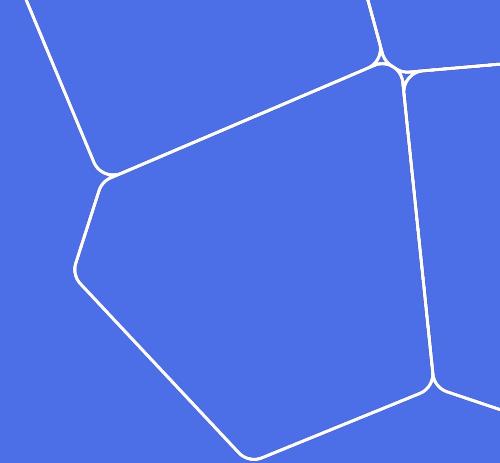


Part I: Understanding the role of data, scale and capacity in recent breakthroughs

MLRS 2023
Sara Hooker - Cohere For AI





I lead Cohere For AI, a research lab that works on fundamental machine learning research.

Open Science – contributing technical work to the open domain.

Full time research staff – focus on large language models and AI at scale.

Cohere For AI Research Programs – Scholars Program.

Team C4AI



**Sara
Hooker**
Head



**Ahmet
Üstün**
Research
Scientist



**Marzieh
Fadaee**
Research
Scientist



**Beyza
Emiş**
Research
Scientist



**Brittawnya
Prince**
Operations
Associate

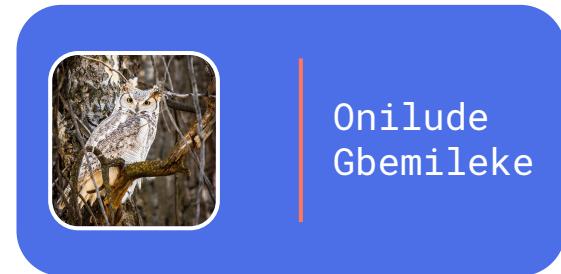


**Madeline
Smith**
Community
& Outreach
Specialist



**Ellie
Evans**
Responsible
AI

C4AI Research Scholars



C4AI Research Scholars Program - Deadline September 11th 2023

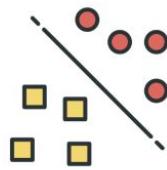
↗ Cohere For AI
Scholars Program

Calling all aspiring
ML researchers.
Your journey starts here.

Application deadline Sept. 11

My research agenda to-date has focused on:

- Making large language models accessible by improving efficiency.
- Training models to fulfill multiple criteria



Model Compression - compact machine learning models to work in resource constrained environments.



Model fragility and security - deploy secure models that protect user privacy.



Fairness - imposes constraint on optimization that reflects societal norms of what is fair.



Model Interpretability - reliable explanations for model behavior.

I currently work on designing large scale language models that are **efficient, multilingual, reliable and trustworthy.**

If any of these topics are interesting the talk, happy to discuss after the talk.

Outline for today's talk:

Session 1:

Understanding the role of data,scale and capacity in recent breakthroughs.

Q&A

Session 2:

The Myth of the perfect model: Trade-offs between desirable properties.

Q&A



We are currently in a very exciting time for large language modeling research.

I'm giving a talk this afternoon about large language models. Can you give me 5 reasons why large language models are exciting?

1. Large language models can help us understand the world better by providing insights into how language works.
2. They can be used to generate content, such as stories and poems, that can entertain and inspire us.
3. They can be used to translate languages, helping us to communicate with people from different cultures.
4. They can be used to summarize and extract information from large amounts of text, making it easier for us to consume and understand.



We are currently in a very exciting time for large language modeling research.

