we care about?

Content filtering



Facial recognition



Spam filtering

Work well across **all** adversaries trying to break the system

What do we care about?

Work reliably in all weather conditions

Work well across all subpopulations

Work well in all locations

Work well across all adversaries trying to break the system

Work reliably with all pedestrian activity

Good performance in worst-case

Robust ML formulation

Classical ML evaluation

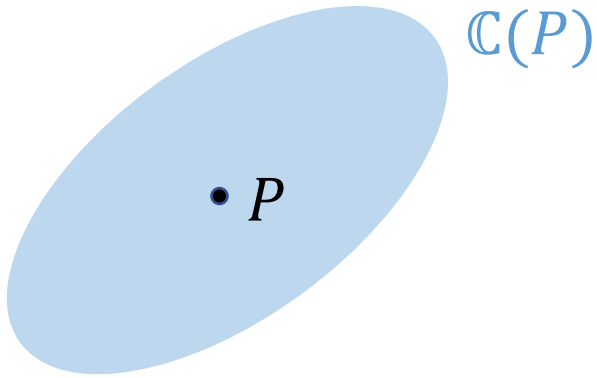
Training data: $z_1, z_2, \ldots z_n \sim P$

Test (θ): $E_{z\sim P}[l(z;\theta)]$

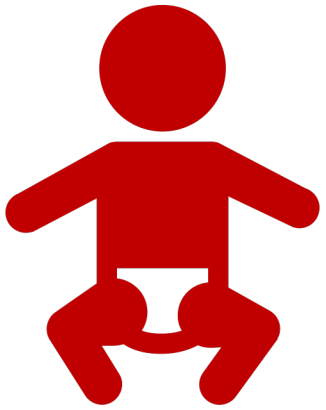
Distributionally robust ML

Training data: $z_1, z_2, \ldots z_n \sim P$

Test (θ): $\max_{Q \in \mathbb{C}(P)} E_{z\sim Q}[l(z;\theta)]$



First few years of my PhD

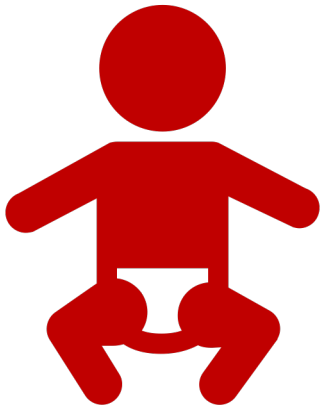


*Let us understand the fundamentals of robust optimization for **any** known uncertainty set* $\mathbb{C}(P)$

Next, we can think about approximating real-world distribution shifts as a union of different uncertainty sets

Put the pieces together and we are done!

Next year of my PhD



Let us understand the fundamentals of robust optimization for ~~***any***~~ ***many*** *known uncertainty sets* $\mathbb{C}(P)$

Next, we can think about approximating real-world distribution shifts as a union of different uncertainty sets

Put the pieces together and we are done!

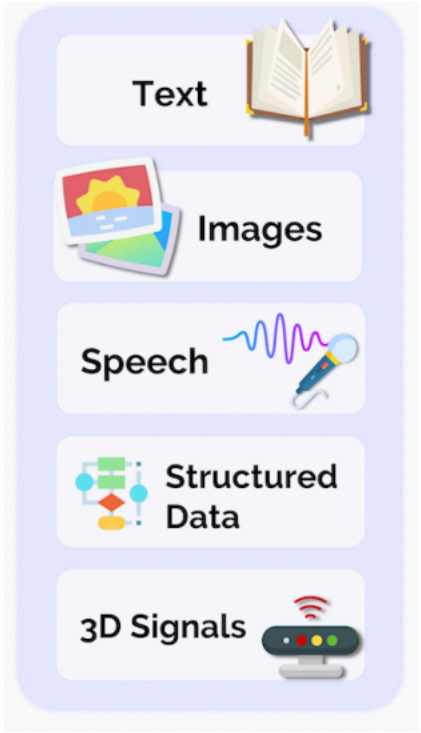
Realizations over time

- Every step is extremely hard and potentially “very lossy”
- Worth simultaneously exploring other approaches, especially in the short run?

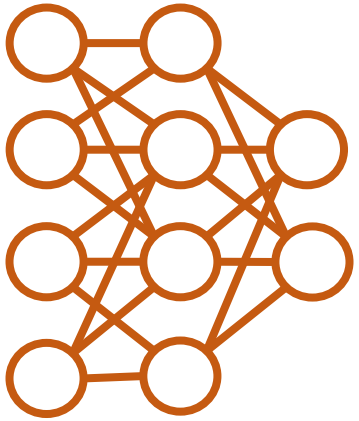
Can we tackle natural distribution shifts directly?

Pre-training on diverse data

The pre-training setup



Step one: pretraining



Step two: adaptation

Diverse (typically unlabeled) data

Pre-trained model

Specialize to narrow distribution

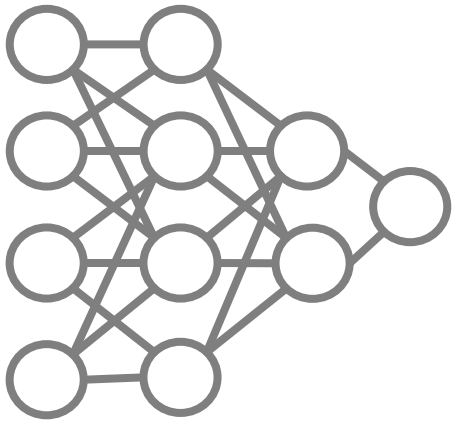
Bommasani et al. 2021

Why pre-training?

Consider satellite remote sensing task

We have training data from **North America**, but very limited data from **Africa**

Standard supervised learning



"In-distribution" training data

North America

Performs poorly on **OOD test data** from Africa

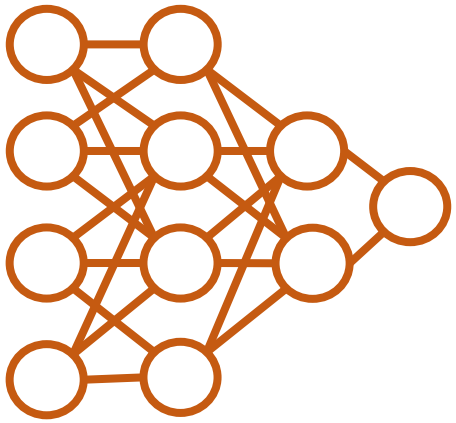
Why pre-training?

Consider satellite remote sensing task

We have training data from **North America**, but very limited data from **Africa**

Transfer learning setting

Pre-trained model



"In-distribution" training data

North America

Performs better on **OOD test data** from Africa

Why pre-training?

Pre-training incorporates information from more diverse data than standard supervised learning

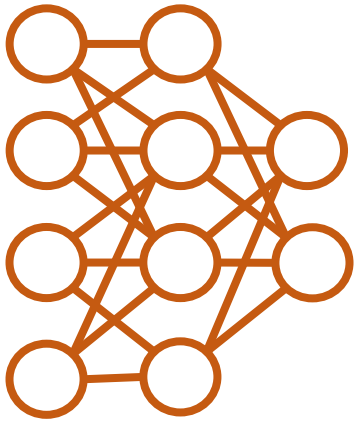
Pre-training typically improves in-distribution performance

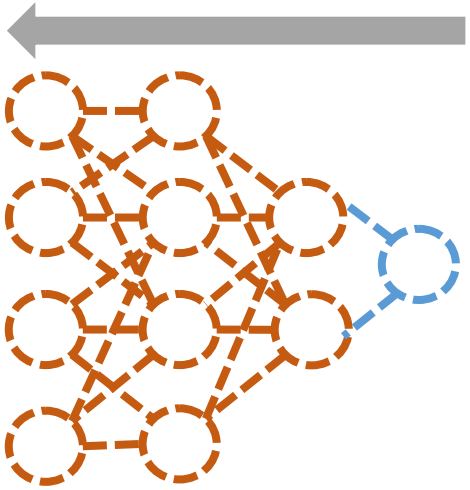
Pre-training substantially improves out-of-distribution performance

One of the most reliable methods to improve robustness across several natural shifts

How to use pre-trained models?

How to leverage the diverse information contained in pre-trained models?



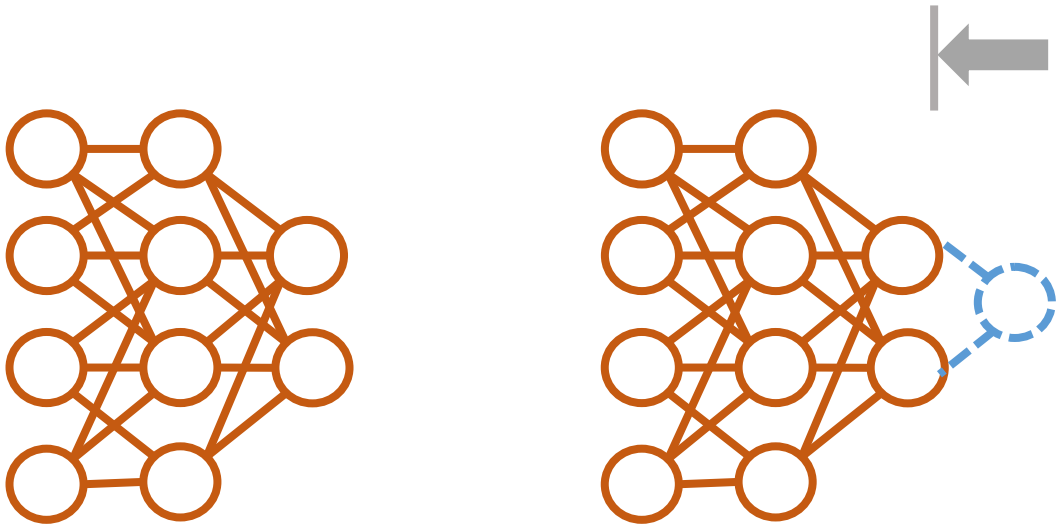


"In-distribution" training data

Method one: Fine-tuning

How to use pre-trained models?

How to leverage the diverse information contained in pre-trained models?

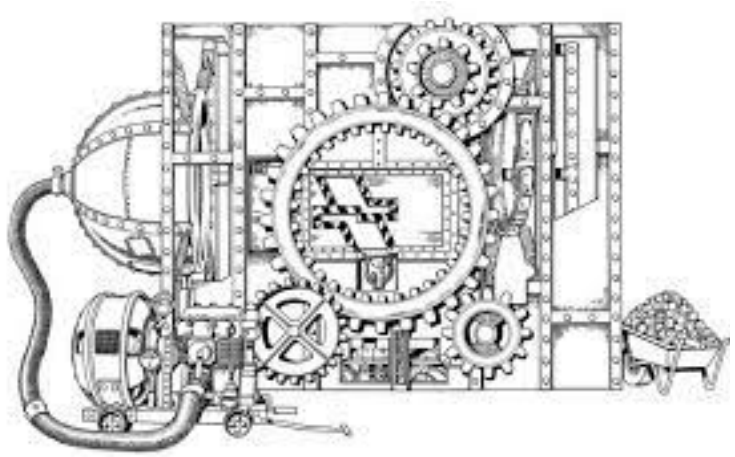


“In-distribution” training data

Method two: Linear probing

Understanding transfer learning

Several moving pieces



Model architecture

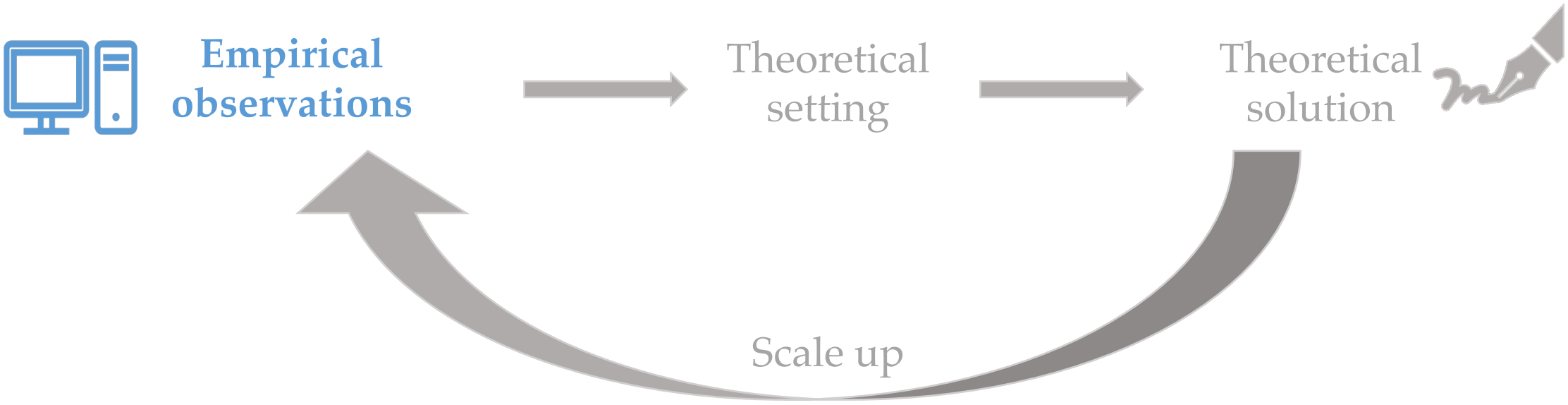
Pre-training distribution

Pre-training procedure

Adaptation distribution

Adaptation procedure

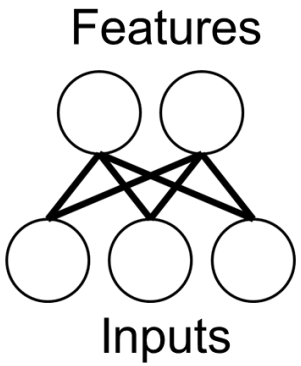
Talk outline: format



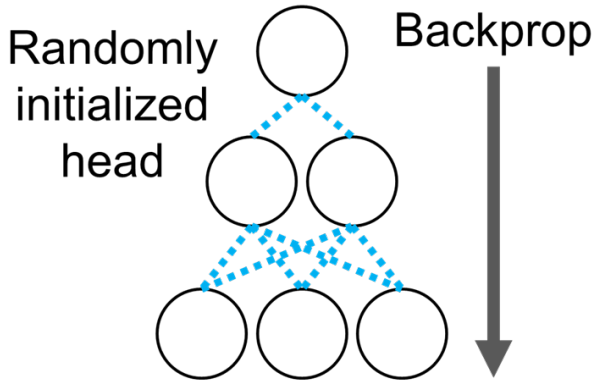
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution*. ICLR 2022.

