Does Distribution Shift Matter In the Era of Pre-trained Language Models?

Robin Jia
University of Southern California
Simons Workshop on Domain Adaptation and Related Areas
November 12, 2024

The Hunt for Meaningful Failures

- LLMs work surprisingly well, but they still fail in many meaningful situations
- **How can we explain and anticipate these failures?**

ASHLEY BELANGER, ARS TECHNICA BUSINESS FEB 17, 2024 12:12 PM

Air Canada Has to Honor a Refund Policy Its Chatbot Made Up

REUTERS

New York lawyers sanctioned for using fake ChatGPT cases in legal brief

By **Sara Merken**

June 26, 2023 1:28 AM PDT · Updated 9 months ago

Aa

Do Users Write More Insecure Code with AI Assistants?

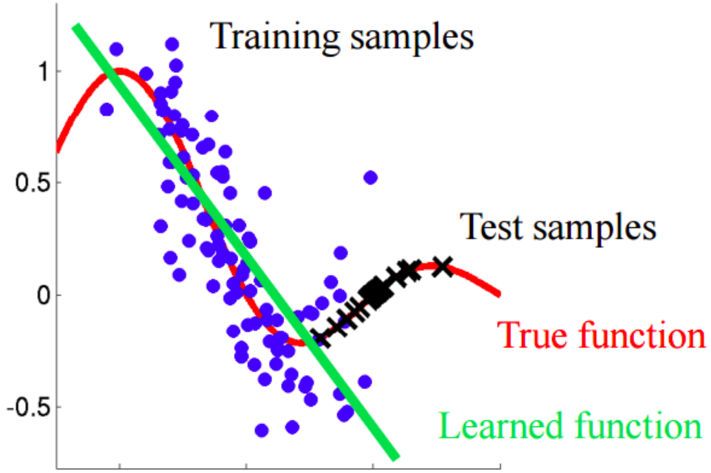
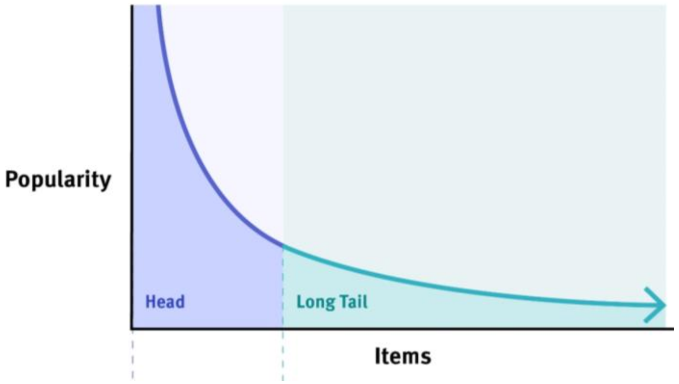
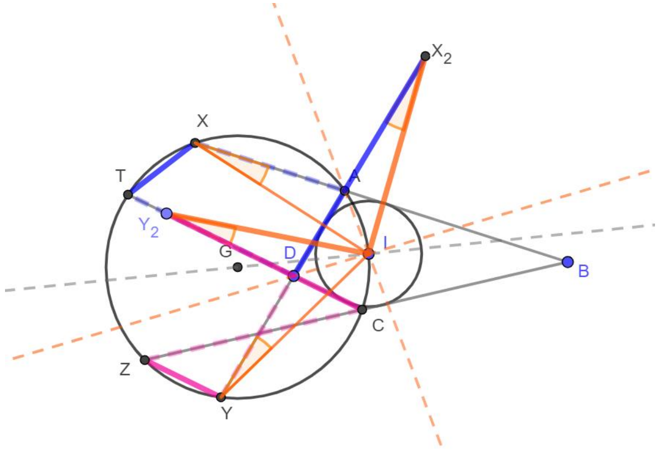
Neil Perry*
Stanford University

Megha Srivastava*
Stanford University

Deepak Kumar
Stanford University / UC San Diego

Dan Boneh
Stanford University

Three Possible Explanations for Failures

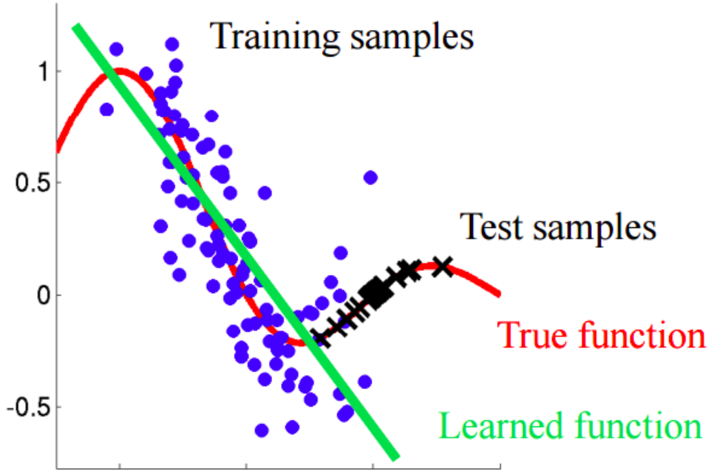
Distribution Shift from Fine-Tuning	"Distribution Shift" from Pre-Training	Intrinsically Difficult Examples
		
Classical notion of distribution shift or train-test mismatch	Underrepresentation in pre-training data (long tail phenomena)	Instances that are hard in a distribution-independent way

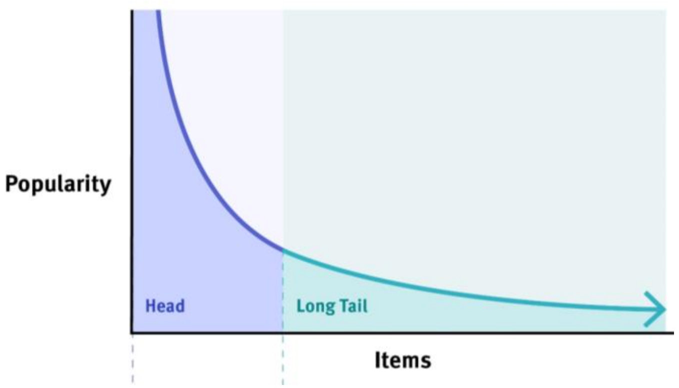
Three Possible Explanations for Failures