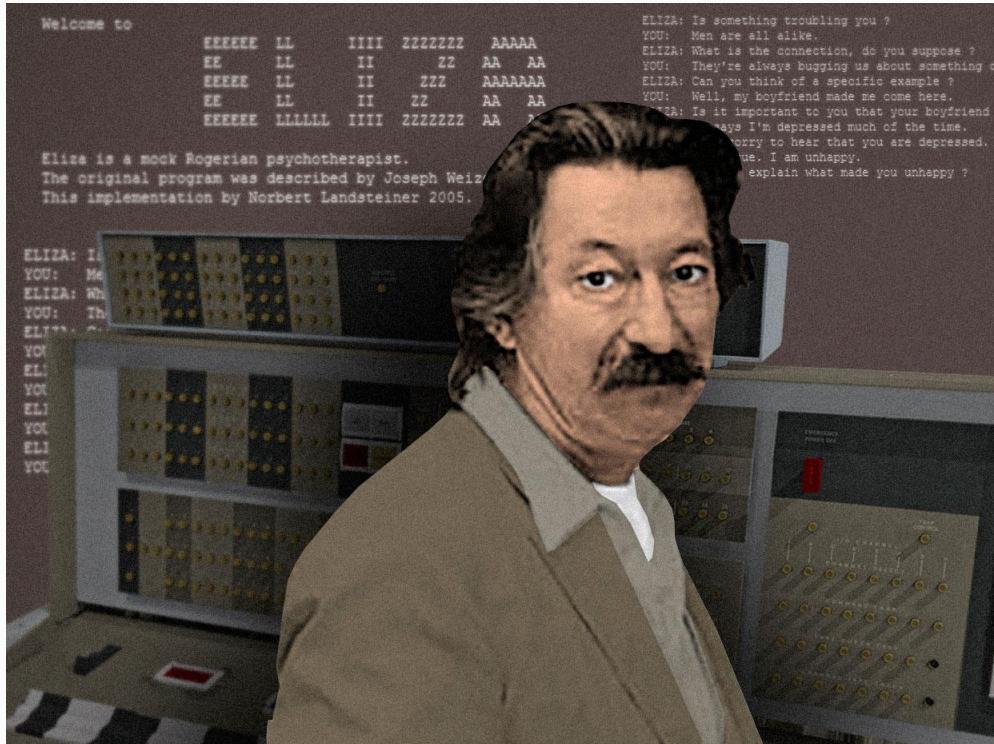
What are good places to visit around bangkok

Bangkok has a lot of must-see attractions that would make for an unforgettable trip. The Grand Palace and temples such as Wat Pho, Wat Arun, and Wat Traimit (the Temple of the Golden Buddha) are popular attractions for visitors.

You could also visit Chatuchak Weekend Market, a massive weekend market with over 15.000

While all of this is very exciting, in
the time we have together today
– I thought it might be fun to
understand how we got here.

I want to introduce you to Joseph Weizenbaum – who made one of the chatbots called Eliza.

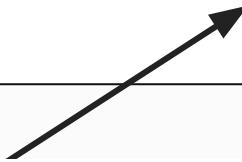


Eliza was based upon a popular form of therapy (rogerian therapy) – that involved repeating back to the patient what they just said.

Eliza used rules to determine the reply. There was no deep learning. You can see how this can fall off a cliff quickly.

Talk to Eliza by typing your questions and answers in the input box.