Linear probing vs fine-tuning

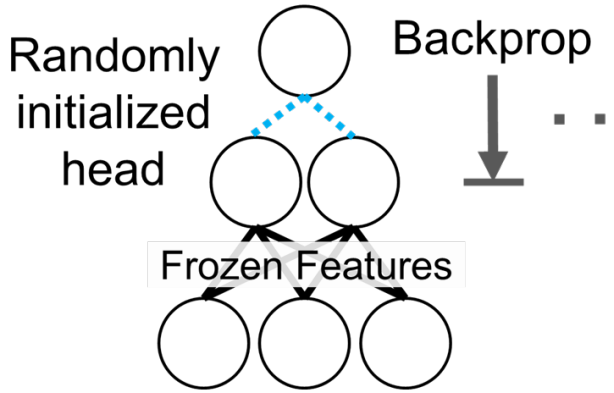
Pretraining



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

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

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?

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

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?

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

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 linear probing do better than fine-tuning 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

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

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?

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

Pop quiz: CIFAR10.1

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

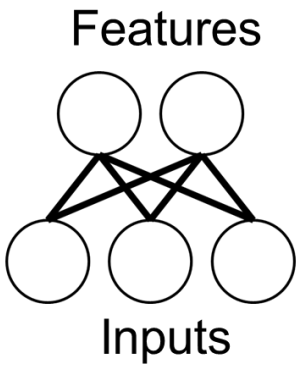
Does linear probing do better than fine-tuning OOD?

No!

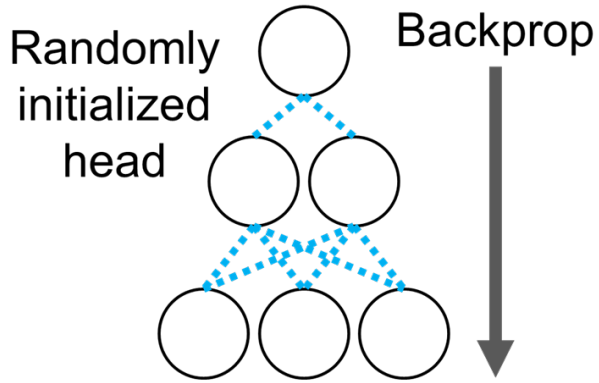
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Linear probing vs fine-tuning summary

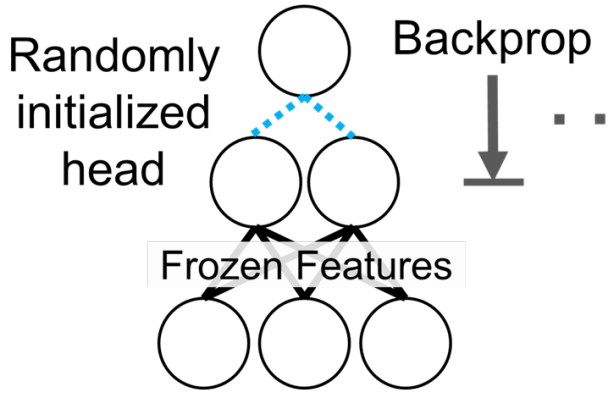
Pretraining



Fine-tuning



Linear probing



Which method does better?

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

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

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

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.

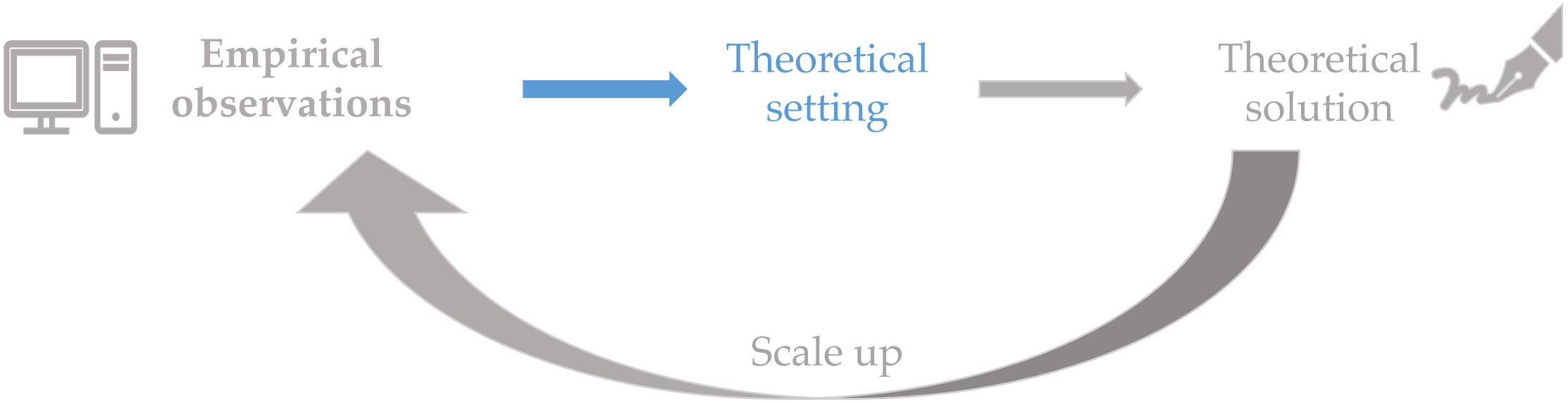
Linear probing vs fine-tuning summary

- Common wisdom is fine-tuning works better than linear probing
- Linear probing can often perform better out-of-distribution
  - Especially with **high quality** pre-trained features and **large** distribution shifts

There is probably a lot we can do to improve ***downstream*** *methods…*

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

Talk outline: format



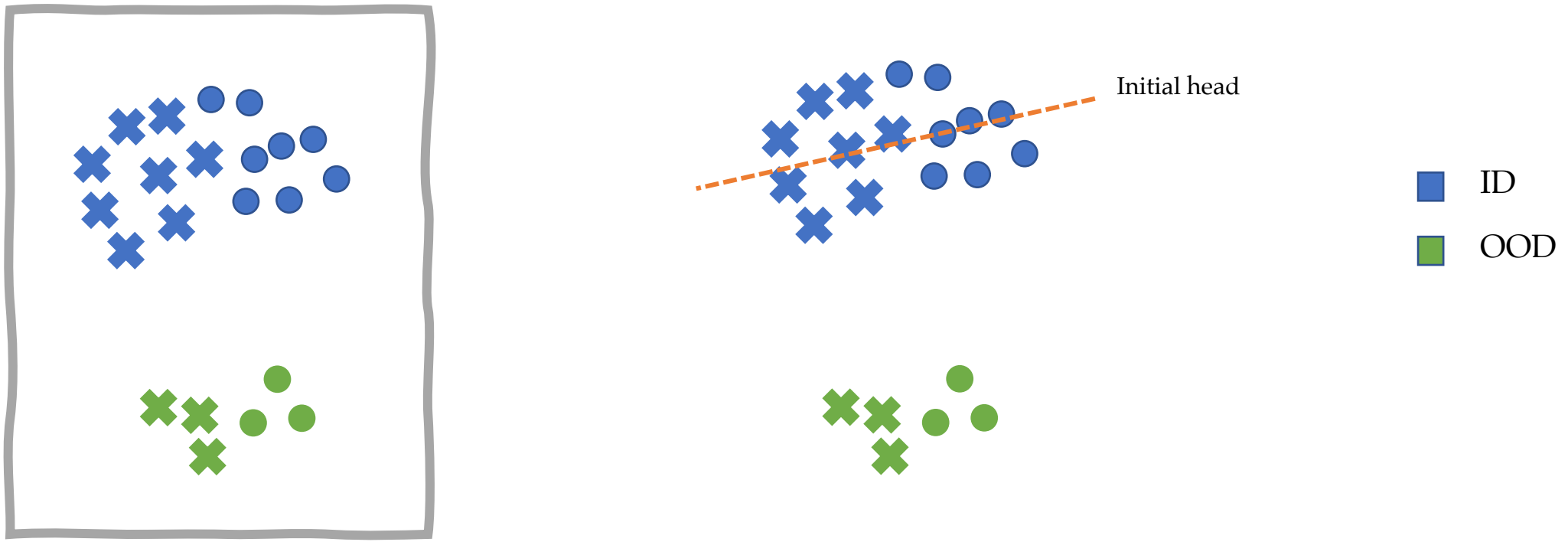
Theoretical analyses

- Prior transfer learning theory mostly looks at only linear probing which is convex (Wu et al. 2020, Tripuraneni et al. 2020, Du et al. 2020, Xie et al. 2020)
- We want to analyze the **non-convex** objective of fine-tuning
- Same objective as training from scratch but **different training dynamics** stemming from pre-trained initialization
- Cannot assume random initialization and associated simplifications

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

Intuition for theoretical result

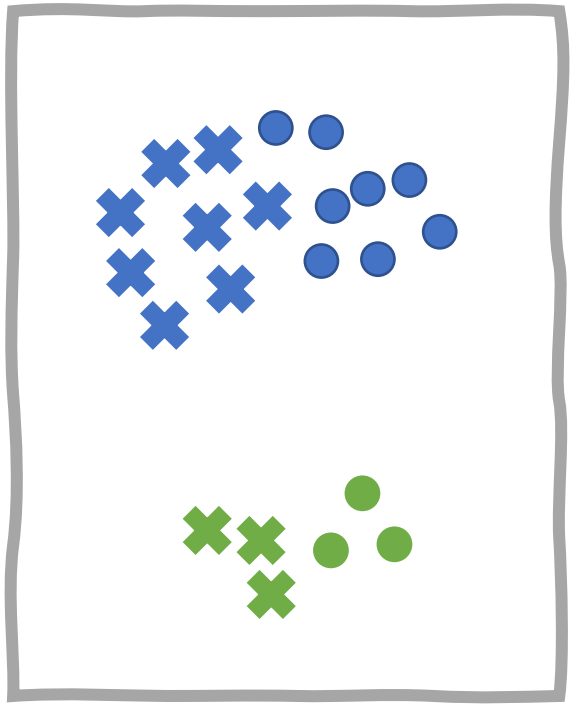
Pretrained Features



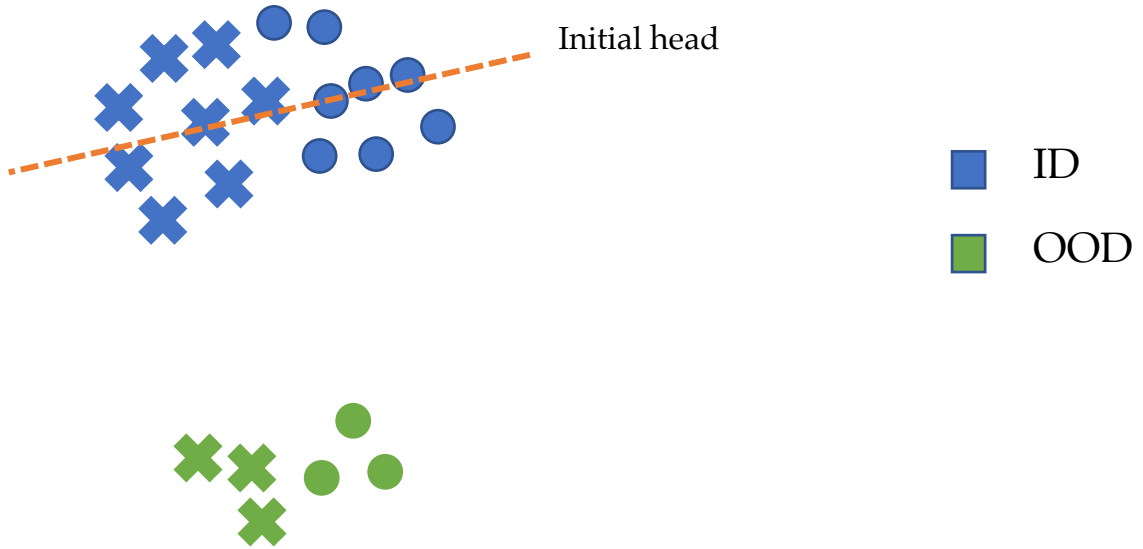
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Intuition for theoretical result