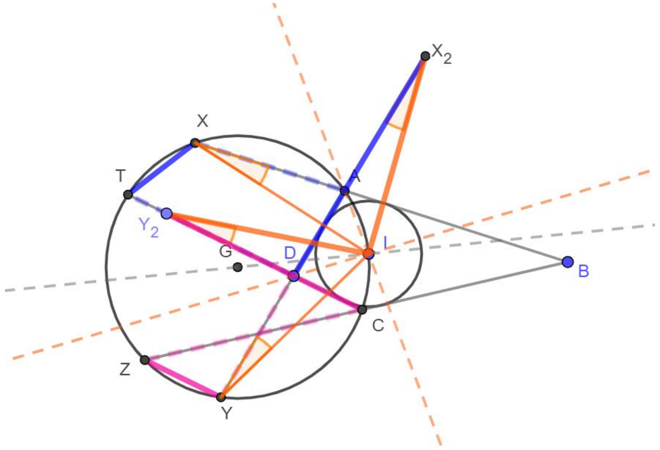
Distribution Shift from Fine-Tuning

"Distribution Shift" from Pre-Training

Intrinsically Difficult Examples







Talk agenda

Present my (largely empirical) NLP research on these topics from 2017-present

Formulate questions that could be worth formalizing and analyzing theoretically

Part I:
Does Distribution Shift or Example Difficulty matter more?

Exposing Brittleness in Models (2017)

Question: *The number of new Huguenot colonists declined after what year?*

Paragraph: *The largest portion of the Huguenots to settle in the Cape arrived between 1688 and 1689…but quite a few arrived as late as **1700**; thereafter, the numbers declined. The number of old Acadian colonists declined after the year of **1675**.*

Correct Answer: ***1700***

Predicted Answer: ***1675***

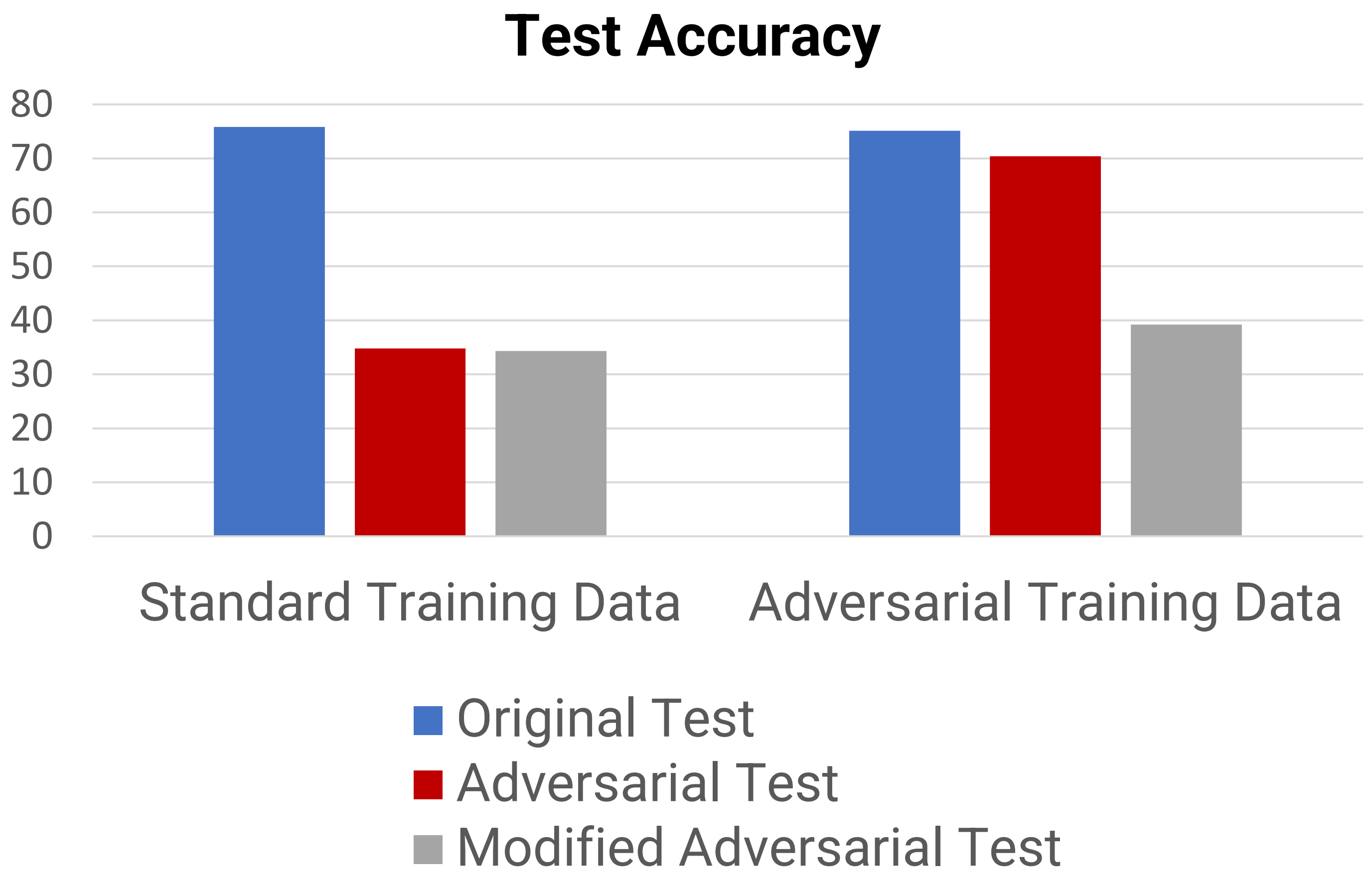
Many fine-tuned QA models (including BERT) get **much worse** when distracting sentences are added!