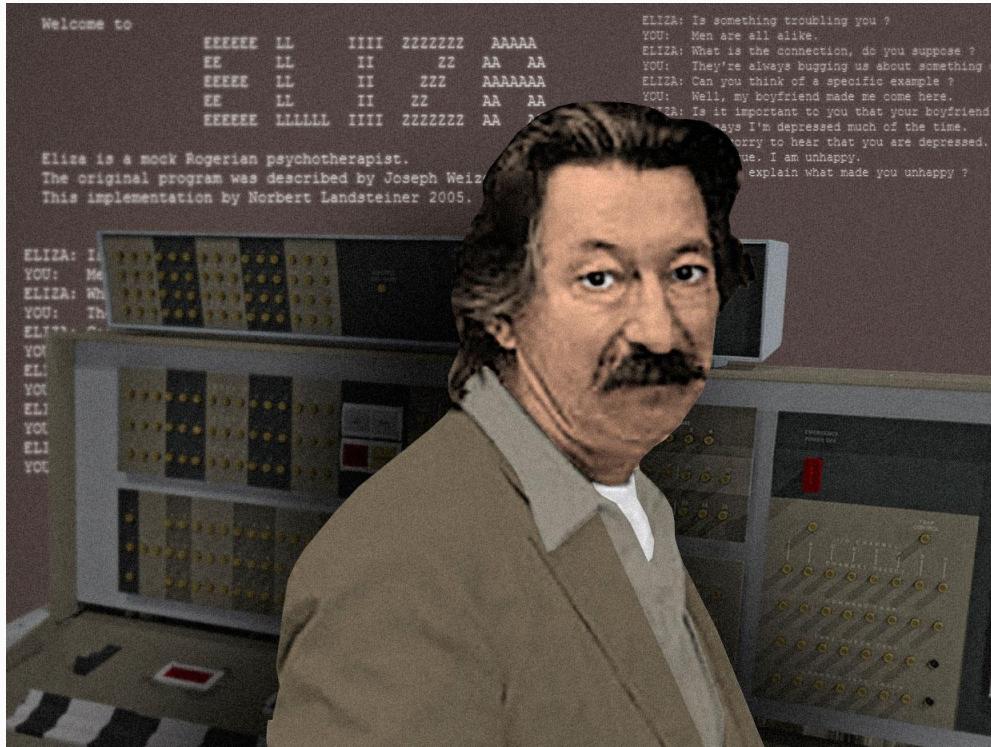
```
> Hello, I am Eliza. I'll be your therapist today.  
* I'm giving a research talk tonight about large language models. Can you  
give me 5 reasons large language models are exciting?  
> Is it that you would like to be able to give you 5 reasons large language  
models are exciting?  
* Yes.  
> I see. Let's explore that a bit more.
```



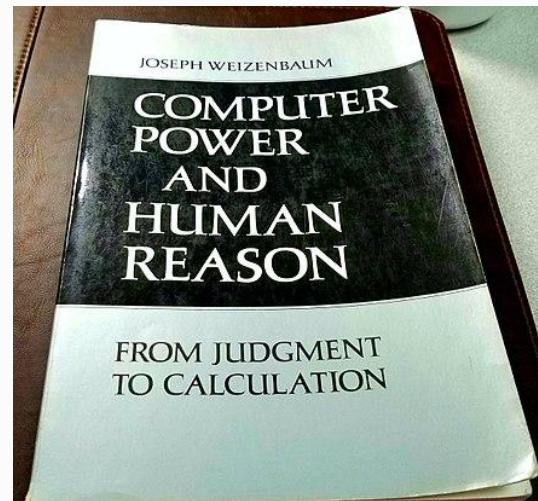
If Eliza couldn't identify a keyword or phrase, it uses a stock phrase, such as "Please go on." or "Let's explore that a bit more."

You can explore Eliza yourself at
<https://web.njit.edu/~ronkowitz/eliza.html>

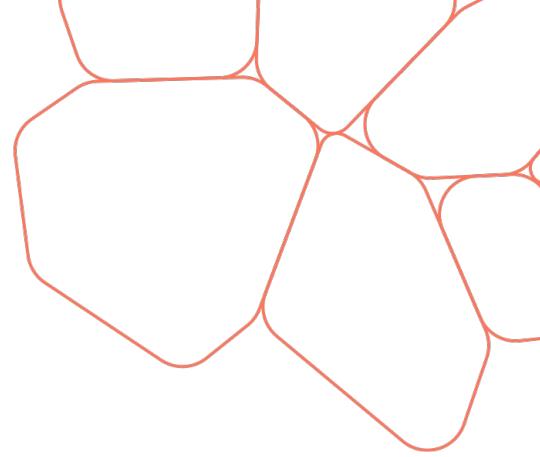
Nonetheless, many users of Eliza were convinced of ELIZA's intelligence and understanding.