Pretrained Features



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

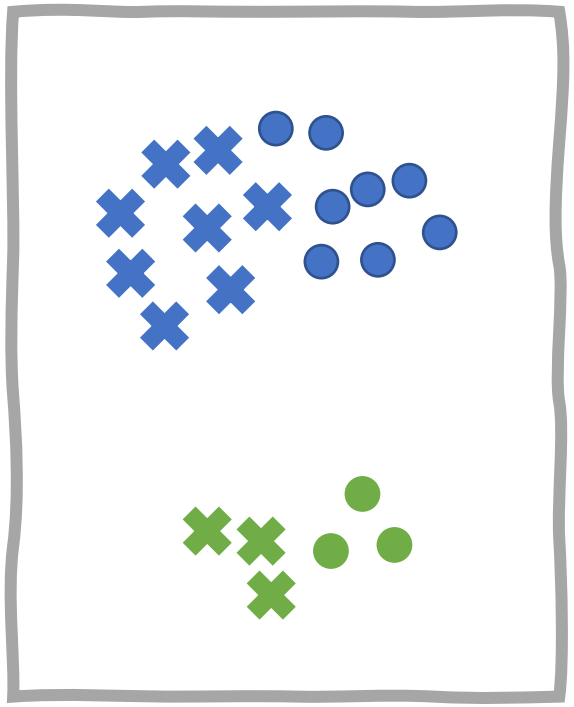


Features for OOD examples change less

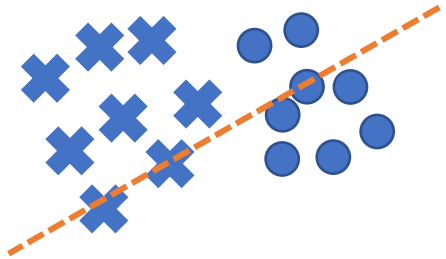
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Intuition for theoretical result

Pretrained Features



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



ID

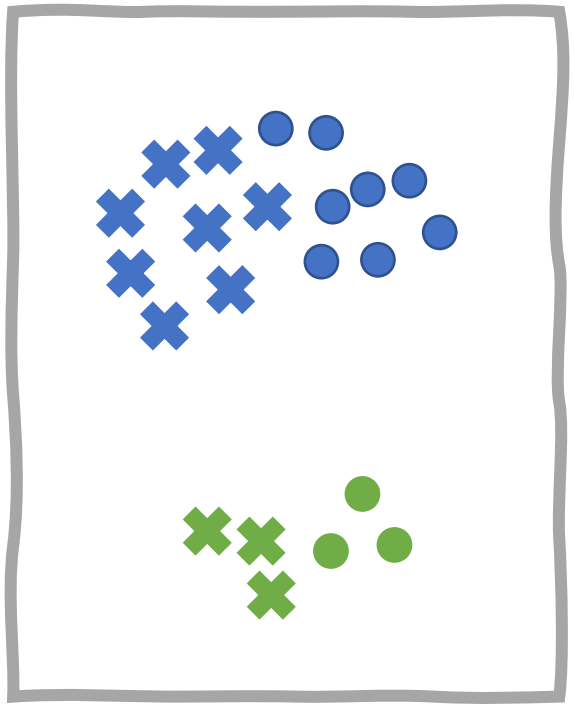
OOD

Features for OOD examples change less

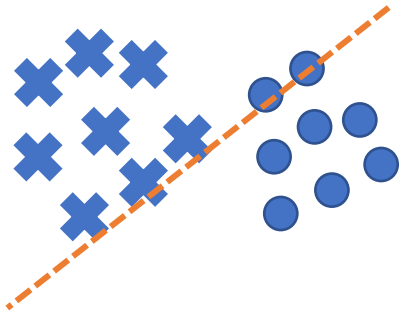
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

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

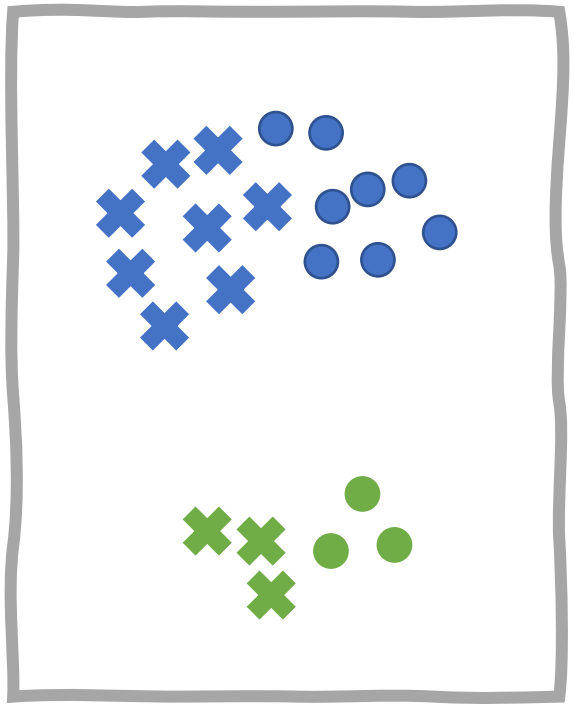
ID

OOD

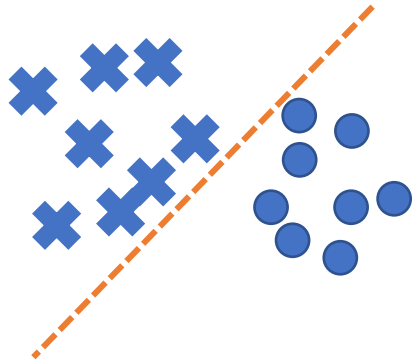
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Intuition for theoretical result

Pretrained Features



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



ID

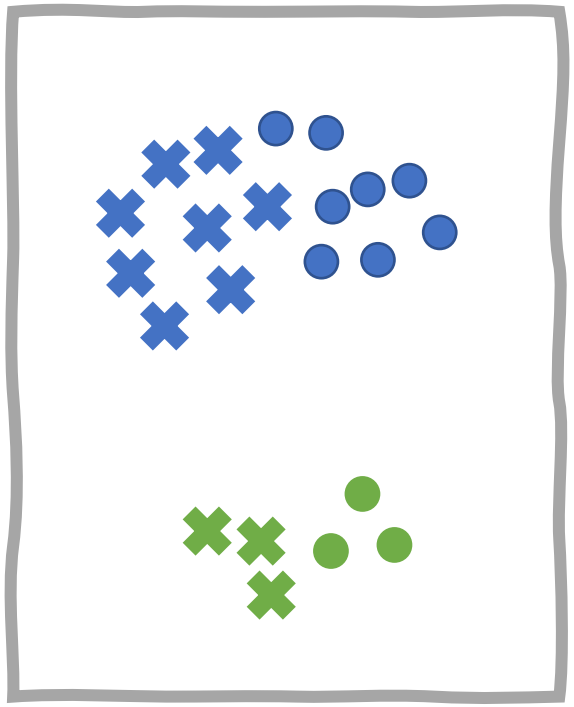
OOD

Features for OOD examples change less

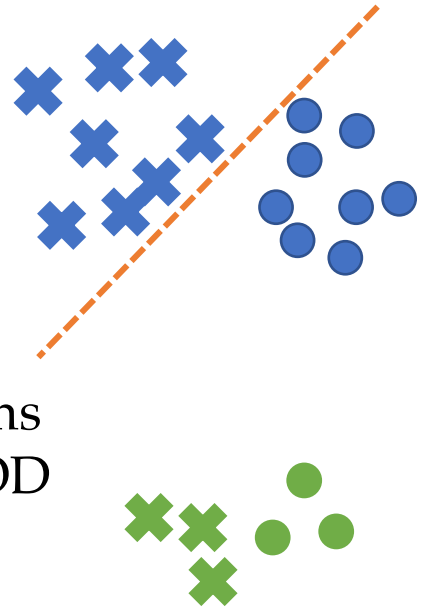
Kumar, Raghunathan, Jones, Ma, and Liang. *Fine-tuning can distort pretrained features and underperform out-of-distribution.* ICLR 2022.

Intuition for theoretical result

Pretrained Features

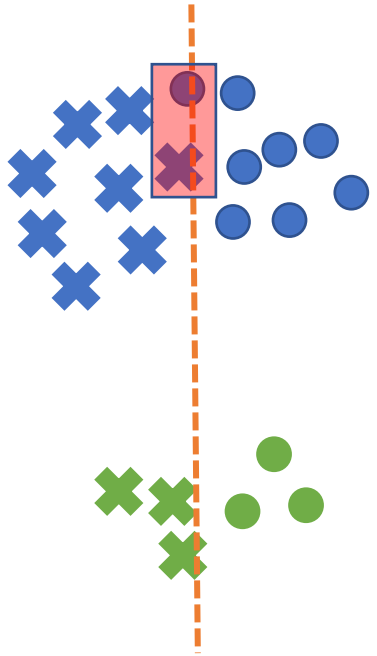


Fine-tuning: features for ID examples change in sync with the linear head



Head performs poorly on OOD examples

Linear probing: freezes pretrained features



Head is decent on OOD examples

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

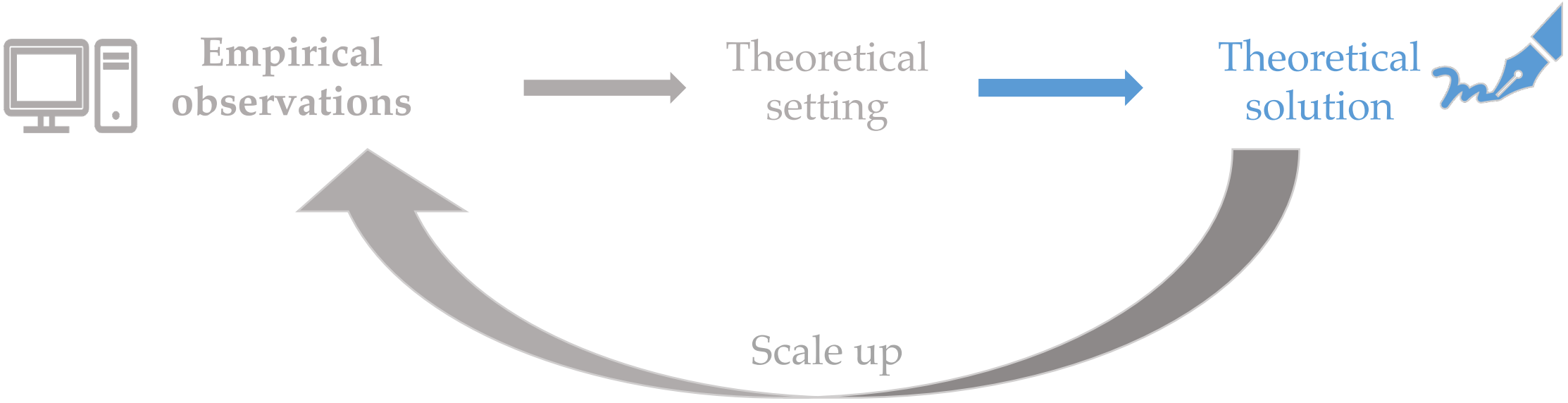
Fine-tuning can lead to feature distortion

Theorem (informal)

Under simplifying assumptions (two-layer linear networks, squared error, OOD data in orthogonal subspace to ID training data),

$$\forall t, \frac{L_{\text{ood}}\left(\theta_{\text{lp}}(t)\right)}{L_{\text{ood}}\left(\theta_{\text{ft}}(t)\right)} \xrightarrow{p} 0, \quad \text{as pretrained features} \to \text{optimal}$$

Talk outline: format



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

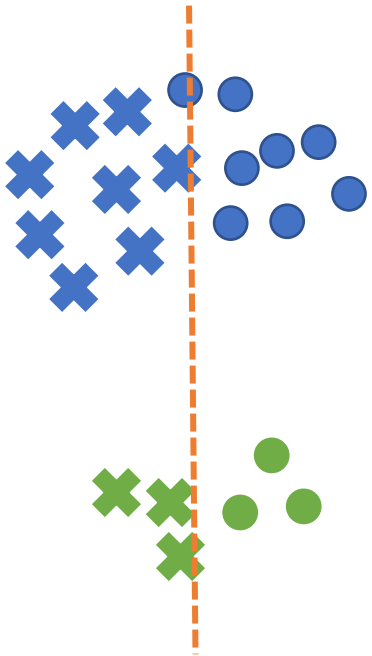
Can we refine features without distorting them too much?