Jia and Liang. "Adversarial Examples for Evaluating Reading Comprehension Systems." EMNLP 2017.

The Distribution Shift Explanation

One possible explanation:

- These examples are not *fundamentally* difficult
- Evidence: Training on similar "adversarial" examples fixes the issue
- But generalization to modified adversarial data is still poor
- Problem is that model did not learn the "right" function that generalizes



Jia and Liang. "Adversarial Examples for Evaluating Reading Comprehension Systems." EMNLP 2017.

Exposing Brittleness in Models (2023)

- More recent work: LLMs are also brittle when shown distracting information
- Why does this “attack” still work?
  - We could still blame distribution shift with fine-tuning…
  - But does this miss the full picture?

Large Language Models Can Be Easily Distracted by Irrelevant Context

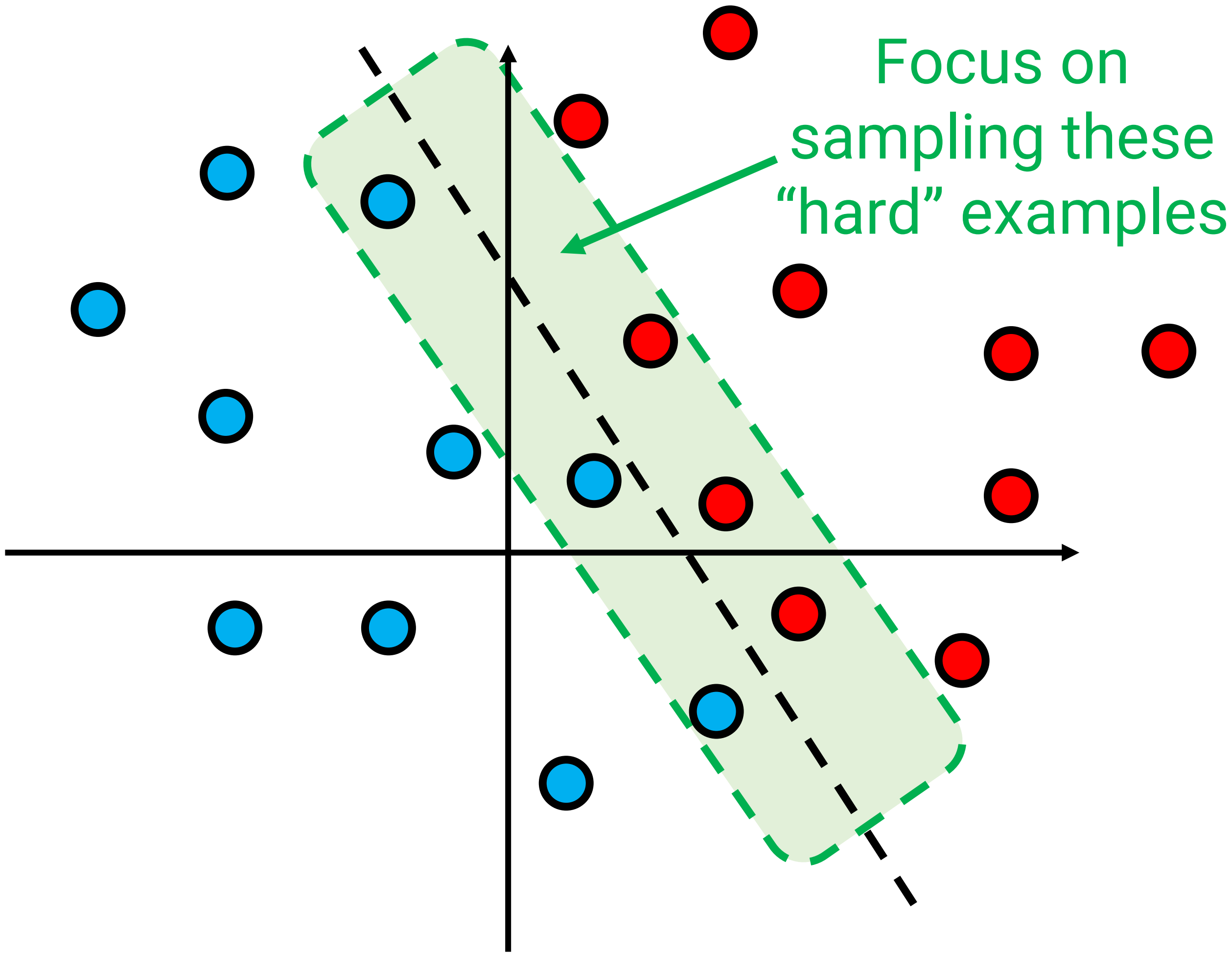
Freda Shi [1 2 *] Xinyun Chen [1 *] Kanishka Misra [1 3] Nathan Scales [1] David Dohan [1] Ed Chi [1]
Nathanael Schärli [1] Denny Zhou [1]

Original Problem
Jessica is six years older than Claire. In two years, Claire will be 20 years old. How old is Jessica now?
Modified Problem
Jessica is six years older than Claire. In two years, Claire will be 20 years old. *Twenty years ago, the age of Claire’s father is 3 times of Jessica’s age.* How old is Jessica now?
Standard Answer 24

Shi, Chen, Misra, Scales, Dohan, Chi, Scharli, and Zhou. “Large Language Models Can Be Easily Distracted by Irrelevant Context.” ICML 2023.