Joseph became concerned when he caught his secretary using the program and telling it confidential things about her boyfriend.



What separates Eliza from our current chatbots? What allows for research breakthroughs.



2012

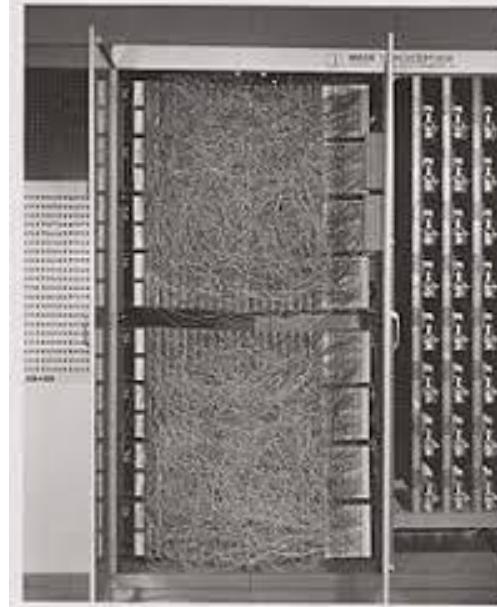
1964

2017

For most of computer science history – two very different visions of how to make progress.



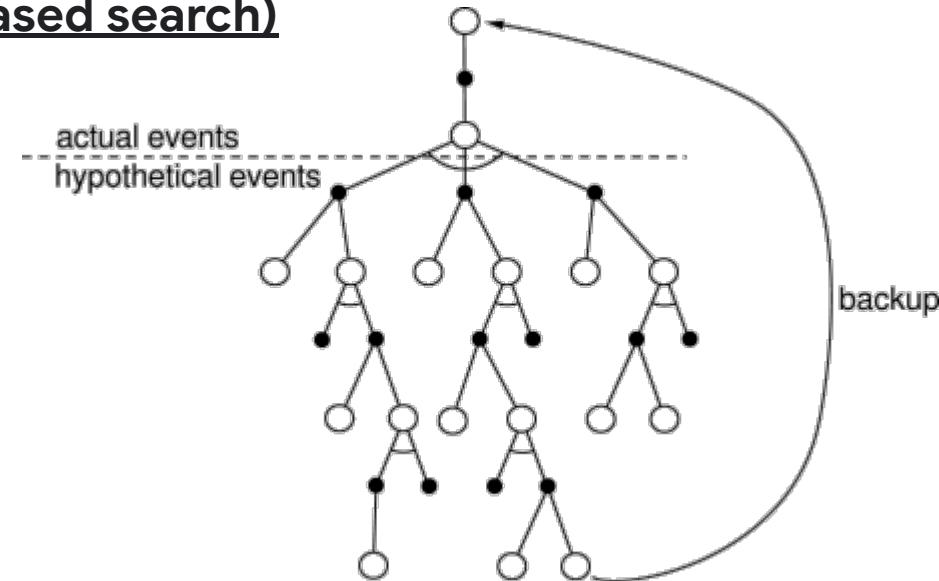
**Rule based expert systems
(ELIZA)**



**Deep Neural Network
Approaches**

Rule based systems dominated mainstream research efforts from 1950-1980's

Checkers (1956) Arthur Samuel (logic based search)



Researchers working on deep neural networks were marginalized for several decades. This continued through 2012.