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

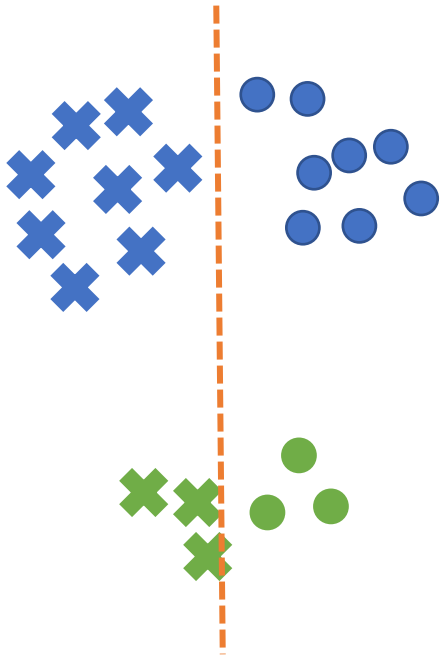
Method to achieve best of both worlds

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

Step 1: Linear probe



Step 2: Fine-tune



Method to achieve best of both worlds

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

Step 1: Linear probe

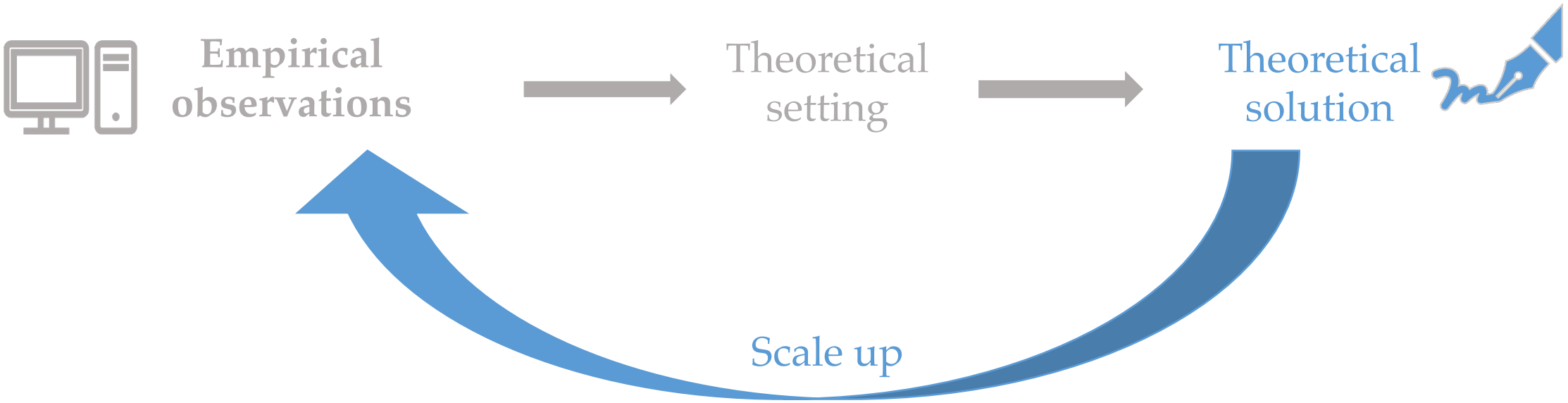
Step 2: Fine-tune

LP-FT method

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

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

Talk outline: format



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

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

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

In-Distribution Accuracies

	CIFAR-10	Ent-30	Liv-17	DomainNet	FMoW	ImageNet	Average
FT	**97.3 (0.2)**	**93.6 (0.2)**	97.1 (0.2)	84.5 (0.6)	**56.5 (0.3)**	**81.7 (-)**	85.1
LP	91.8 (0.0)	90.6 (0.2)	96.5 (0.2)	89.4 (0.1)	49.1 (0.0)	79.7 (-)	82.9
LP-FT	**97.5 (0.1)**	**93.7 (0.1)**	**97.8 (0.2)**	**91.6 (0.0)**	51.8 (0.2)	**81.7 (-)**	**85.7**

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

Out-of-Distribution Accuracies

	STL	CIFAR-10.1	Ent-30	Liv-17	DomainNet	FMoW
FT	82.4 (0.4)	92.3 (0.4)	60.7 (0.2)	77.8 (0.7)	55.5 (2.2)	32.0 (3.5)
LP	85.1 (0.2)	82.7 (0.2)	**63.2 (1.3)**	82.2 (0.2)	79.7 (0.6)	**36.6 (0.0)**
LP-FT	**90.7 (0.3)**	**93.5 (0.1)**	**62.3 (0.9)**	**82.6 (0.3)**	**80.7 (0.9)**	**36.8 (1.3)**

	ImNetV2	ImNet-R	ImNet-Sk	ImNet-A	Average
FT	**71.5 (-)**	52.4 (-)	40.5 (-)	27.8 (-)	59.3
LP	69.7 (-)	70.6 (-)	46.4 (-)	45.7 (-)	66.2
LP-FT	**71.6 (-)**	**72.9 (-)**	**48.4 (-)**	**49.1 (-)**	**68.9**

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

Experimental investigation

- ID features change much more than OOD features (l_2 distance) when doing vanilla fine-tuning

- ID features change an order of magnitude less when doing LP-FT rather than vanilla fine-tuning (same training loss)

Discussion

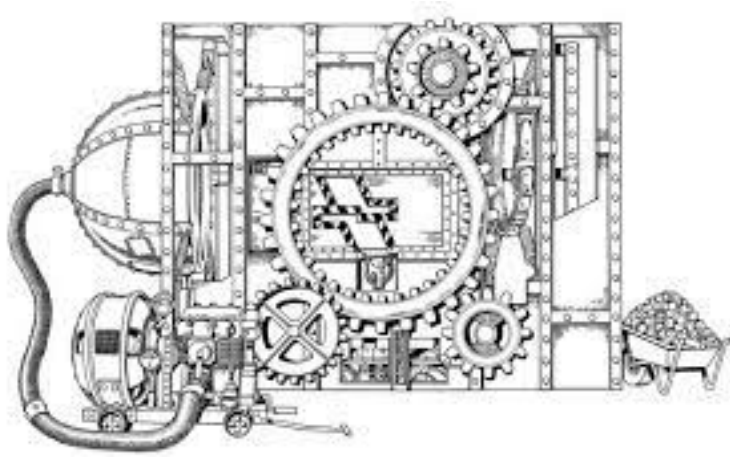
- Pretrained models give large improvements in accuracy, but how we fine-tune them is key

- LP-FT is just a starting point and one example
  - More broadly, light-weight fine-tuning (in NLP) improves robustness

- What happens with non-linearities and deeper networks?
- What's a broader class of adaptation procedures to think about?

Discussion

Several moving pieces



Model architecture

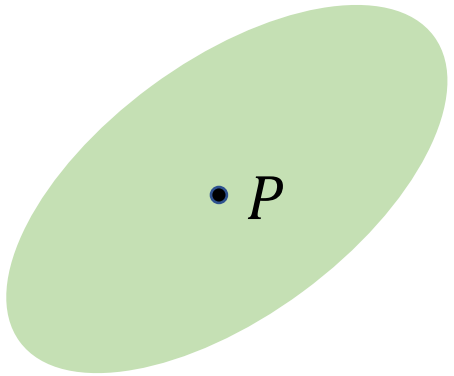
Pre-training distribution

Pre-training procedure

Adaptation distribution

Adaptation procedure

Estimating accuracy under natural shifts



You know what you get..

How to evaluate robustness to natural distribution shifts?

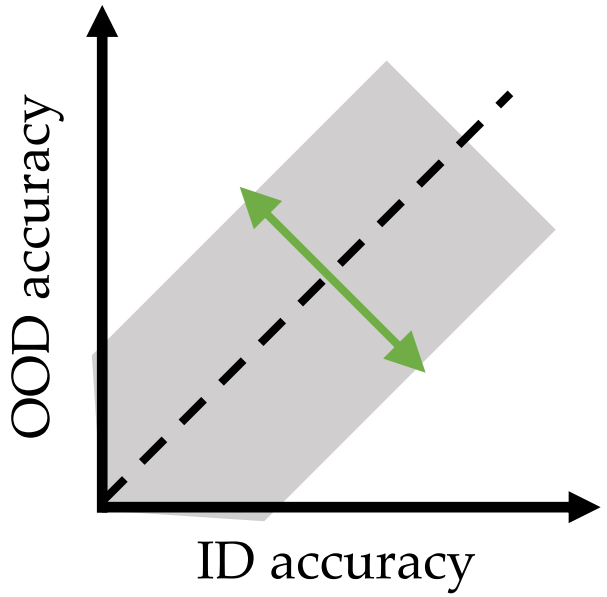
Gold standard: simply run the model on the shifted distribution and see what the performance is

This requires careful monitoring on-the-fly and constant collection of annotation data

Estimating accuracy under natural shifts

Setting: Have labeled training data in-domain (ID) and unlabeled data out-of-domain (OOD)
Goal: predict OOD accuracy of a model trained on ID labeled data

Classic idea: Bound OOD performance via domain discrepancy

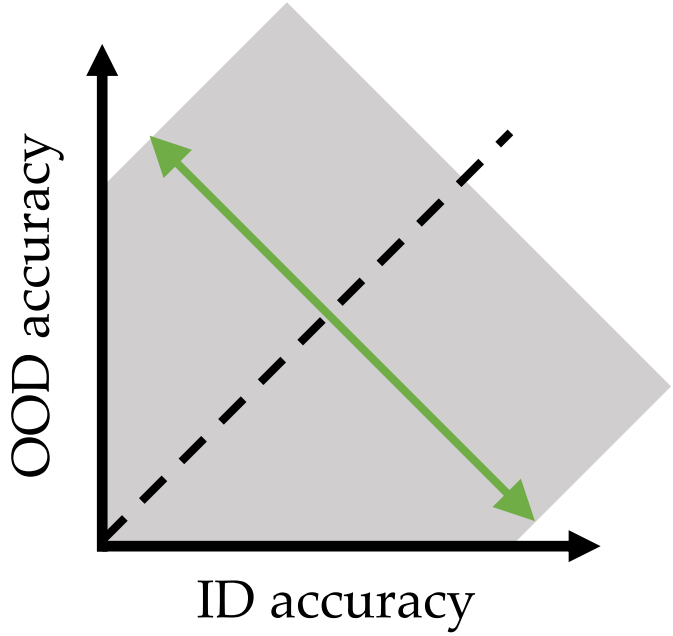


Bounds are vacuous if domain discrepancy is large

Estimating accuracy under natural shifts

Setting: Have labeled training data in-domain (ID) and unlabeled data out-of-domain (OOD)
Goal: predict OOD accuracy of a model trained on ID labeled data

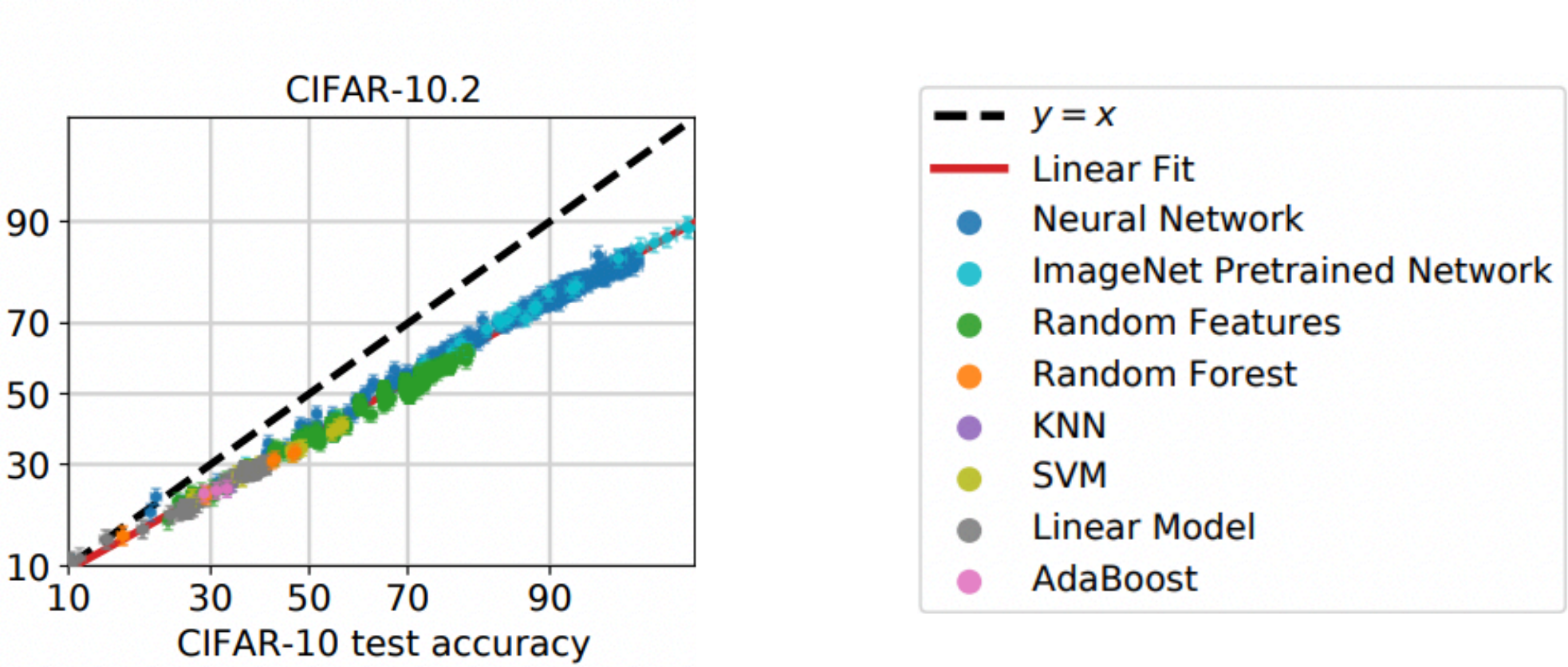
Classic idea: Bound OOD performance via domain discrepancy