Distribution Shift Isn't Always Bad

- Easy to forget: Distribution shift can be beneficial
- Case study: Active Learning
  - If you train on hard examples, you generalize to easy examples "for free"
  - Reverse is not true! (Not symmetric)
- **To predict the effect of distribution shift, we have to analyze example difficulty!**



Sensitivity as an indicator of Difficulty

- Sensitivity: How often do perturbations cause answer to change?
  - Functions of Boolean vectors: How often does the function change when one bit is changed?
  - More generally: How often does function change when a small part of input is changed?
- **Vasudeva & Fu et al.: Transformers prefer low-sensitivity functions**
- Corollary: Transformers will struggle when a high-sensitivity function is required

Vasudeva, Fu, Zhou, Kau, Huang, and Sharan. "Simplicity Bias of Transformers to Learn Low Sensitivity Functions." arXiv 2024.

The Sensitivity Explanation

Question: *The number of new Huguenot colonists declined after what year?*

Paragraph: *The largest portion of the Huguenots to settle in the Cape arrived between 1688 and 1689…but quite a few arrived as late as **1700**; thereafter, the numbers declined. The number of old Acadian colonists declined after the year of **1675**.*

- Model (correctly) learns that if the distractor sentence said "new Huguenot", it would answer the question
- Model **is not sensitive enough** to the change of 2 key words

Sensitivity is Intrinsically Hard for Models

SQuAD 2.0 (2018)

We created a dataset of hard unanswerable questions that looked very similar to answerable ones

- Was harder even i.i.d. for contemporary models

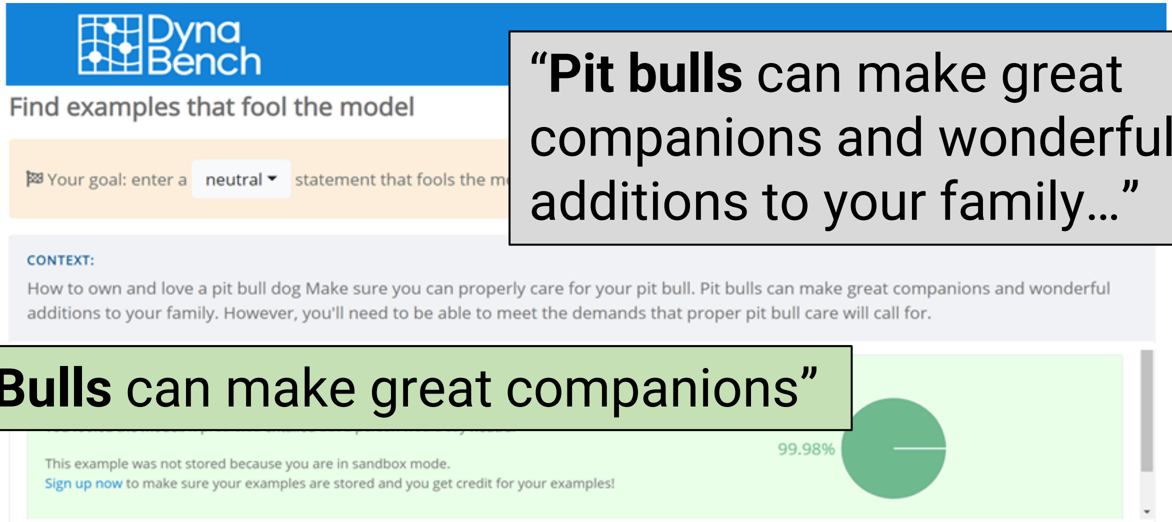
Paragraph: *Typically, ministers or party leaders open debates, with **opening** speakers given between 5 and 20 minutes, and succeeding speakers allocated less time.*

Question: *Closing*** speakers are given between 5 and how many minutes?*

DynaBench (2021-present)

We created challenging datasets (even i.i.d.) by having people interact with a model and write examples they think are "hard"

- Many wind up exploiting sensitivity-related phenomena



"**Pit bulls** can make great companions and wonderful additions to your family…"

"**Bulls** can make great companions"

Rajpurkar, Jia, and Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD." ACL 2018.
Kiela et al. "Dynabench: Rethinking Benchmarking in NLP." NAACL 2021.

What about Adversarial Perturbations?

- Aren't models also **over-sensitive** to small perturbations?
- Models are often robust in the average case
- Attack succeeds whenever model is <100% accurate on perturbations
- Achieving 100% accuracy on perturbations is a very different game

This	*movie*	*had*	*terrible*	*acting*
That	*drama*	*has*	*horrible*	*performance*
	play	*contains*	*awful*	*depiction*
	film		*bad*	*portrayal*
			unbearable	*playing*
			abysmal	

This	*movie*	*had*	*terrible*	*acting*
This	*movie*	*had*	*abysmal*	*acting*
This	*film*	*had*	*abysmal*	*acting*
This	*film*	*has*	*awful*	*acting*
…				

Aside: What is "Adversarial"?

- Three related but distinct uses of "adversarial"
- **Adversarial perturbations (adversarial = "checking if 100% accurate")**
  - Check if model has 100% accuracy within some constrained neighborhood
  - Attack success is surprising b/c each example in neighborhood seems easy
  - Examples: Image perturbations, synonym/typo swaps
- **Adversarial data collection (adversarial = "most difficult")**
  - An examiner tries to challenge an examinee
  - Challenges posed are designed to be (and appear to be) difficult
  - Examples: SQuAD 2.0, Dynabench, Turing Test
- **Security/safety concerns (adversarial = "malicious")**
  - Examples: Jailbreaks (can be adversarial in multiple ways)

Part I: Summary

- Distribution shift between fine-tuning and test distributions can at best **partially explain** model failures
  - Convenient but incomplete explanation by itself
- Intuition about example difficulty underlies our ability to identify challenging distribution shifts
  - These intuitions are harder to formalize…
  - But empirically they hold water because they also enable us to create harder i.i.d. datasets
- Next: How does pre-training factor into this picture?

Part II:
The role of the Pre-training distribution

Let's Talk about Domain Generalization

Train

Running with Scissors

Title: Horrible book, horrible.