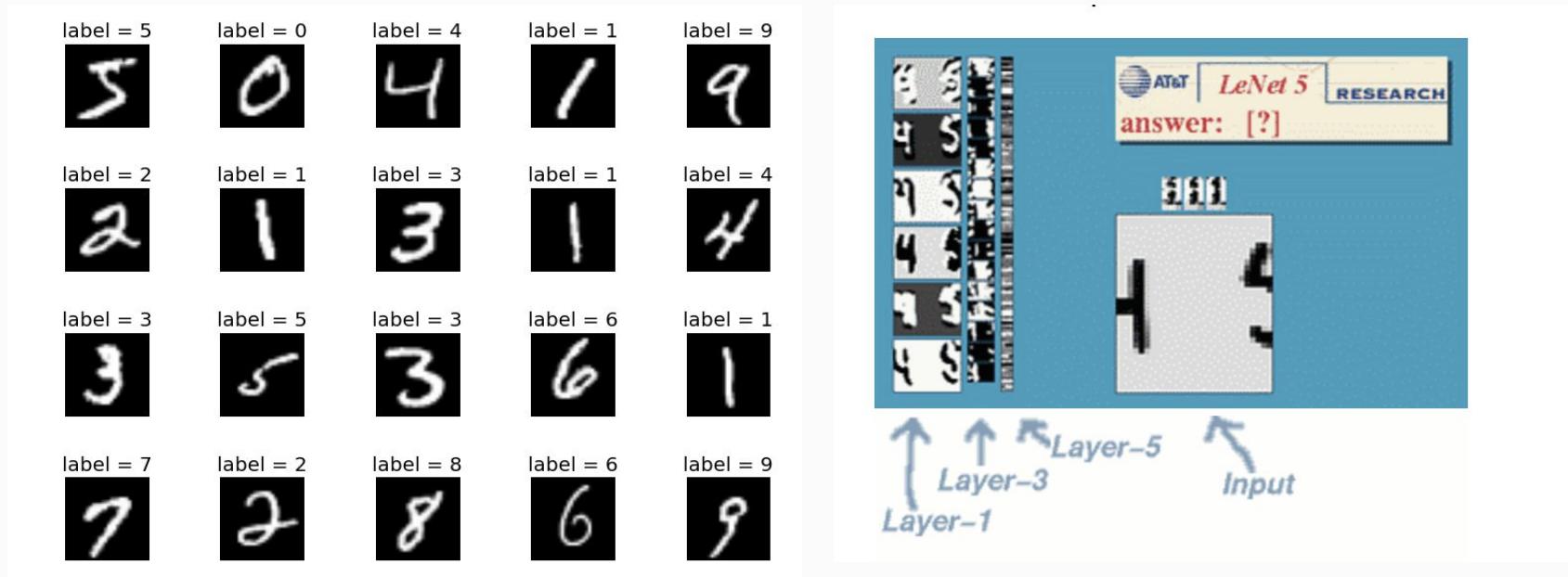


"Further discussion of it merely incumbers the literature and befogs the mind of fellow students."

- 2007: NIPS program committee rejects a paper on deep learning by *al. et.* Hinton because they already accepted a paper on deep learning and two papers on the same topic would be excessive.
- ~2009: A reviewer tells Yoshua Bengio that papers about neural nets have no place in ICML.
- ~2010: A CVPR reviewer rejects Yann LeCun's paper even though it beats the state-of-the-art. The reviewer says that it tells us nothing about computer vision because everything is learned.

Link [here](#)

During this time, only a few countries preserved funding for connectionist ideas.