Bounds are vacuous if domain discrepancy is large

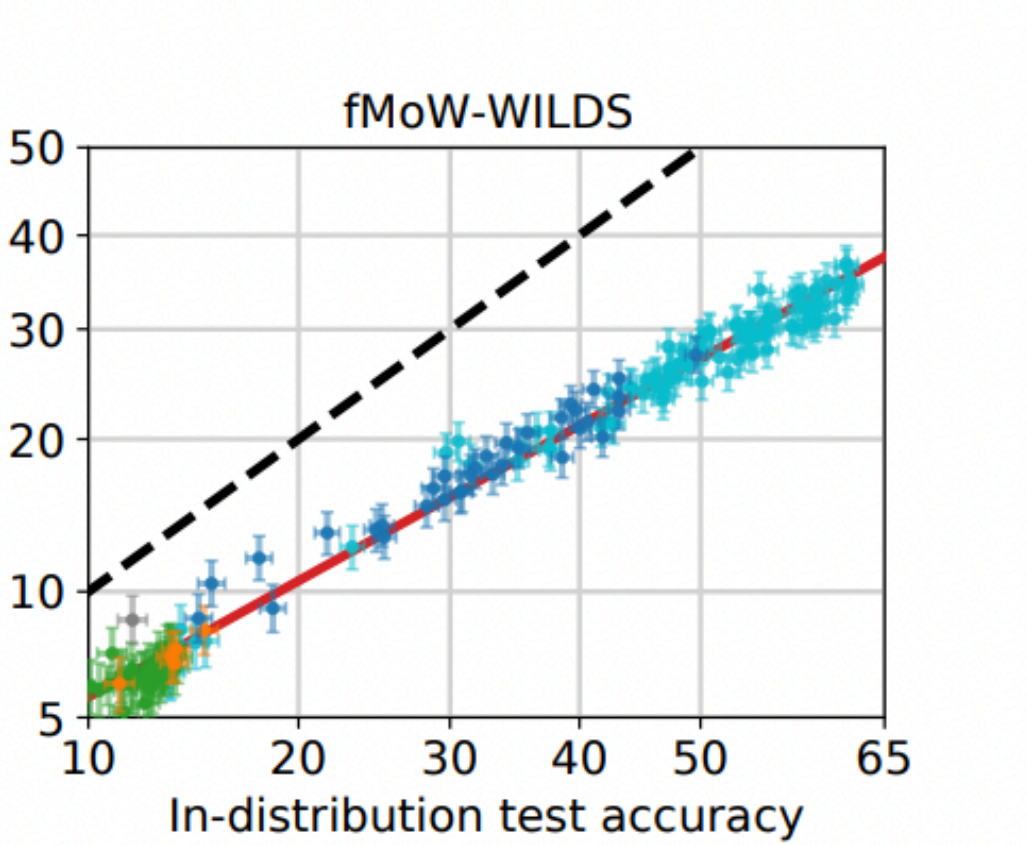
Miller*, Taori*, Raghunathan, Sagawa, Koh, Shankar, Liang, Carmon, Schmidt. *Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization*. ICML 2021.

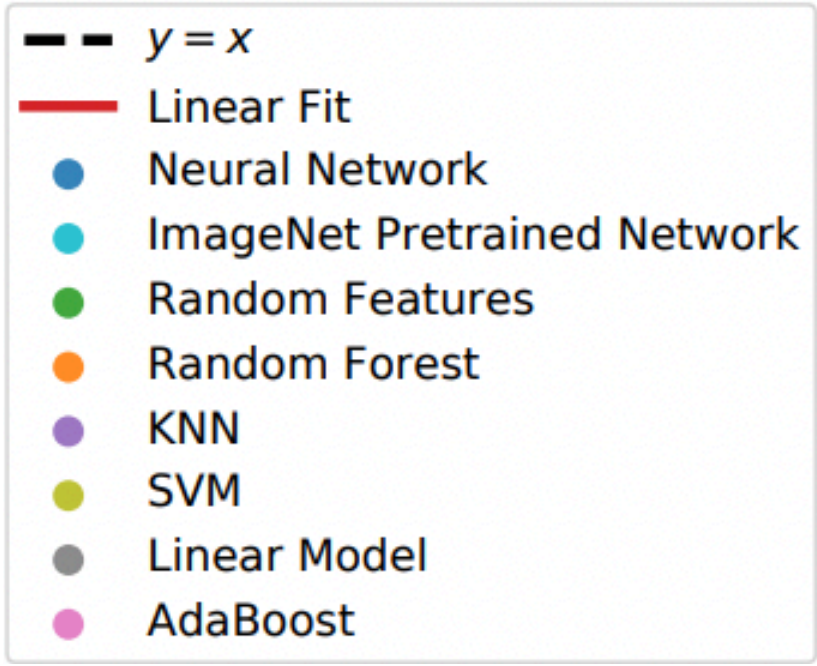
What happens in practice?



Miller*, Taori*, Raghunathan, Sagawa, Koh, Shankar, Liang, Carmon, Schmidt. *Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization*. ICML 2021.

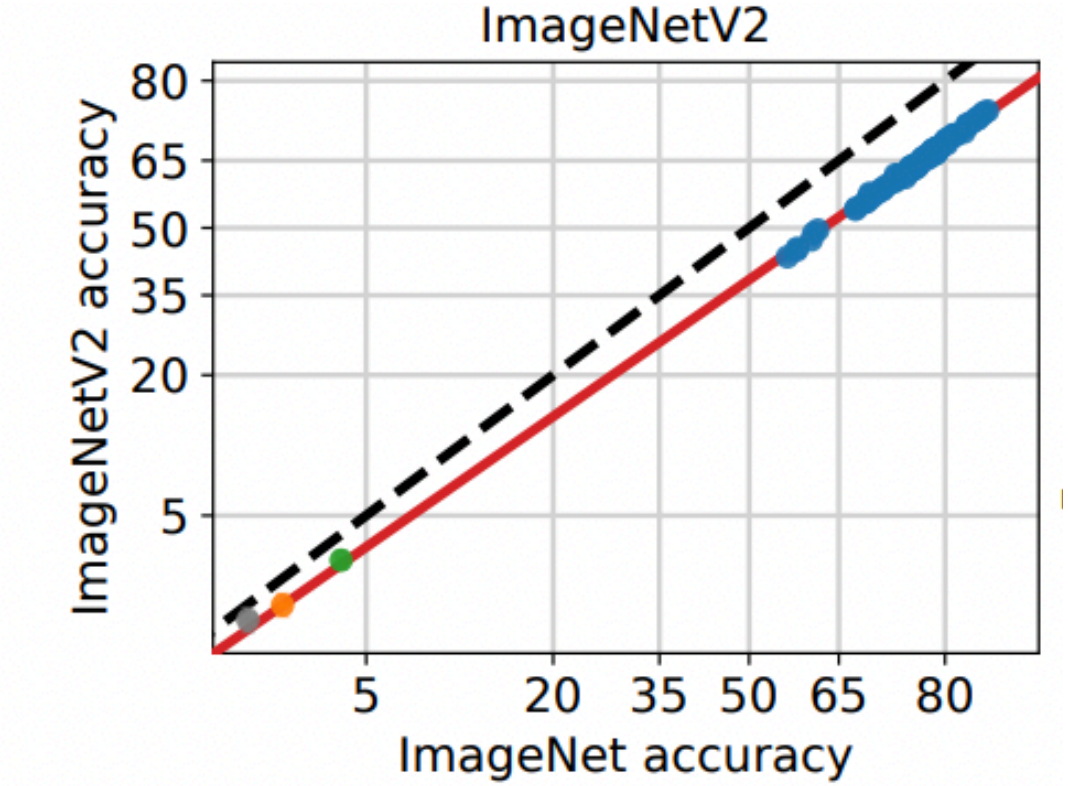
What happens in practice?

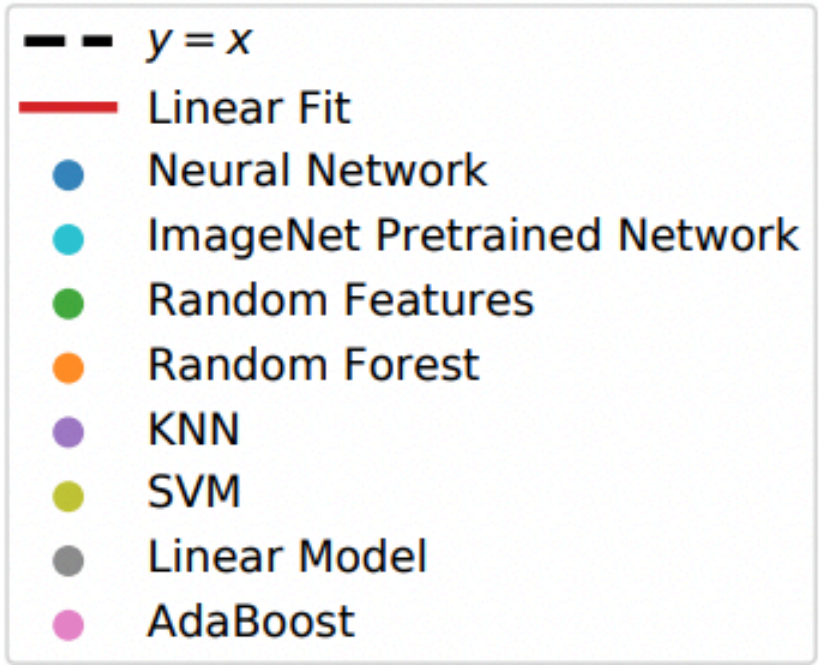




Miller*, Taori*, Raghunathan, Sagawa, Koh, Shankar, Liang, Carmon, Schmidt. *Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization*. ICML 2021.

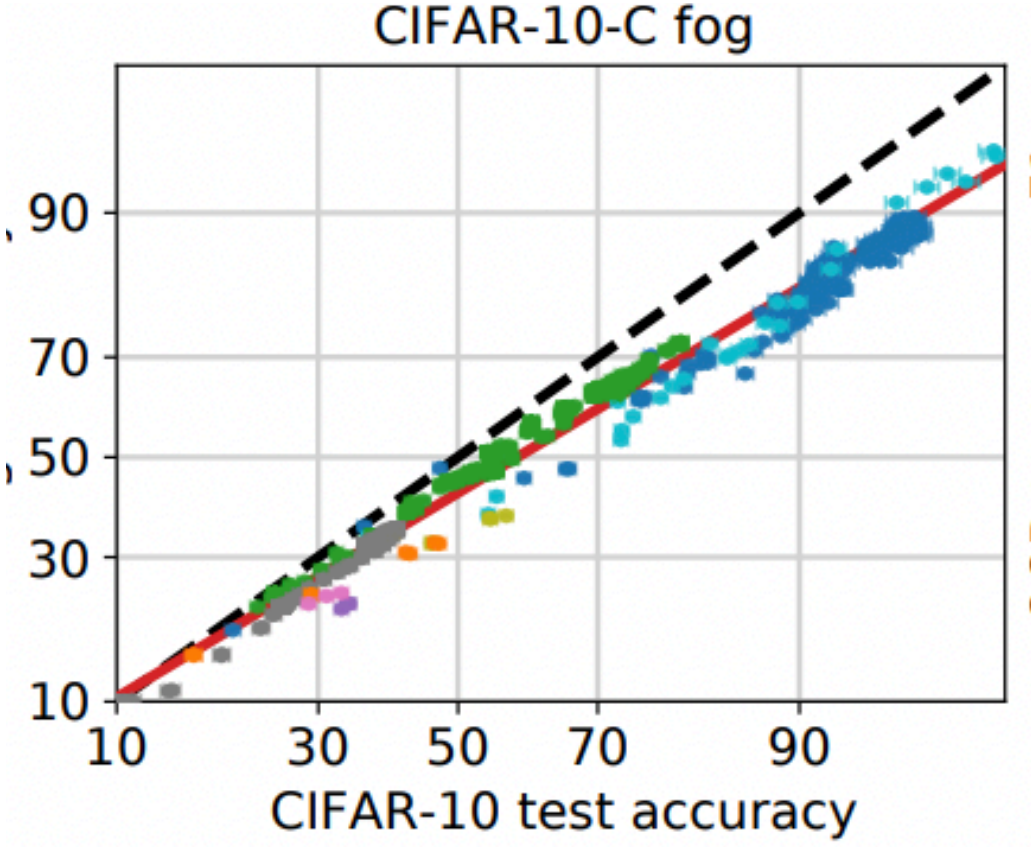
What happens in practice?