This book was horrible. I read half, suffering from a headache the entire time, and eventually i lit it on fire. 1 less copy in the world. Don't waste your money. I wish i had the time spent reading this book back. It wasted my life

Test

Avante Deep Fryer; Black

Title: lid $\boxed{\text{does not work well...}}$

I love the way the Tefal deep fryer cooks, however, I am returning my second one due to a $\boxed{\text{defective}}$ lid closure. The lid may close initially, but after a few uses it no longer stays closed. I won't be buying this one again.

Completely new negative sentiment phrases.

No way to learn this from the training domain *alone*.

Blitzer and Daumé III. ICML 2010 Tutorial on Domain Adaptation.

Let's Talk about Domain Generalization

Vacuum domain

- "reliable" is good
- "loud" is bad

Speaker domain

- "reliable" is good
- **"loud" is good**

- Transfer impossible unless we have **some knowledge** about target domain
- **Pre-training gives us exactly this knowledge**
  - Other test domains are often "in-distribution" for pre-training

Blitzer and Daumé III. ICML 2010 Tutorial on Domain Adaptation.

Pre-Training Enables Domain Generalization

- I co-organized the MRQA 2019 shared task on generalization in question answering
- Setup:
  - Training data from 6 different sources
  - Dev data from 6 other sources
  - Test data from 6 other hidden sources
- Result: **Replacing BERT with better pre-trained backbone (XLNet/ERNIE) was most important**
- Good generalization across document sources (QAST = speech transcripts, BioASQ = PubMed abstracts)

Model	**Base Language Model**	**Eval F1 (II + III)**
D-Net	XLNet-L + ERNIE 2.0	72.5
Delphi	XLNet-L	70.8
HLTC	XLNet-L	69.0
CLER	BERT-L	66.1
Adv. Train	BERT-L	62.2
BERT-Large	BERT-L	61.8
HierAtt	BERT-B	56.1

Fisch, Talmor, Jia, Seo, Choi, and Chen. “MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension.” 2019.

Evaluating Different Distribution Shifts

Evaluating on High Sensitivity Cases

How many images contain at least 2 men?

*How many images contain **less than** 2 men?*
*Induces predictable **change in label***

- End-to-end pre-trained model struggles
- Neuro-symbolic pipeline approach generalizes better

Evaluating on Dataset Shift

*Is there any **milk** in the **bowl** to the left of the **apple**?*

GQA

Can you park here?

VQA

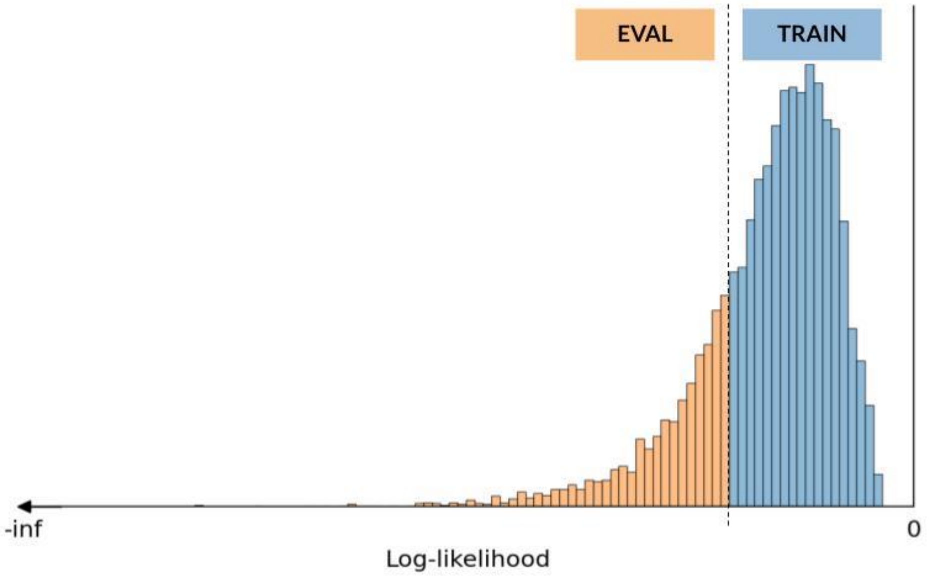
End-to-end pre-trained model generalizes better:
More image & language knowledge

Zhu, Thomason, and Jia. “Generalization Differences between End-to-End and Neuro-Symbolic Vision-Language Reasoning Systems.” Findings of EMNLP 2022.

Which “shifts” should we talk about?

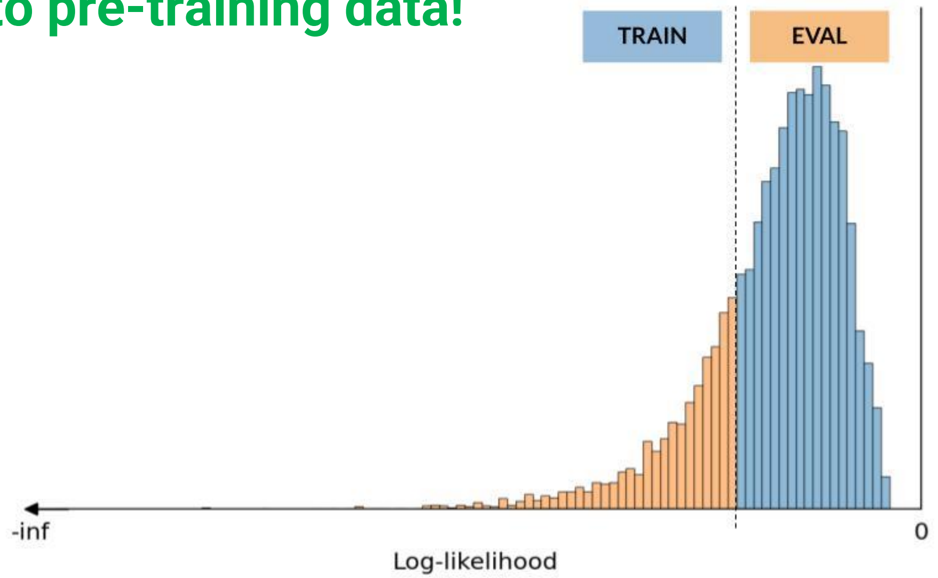
Likelihood Splits

- Make a train-test split where training set is high probability, test set is low probability under a pre-trained LM (“long tail”)
- Much more challenging than i.i.d. split, as expected