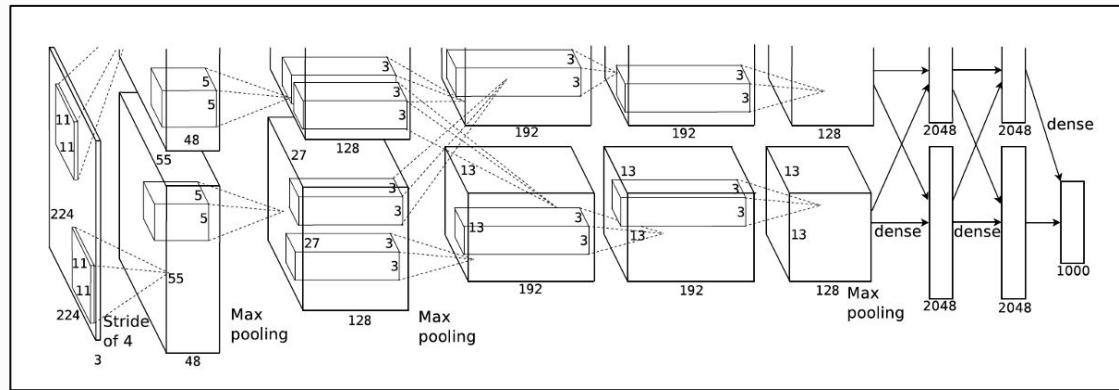
Breakthroughs in Digit Recognition (1998)

Image source: Yann Lecun [site](#), [CS Toronto](#), [ImageNet.org](#)

We know in hindsight that the breakthrough for deep neural networks was in 2012, when overnight AlexNet dominated ILSVRC.

AlexNet achieved a top-5 error of 15.3%, more than 10.8 percentage points lower than that of the runner up.

Everyone switched to convolutional neural networks overnight.



This breakthrough was in part due to a lucky fluke.

Hardware designed for video games was repurposed for training deep neural networks.

- GPUs developed for video games
- Good at parallelizing matrix multiplies

The Hardware Lottery

Sara Hooker

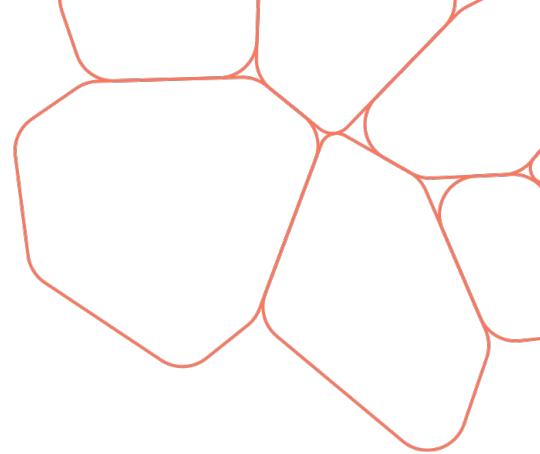
Google Research, Brain Team
shooker@google.com

Abstract

Hardware, systems and algorithms research communities have historically had different incentive structures and fluctuating motivation to engage with each other explicitly. This historical treatment is odd given that hardware and software have frequently determined which research ideas succeed (and fail). This essay introduces the term hardware lottery to describe when a breakthrough idea is “lucky” in that it is enabled by the available hardware.



Overnight in 2012, everyone switched to deep neural networks.



Perseverance over
decades led to the
breakthrough of deep
neural networks in
computer vision.