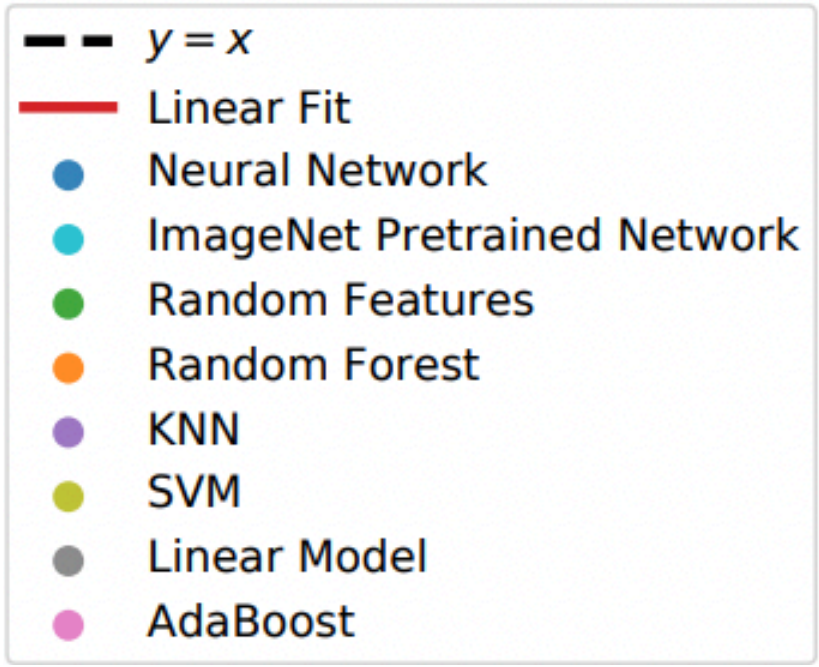




Miller*, Taori*, Raghunathan, Sagawa, Koh, Shankar, Liang, Carmon, Schmidt. *Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization*. ICML 2021.

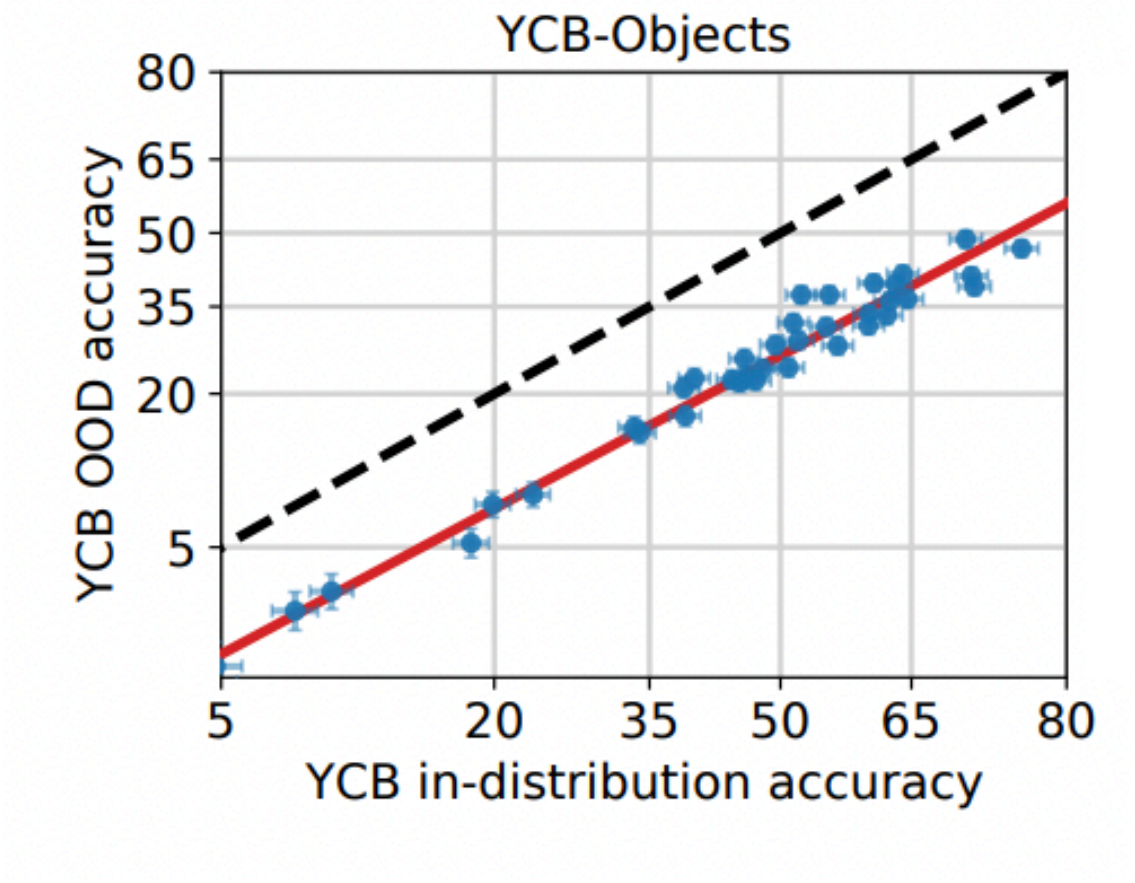
What happens in practice?

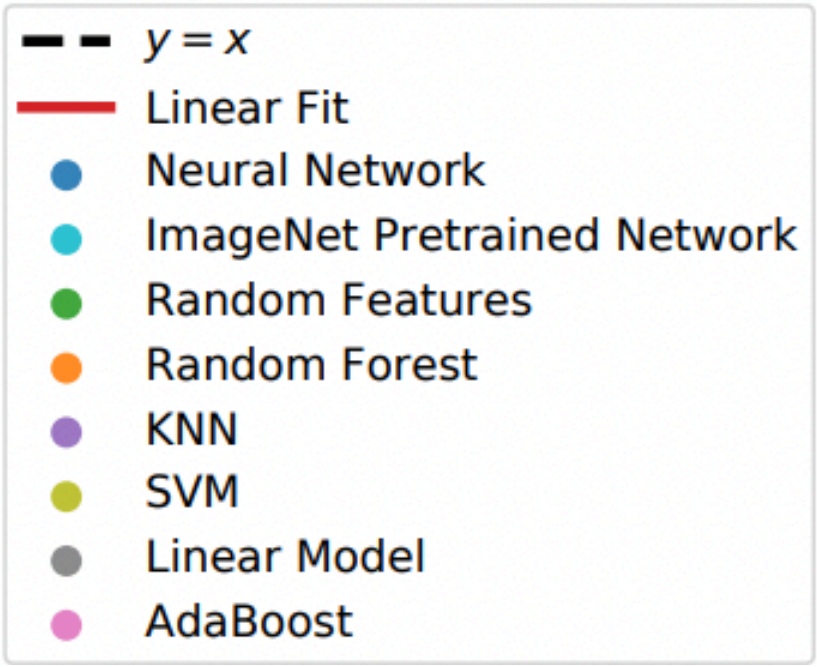




Miller*, Taori*, Raghunathan, Sagawa, Koh, Shankar, Liang, Carmon, Schmidt. *Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization*. ICML 2021.

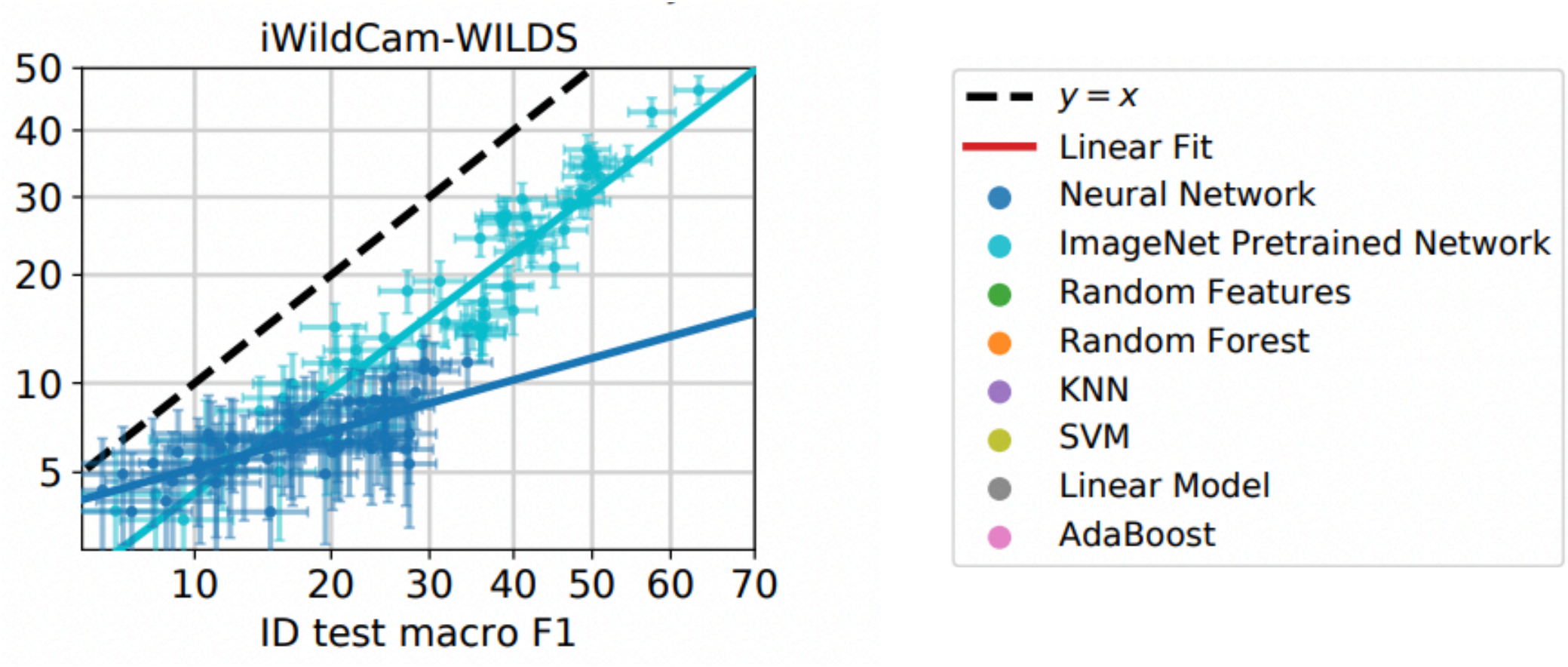
What happens in practice?





Miller*, Taori*, Raghunathan, Sagawa, Koh, Shankar, Liang, Carmon, Schmidt. *Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization*. ICML 2021.

What happens in practice?



Miller*, Taori*, Raghunathan, Sagawa, Koh, Shankar, Liang, Carmon, Schmidt. *Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization*. ICML 2021.

What happens in practice?

On most datasets, OOD accuracy is almost perfectly linearly correlated with ID accuracy (after a probit transform)

There is some structure that we could potentially exploit in assessing performance under distribution shifts…

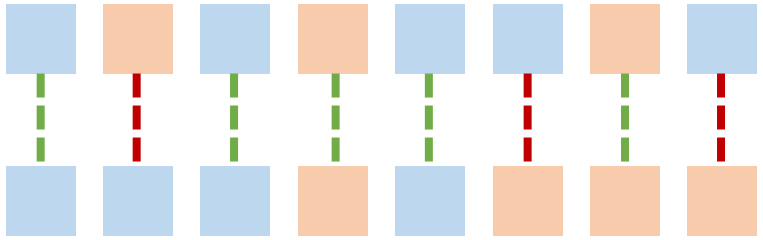
We are not done yet… the slopes and bias of the line vary depending on the shift, and require labeled OOD data to estimate

Model agreement

Accuracy

Neural net predictions

Ground truth



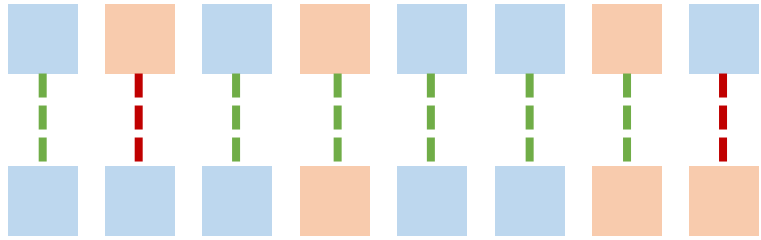
62.5%

Requires ground truth labels

Agreement

Neural net M1 predictions

Neural net M2 predictions



75%

Does NOT require ground truth labels

Model agreement vs accuracy

Take two models with accuracy $1 - \epsilon$:

- their disagreement can be anywhere between $1 - \epsilon/2$ and $1 - 2\epsilon$
- For networks with trained on the same dataset with same hyperparameters but different random seed, disagreement is very close to $\mathbf{1 - \epsilon}$