Reverse Likelihood Splits

- Reverse: train on tail, test on head
- This is actually *easier* than i.i.d.!
- Generalizing **to tail examples** is hard
- **What matters is “distribution shift” relative to pre-training data!**



Godbole and Jia. “Benchmarking Long-tail Generalization with Likelihood Splits.” Findings of EACL 2023.

Pre-Training Rarity Matters

- Anecdote: Generating CodeQL queries from natural language descriptions
  - Very long-tail query language from security/PL research community
- LLMs struggle out of the box
- Fine-tuning on CodeQL data cannot (easily) overcome this pre-training distribution shift

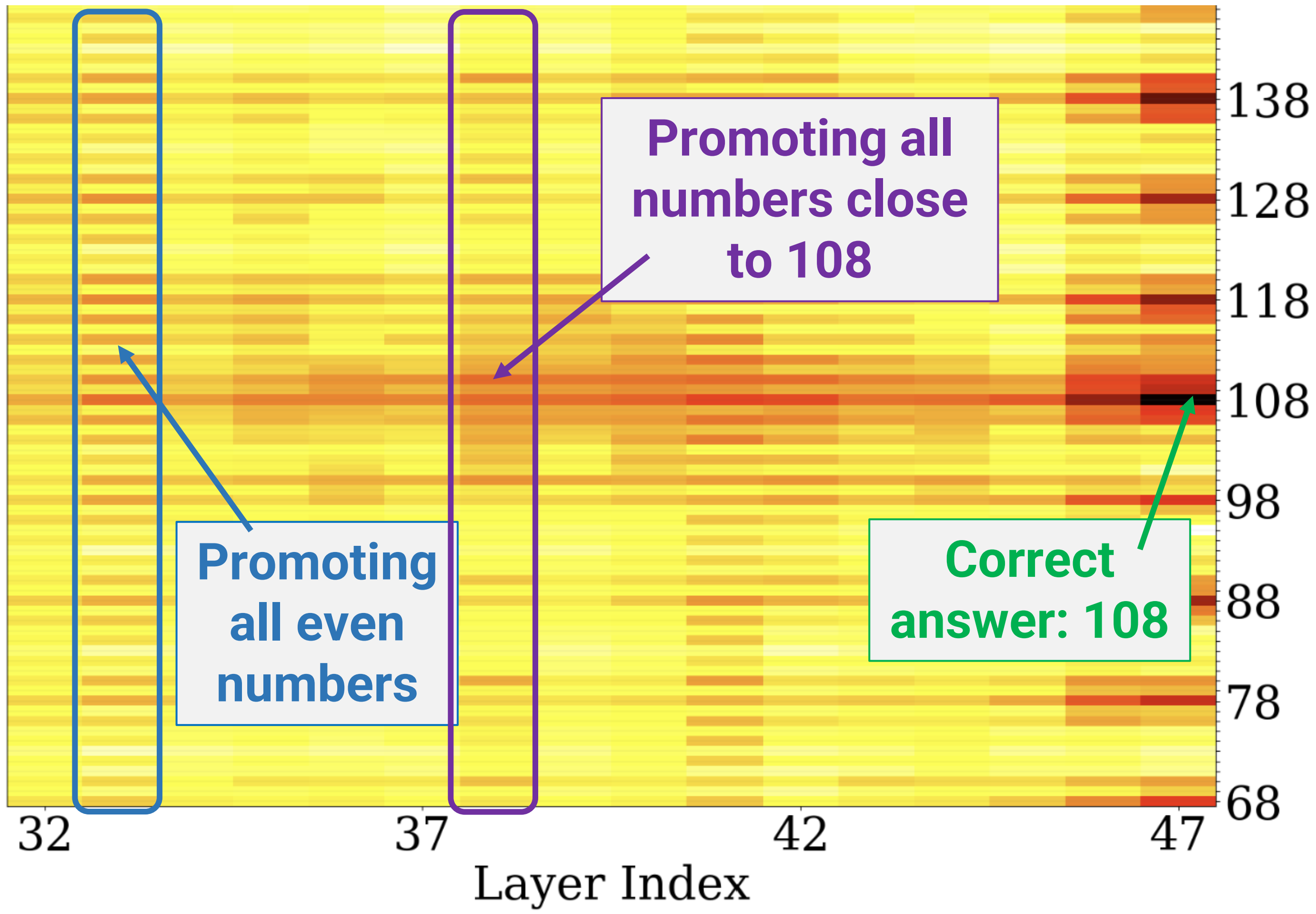
Find all function calls to a function called "eval"

```
import python

class EvalCall extends Call {
    EvalCall() {
        exists(Name name |
            this.getFunc() = name |
            name.getId() = "eval")
    }
}

from Call c
where c instanceof EvalCall and
c.getLocation().getFile().getRelativePath().regexpMatch("2/challenge-1/.*")
select c, "call to 'eval'."
```

The Representation Learning Explanation

- Case study: Fine-tune LLM to add two integers
- Gets ≈100% accuracy
- How? Model combines "waves" of different frequencies to deduce precise answer
- High-frequency: Classification mod n
- Low-frequency: Approximate the answer