2012

1964

2017

2017: The Transformer, the culmination of a rich history of language modelling

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

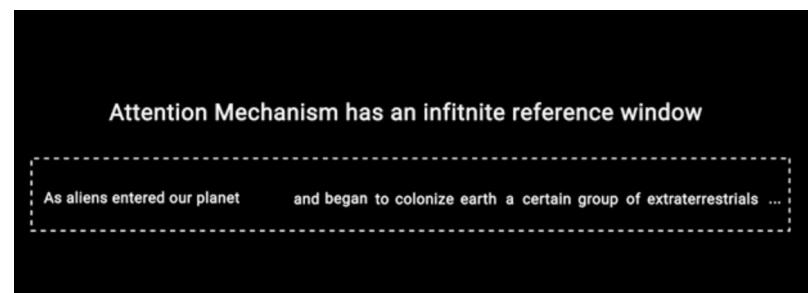
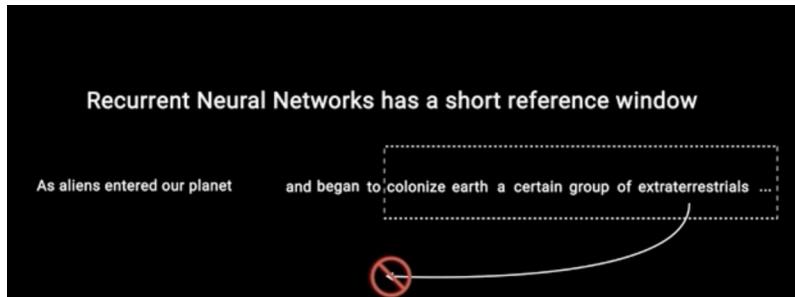
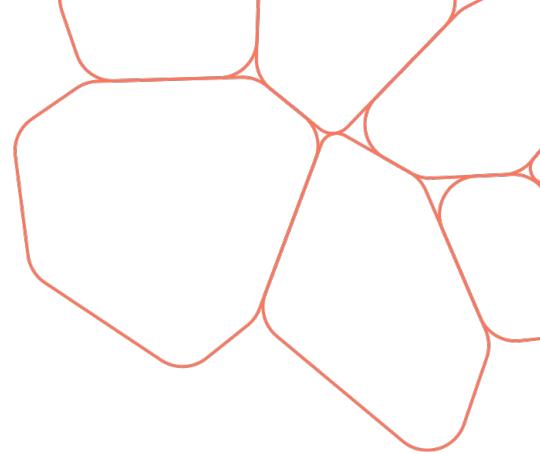
Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

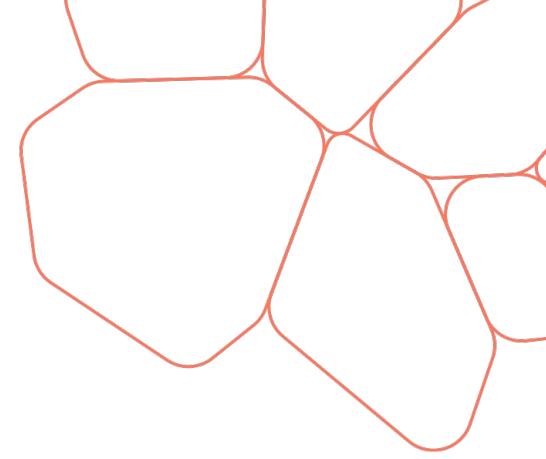
Lukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Transformers were a breakthrough at modelling longer text dependencies. Overnight everyone abandoned LSTMs and switched to transformers.



Transformers are the shared building block of almost all large language models.



1964

Perseverance over decades led to the breakthrough of deep neural networks in computer vision.

2012

Algorithm breakthrough – everyone switches to Transformers.

2017 2023

2017-2023: The great acceleration.

Pretraining on larger and larger datasets in an unsupervised fashion.

Step 1:
Unsupervised
pre-training of
a transformer
model on a
massive web
crawled dataset
(i.e. train on
the internet).

Text: Second Law of Robotics: A robot must obey the orders given it by human beings



Generated training examples

Example #	Input (features)	Correct output (labels)
1	Second law of robotics :	a
2	Second law of robotics : a	robot
3	Second law of robotics : a robot	must
...		

<https://jalammar.github.io/how-gpt3-works-visualizations-animations/>

Why do we pretrain?

Changes in optimization strategy – previously, we would finetuning the model one task at a time.

3 Fine-tuning

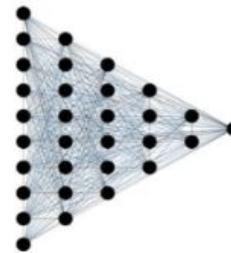


Small labeled
dataset

We wish to suggest a structure for the salt of deoxyribose nucleic acid (D.N.A). This structure has novel features which are of considerable biological interest.