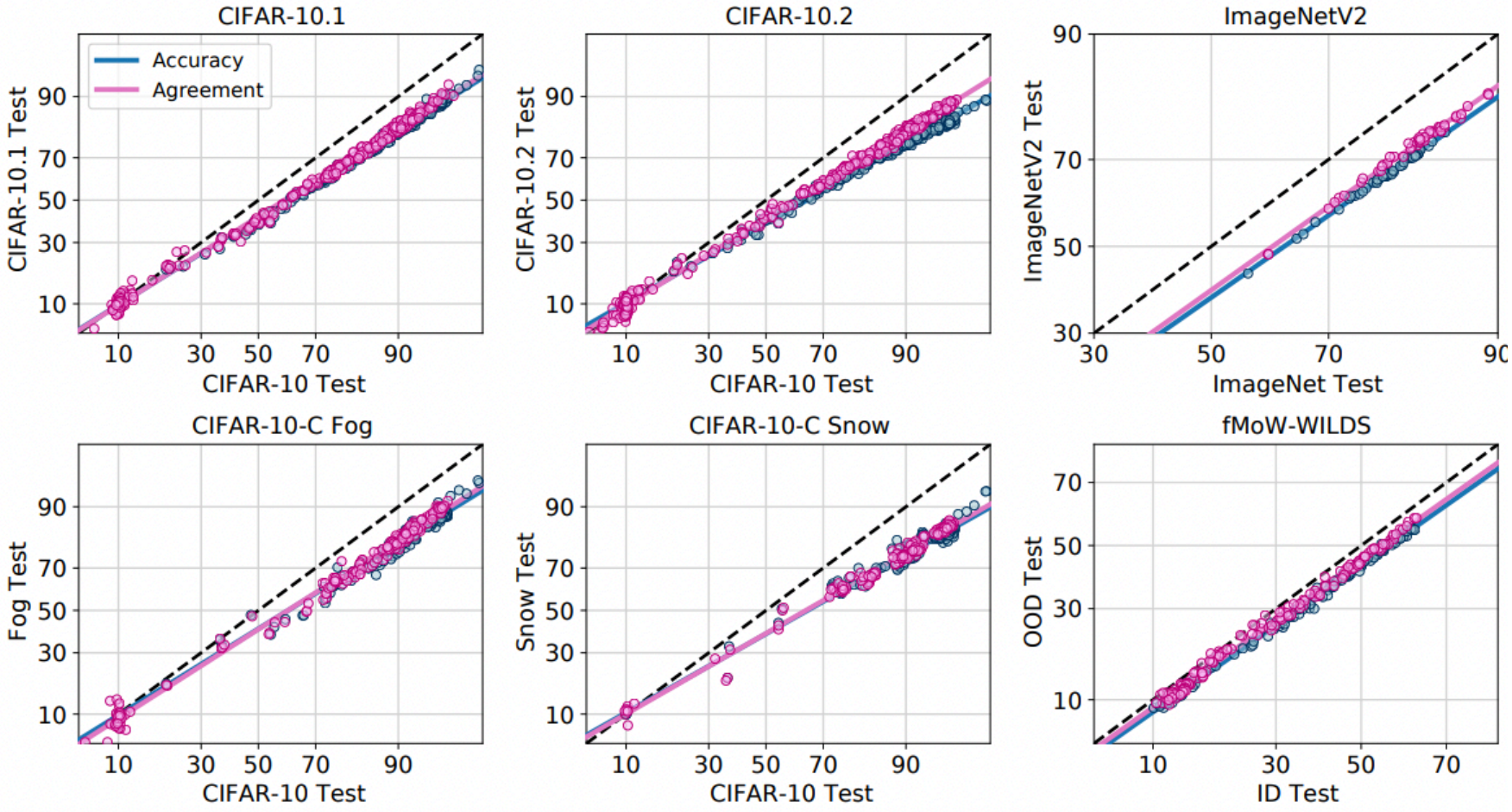
In-distribution disagreement approximates in-distribution accuracy

What happens out-of-distribution?

Jiang et al. 2022 Nakkiran & Bansal. 2020

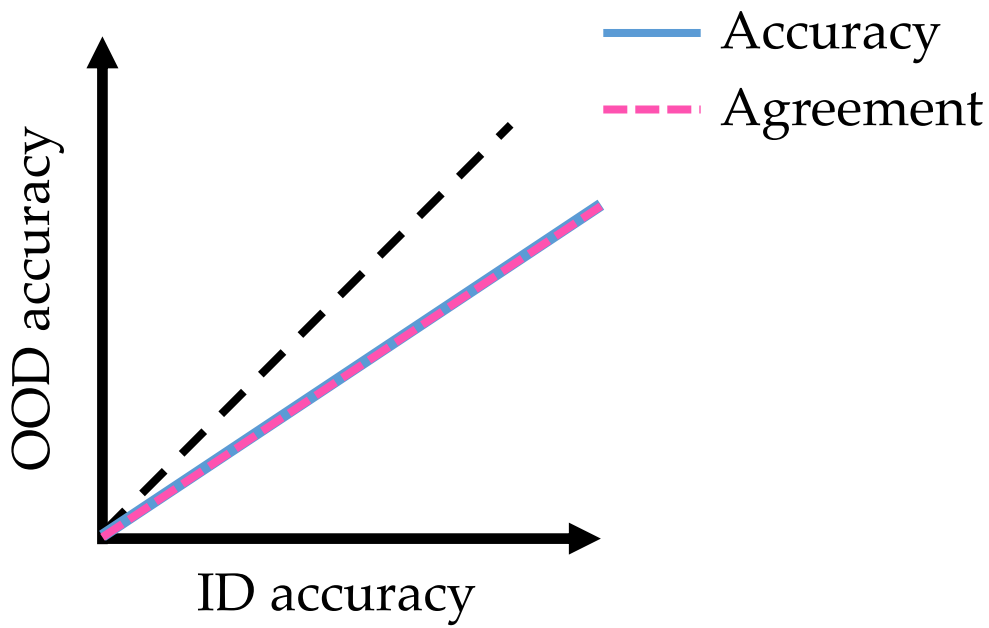
Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

Agreement-on-the-line



Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

Agreement-on-the-line

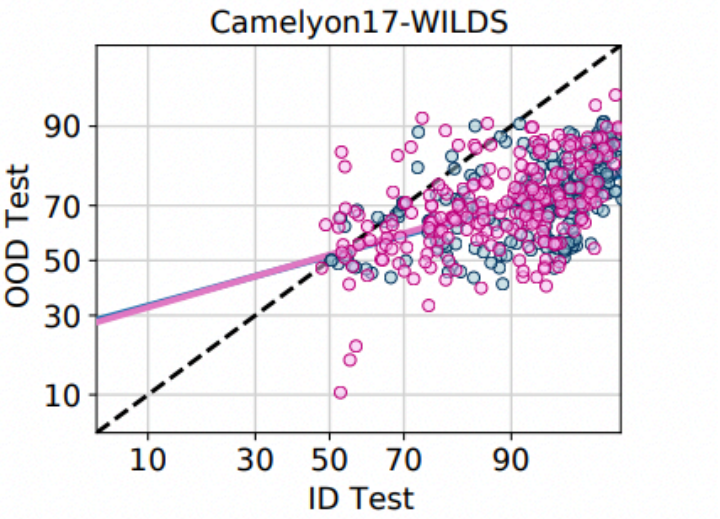


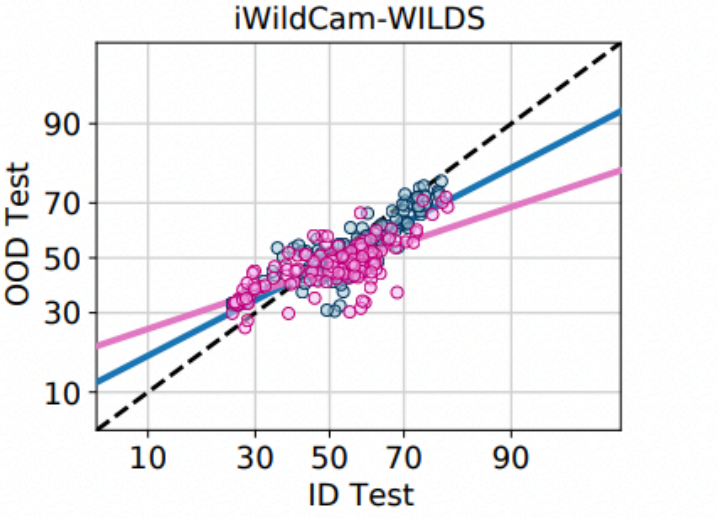
ID vs OOD agreement is on a line *if* ID vs OOD accuracy is on a line

Accuracy and agreement lines share the same slope and bias

Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

Agreement-on-the-line



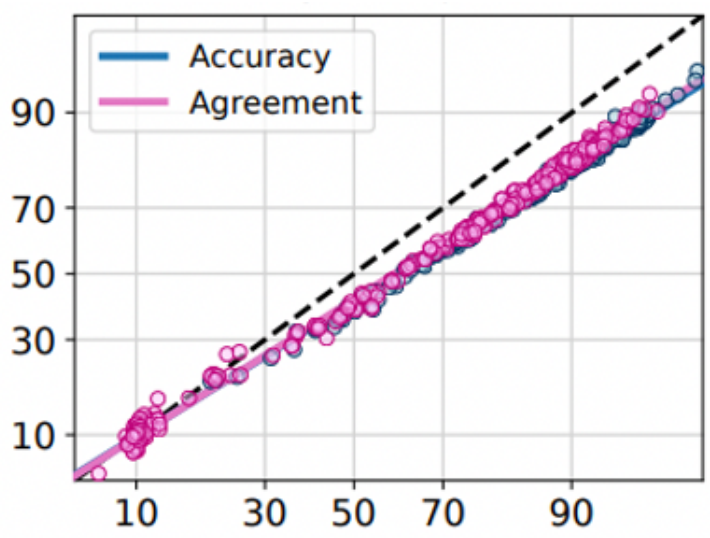


ID vs OOD accuracy is *not* on a line *if*
ID vs OOD agreement is *not* on a line

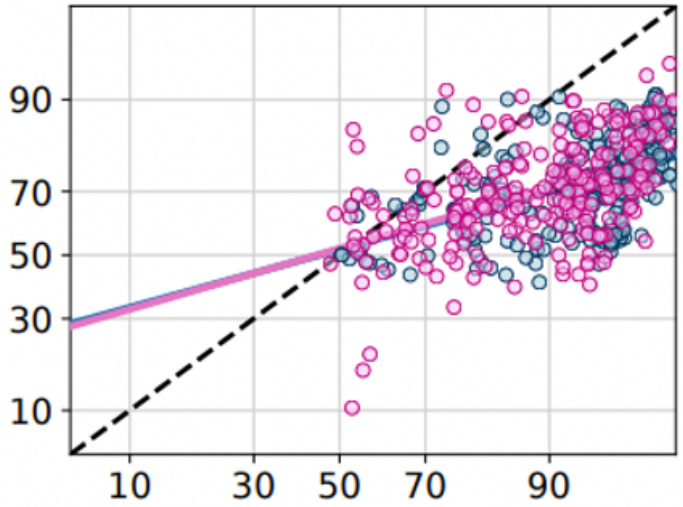
Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

Predicting OOD accuracy

Step one: Train several neural networks with different architectures and random seeds



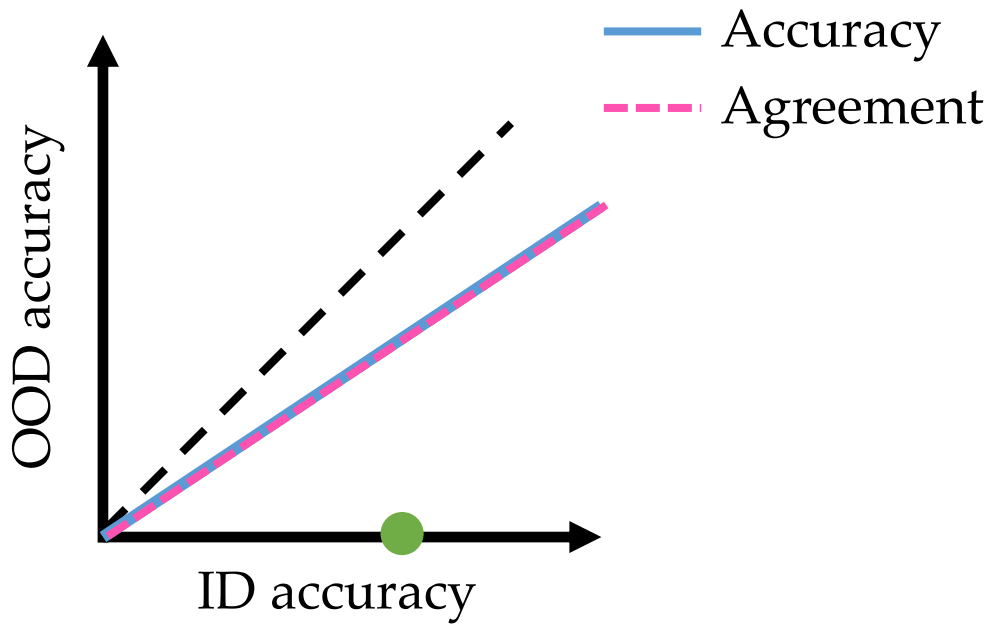
Step two: Compute and plot ID vs OOD agreement



Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

Predicting OOD accuracy

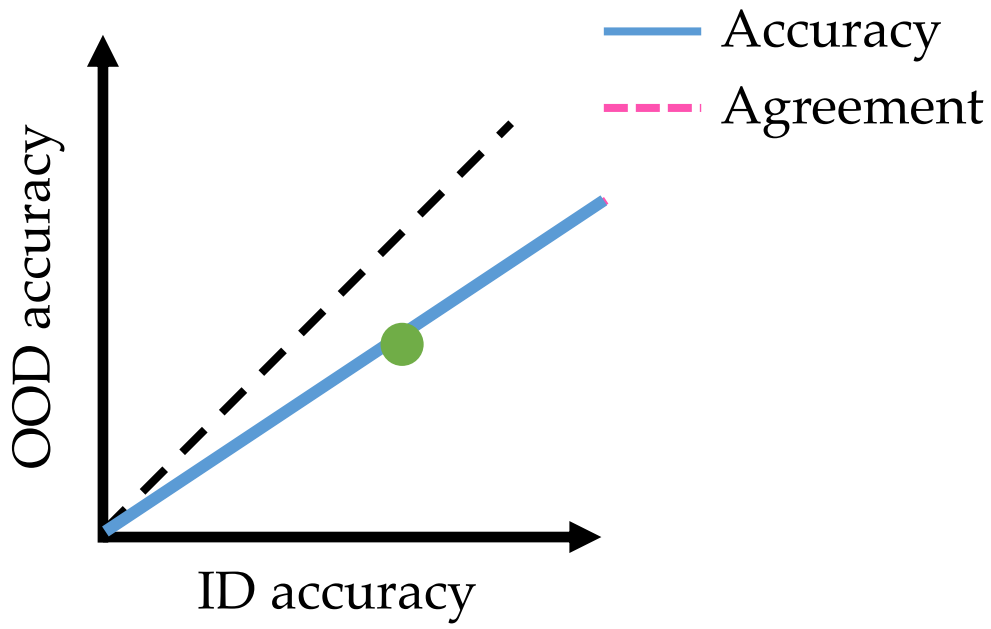
Step three: Use slope and bias from agreement correlation to estimate OOD accuracy



Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

Predicting OOD accuracy

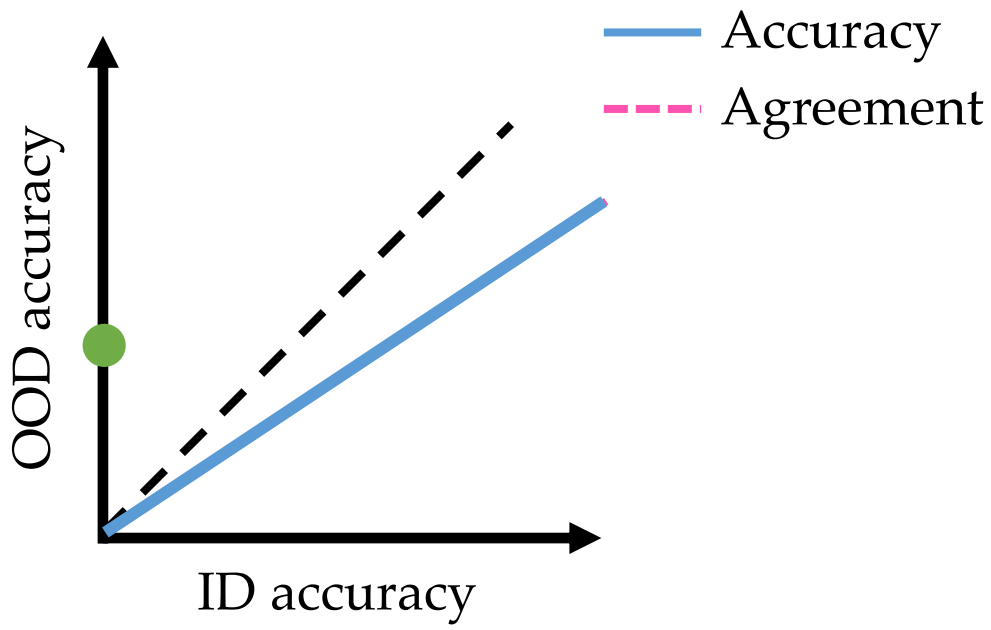
Step three: Use slope and bias from agreement correlation to estimate OOD accuracy



Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

Predicting OOD accuracy

Step three: Use slope and bias from agreement correlation to estimate OOD accuracy



Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

How well does this work?

Mean absolute estimation error

Dataset	ALine-D	ALine-S	ATC [Garg '22]	AC [Hendrycks '17]	DOC [Guillory '17]	Agreement
CIFAR-10.1	**1.11**	1.17	1.21	4.51	3.87	5.98
CIFAR-10.2	**3.93**	**3.93**	4.35	8.23	7.64	5.42
ImageNetV2	2.06	2.08	**1.12**	66.2	11.50	6.70
CIFAR-10C-Fog	**1.45**	1.75	1.78	4.47	3.93	3.47
CIFAR-10C-Snow	**1.32**	1.97	**1.31**	5.94	5.49	2.57
CIFAR10C-Saturate	**0.41**	0.77	0.69	2.03	1.51	4.14
fMoW-wilds	**1.30**	1.44	1.53	2.89	2.60	8.99
RxRx1-wilds	**0.27**	0.52	2.97	2.46	0.65	8.67
Camelyon17-wilds	**5.47**	8.31	11.93	13.30	13.57	6.79
iWildCam-wilds	4.95	6.01	12.12	**4.46**	5.02	7.53

Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

How well does this work?

- When agreement is on a line, this method typically outperforms previous methods and provides reliable OOD error estimates

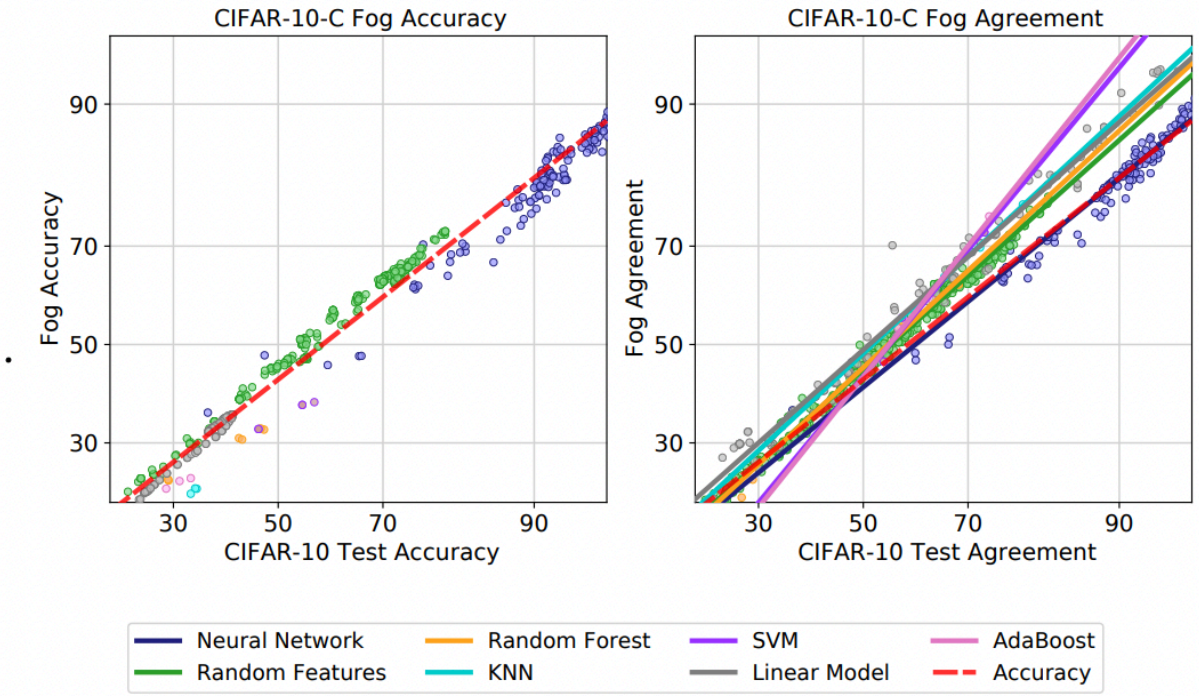
- When agreement is not on a line, this method still outperforms alternatives
  - However, OOD error estimates are less reliable

A simple and reliable method to determine OOD error

Baek, Jiang, Raghunathan, Kolter. *Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift*. Under submission.

Discussion

- Agreement-on-the-line provides a simple and reliable method to determine OOD performance
- Why does this work?



Phenomenon is specific to neural networks!

Discussion

- Agreement-on-the-line provides a simple and reliable method to determine OOD performance
- Why does this work?

Worst-case shifts?

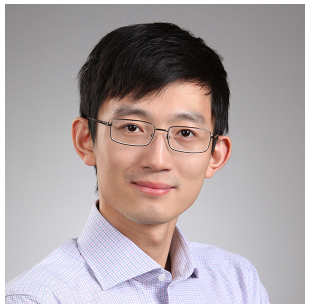
- ID vs OOD agreement lies on a line, but accuracy does not, or they have a different slope

Parting comments

- Robustness in ML is exciting and largely unsolved
- We need to understand the foundations of robustness to mathematically well-defined shifts
- Natural shifts may not lend themselves to a simple mathematical definition but still seem to encode some useful structure

Ananya Kumar

Robbie Jones



Tengyu Ma

Percy Liang

Zico Kolter

Yiding Jiang

Christina Baek

Shiori Sagawa

Open Philanthropy

John Miller

Rohan Taori

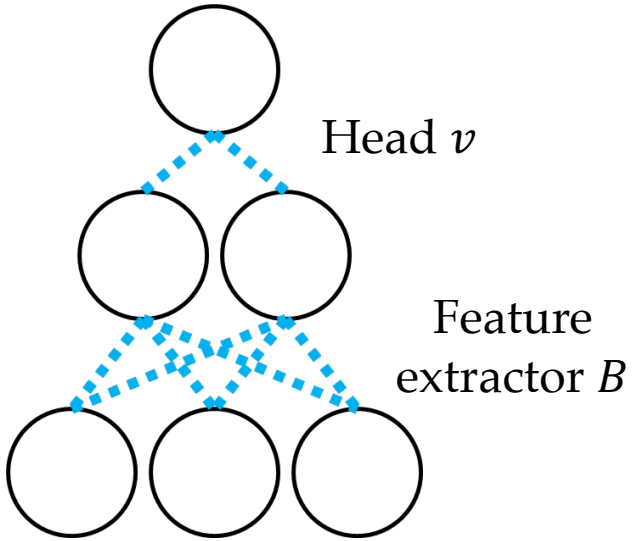
Pang Wei Koh

Yair Carmon

Ludwig Schmidt

Theoretical setting

- Two-layer linear networks
  - High dimensional input: x in R^d
  - Lower dimensional features: $B_* x$ in R^k
  - Ground truth outputs: $y = v_*^{\mathrm{T}} B_* x$



- Squared loss: $L(v, B) = \left(y - v^{\mathrm{T}} B x\right)^2$

- FT updates v and B, LP only updates v
  - v is randomly initialized in both cases

Theoretical analysis

Models: two-layer linear networks, squared loss

Assumptions:

- B_0 close to B_*, so $\min_U ||B_* - UB_0||_2 \leq \epsilon$ for rotation matrices U
- Let B_0, B_* have orthonormal rows
- Infinitesimally small learning rate => updates are gradient flow

Theoretical analysis

Models: two-layer linear networks, squared loss

Assumptions: orthonormal matrices, B_0 close to B_*, gradient flow

Data distribution (train):

- S is the span of training samples $x_1, \ldots, x_n \in R^d$
- Overparameterized setting: $1 \leq \dim(S) \leq d - k$

Data distribution (test):

- P_{ood} has invertible covariance matrix Σ
- $L_{\text{ood}} = E_{x \sim P_{\text{ood}}}[(y - v^\top B x)]^2$

Theoretical analysis

Models: two-layer linear networks, squared loss

Assumptions: orthonormal matrices, B_0 close to B_*, gradient flow

Data distribution (train):

- S is the span of tra

OOD includes directions not seen in training data

- Overparameterized setting: $1 \leq \dim(S) \leq d - k$

Data distribution (test):

- P_{ood} has invertible covariance matrix Σ
- $L_{\text{ood}} = E_{x \sim P_{\text{ood}}}[(y - v^\top B x)]^2$

Lower bound on FT error

Theorem 3.3 (FT error, simplified & informal)

$$L_{\text{ood}}(v_{\text{ft}}(t), B_{\text{ft}}(t)) \geq O\left(\frac{\alpha}{k}\varphi\right) \textit{ for small } \epsilon$$

Lower bound on FT error

Theorem 3.3 (FT error, simplified & informal)

$$L_{\text{ood}}(v_{\text{ft}}(t), B_{\text{ft}}(t)) \geq O\left(\frac{\alpha}{k}\varphi\right) \text{ for small } \epsilon$$

α is the principal angle between $S^{\perp}$ and the rowspace(B_0)

Principal angle measures alignment of subspaces, generalizes cosine angle between two vectors

The more B_0 interacts with the missing directions of training data, the larger the OOD error can get

Lower bound on FT error

Theorem 3.3 (FT error, simplified & informal)

$$L_{\text{ood}}(v_{\text{ft}}(t), B_{\text{ft}}(t)) \geq O\left(\frac{\alpha}{k}\boldsymbol{\varphi}\right) \textit{ for small } \epsilon$$

$\varphi^2 = |(v_0^\top v_*)^2 - (v_*^\top v_*)^2|$ is the initial head alignment error

The farther the randomly initialized v_0 is from v_*, the worse the OOD error can get

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

LP vs FT

Theorem 3.5 (LP vs FT OOD error, informal)

$$\forall t, \frac{L_{\text{ood}}\left(v_{\text{lp}}^{\infty}, B_0\right)}{L_{\text{ood}}\left(v_{\text{ft}}(t)\, B_{\text{ft}}(t)\right)} \xrightarrow{p} 0, \quad \text{as } B_0 \to B_* \text{ (up to rotations)},$$

when principal angles between S and R_* and between $S^{\perp}$ and R_* is non-zero

ID error: fine-tuning is always better than linear probing

Theoretical tool: **leverage invariants** that hold throughout process of fine-tuning