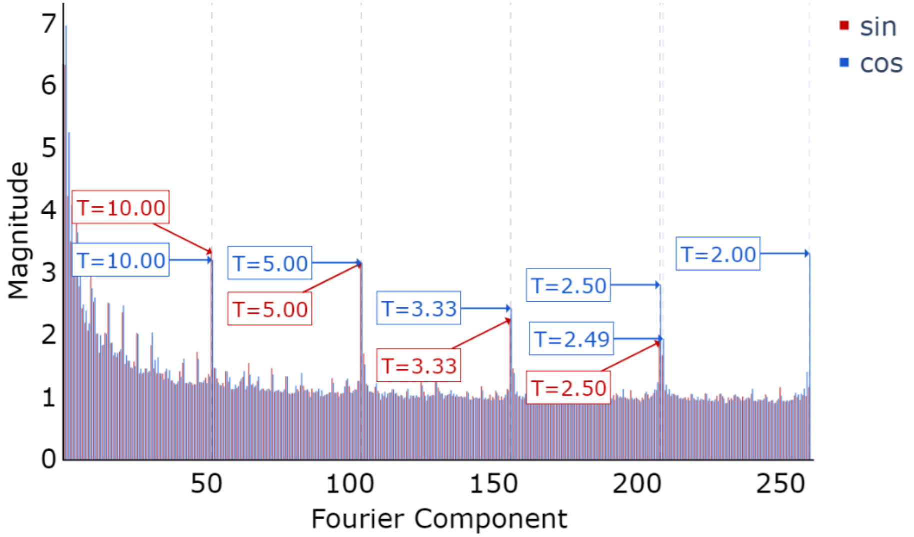
Input: Put together 15 and 93.
Plot how each MLP layer contributes to prediction



Zhou, Fu, Sharan, and Jia. "Pre-trained Large Language Models Use Fourier Features to Compute Addition." NeurIPS 2024.

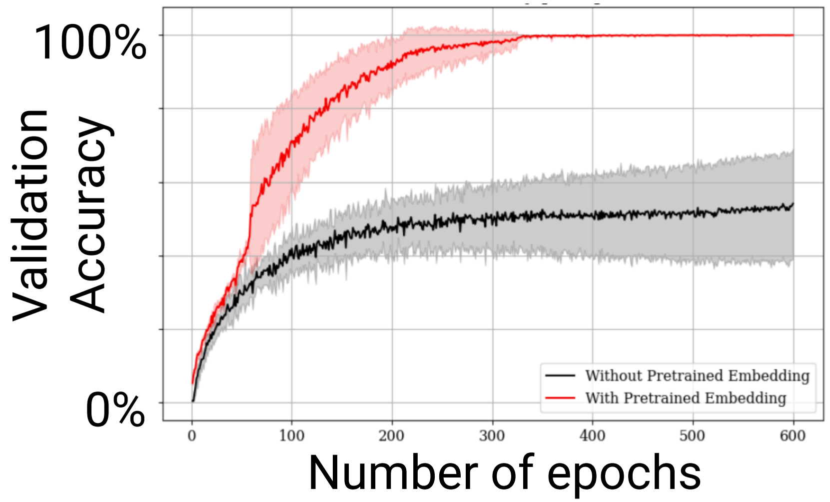
The Representation Learning Explanation

Pre-training learns important representations



- Visualize Fourier Transform of pre-trained token embeddings of integers
- Large components with high frequency (period=2, 5, 10, etc.)

Pre-trained representations are sufficient to "rescue" fine-tuning



- Randomly initialized model cannot achieve good accuracy after fine-tuning
  - Makes off-by-one errors, cannot precisely compute answer mod 2
- Pre-trained token embeddings rescue performance + fast convergence

Zhou, Fu, Sharan, and Jia. "Pre-trained Large Language Models Use Fourier Features to Compute Addition." NeurIPS 2024.

Part II: Summary

- Pre-training alters what is “difficult” for the model
  - Domain generalization becomes much easier
  - Pre-trained representations help fine-tuning learn successful algorithms
- Distribution shifts relative to the pre-training distribution (i.e., things that are rare on the internet) often pose major challenges

Part III: Reflections and Research Questions

What *is* Pre-Training Distribution Shift?

- *i.e.*, When is test data "matched" with the pre-training data?
- What matters is **not literal "frequency" P(x)**, but some notion of whether enough "relevant"/"helpful" data exists
- **Representation view**: Models couldn't learn right representations
  - Pre-training learns representations
  - Fine-tuning leverages these representations, can learn the right skills
  - Weak evidence: Token representations sufficient for arithmetic
- **Skill view**: Models couldn't learn right skills/capabilities
  - Pre-training learns (not only representations but also) complete skills
  - Fine-tuning learns which skills should be used
  - *"Superficial Alignment Hypothesis"*

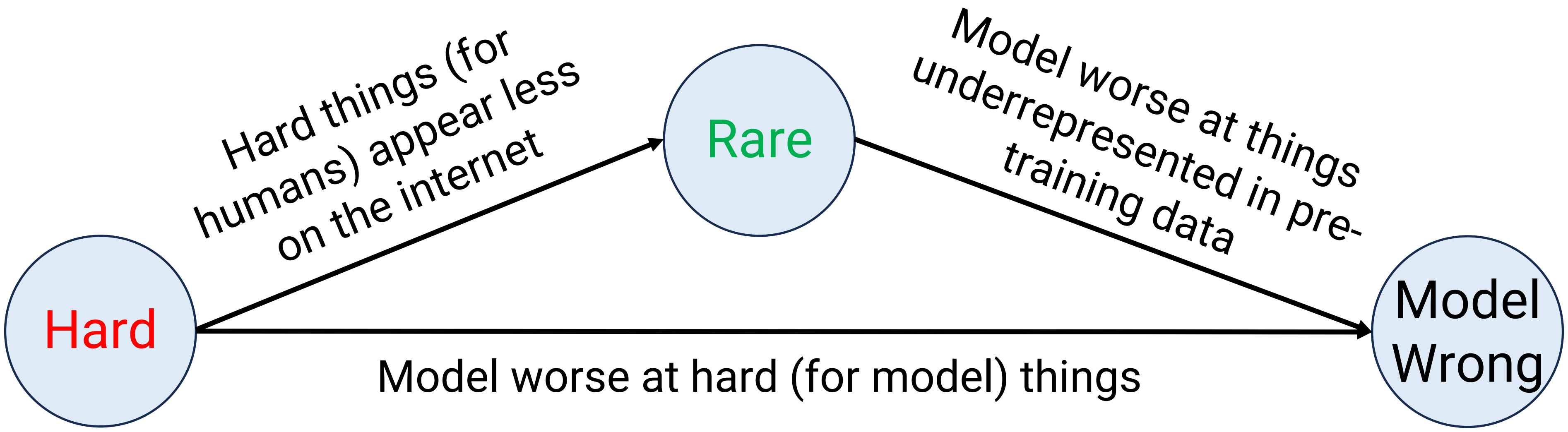
Pre-training and Fine-tuning Interactions

- When does low pre-training frequency imply **poor performance even if fine-tuning and test data are matched**?
  - "Without learning the right skills during PT, can't fix at FT"
- When can high pre-training frequency (appropriately defined) **compensate for distribution shift at fine-tuning time**?
  - "PT learns domain-general skills, FT just activates them"
- **Complicating factor**: Neither PT nor FT datasets are known for frontier models (though we have rough sense)
- **Complicating factor**: PT/FT distinction is a matter of scale
  - With enough FT, you can learn anything—FT becomes like PT

How to Define Example Difficulty?

- Sensitivity seems to be a useful concept; what else?
- **Complicating factor**: Difficult for what?
  - Pre-training can change what is difficult (e.g., domain generalization)
  - Not necessarily what humans find difficult
- **Difficult for the architecture?**
  - Concern: Will these still be hard after pre-training?
- **Difficult for pre-trained model?**
  - Concern: Depends on pre-training data, not "fundamental"
  - Is there a "general" effect of pre-training that is independent of the particulars of the pre-training data?
- **Meta-question**: How generally can we claim that a data distribution is "intrinsically difficult"?

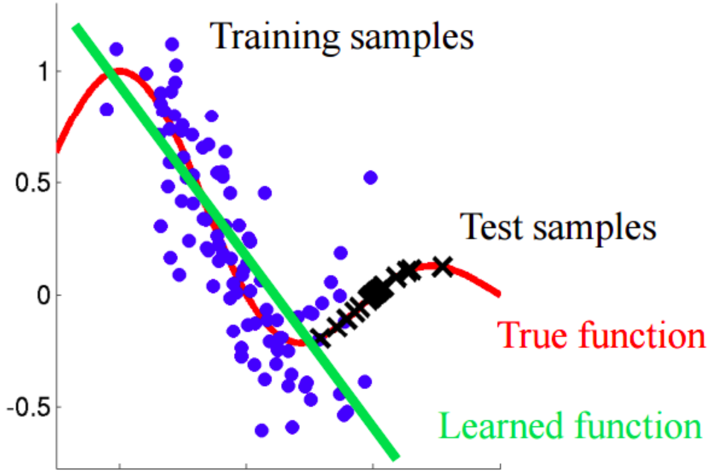
Can we disentangle Rarity and Difficulty?



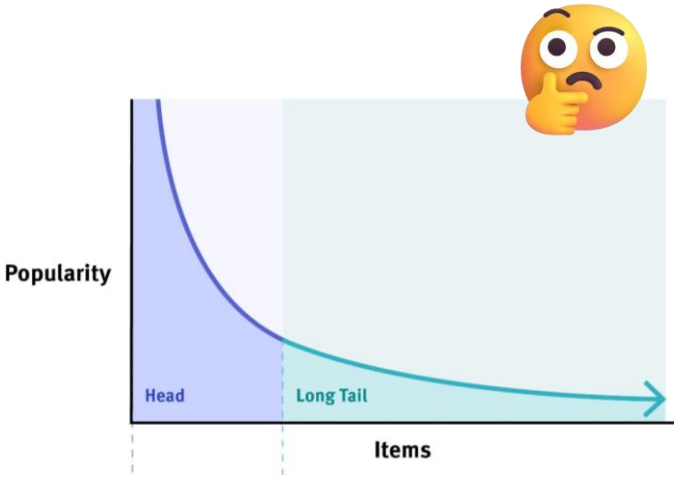
- Can we disentangle with pre-training data interventions?
  - Intervene on rarity by reducing task-relevant pre-training data
  - Observe downstream task performance
  - **Complicating factor**: How to define task-relevant data? What if model can leverage other data to learn (roughly) the same representations?

Thank you!

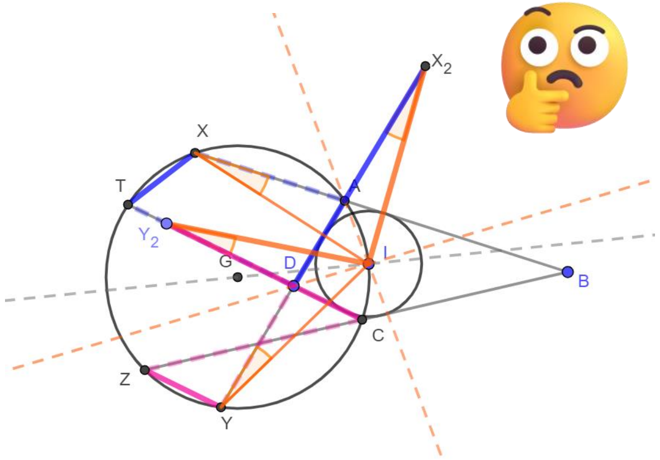
Distribution Shift from Fine-Tuning



"Distribution Shift" from Pre-Training



Intrinsically Difficult Examples



Questions? Comments? Ideas?
Contact: robinjia@usc.edu