Text

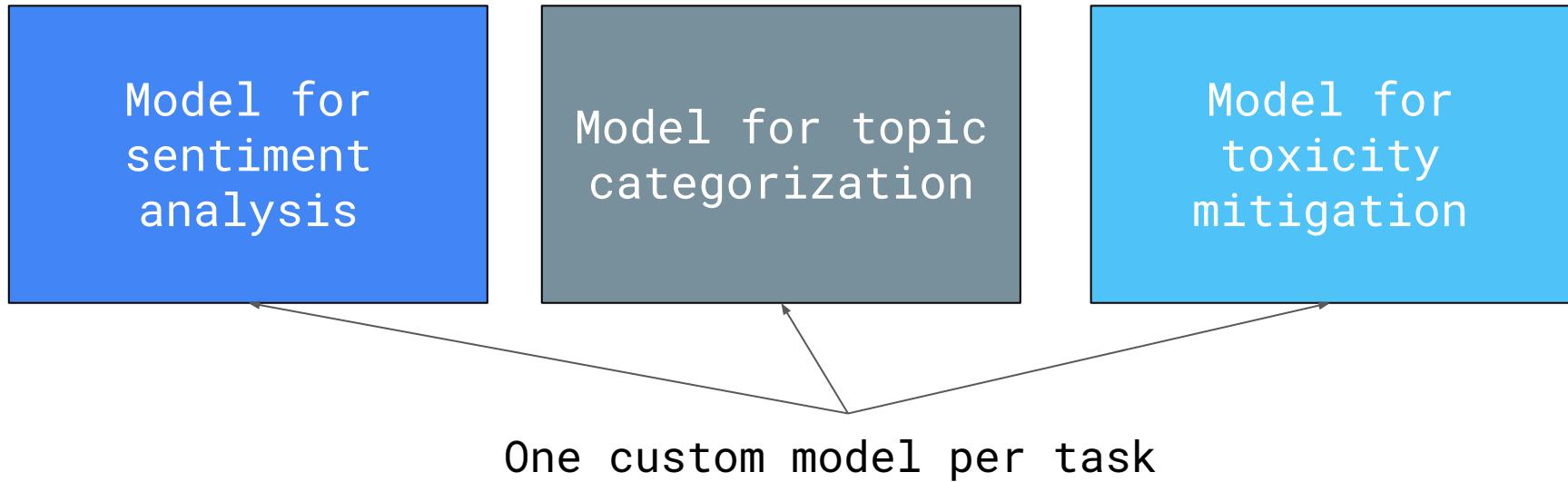
Fine-tuned model



Topic: Biology (97%)

Prediction

Changes in optimization. This meant we would end up with dedicated independent models for each downstream task:



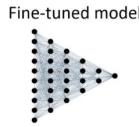
Changed to multi-task fine-tuning. Moving to a single global model – train on multiple tasks at once.

3 Fine-tuning



We wish to suggest a structure for the salt of deoxyribose nucleic acid (DNA). This structure has novel features which are of considerable biological interest.

Text



Topic: Biology (97%)
Prediction

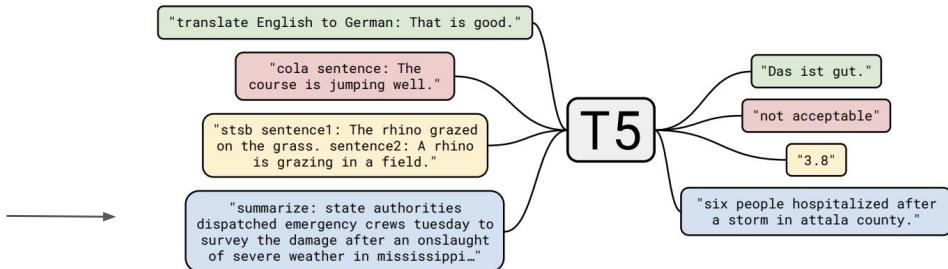


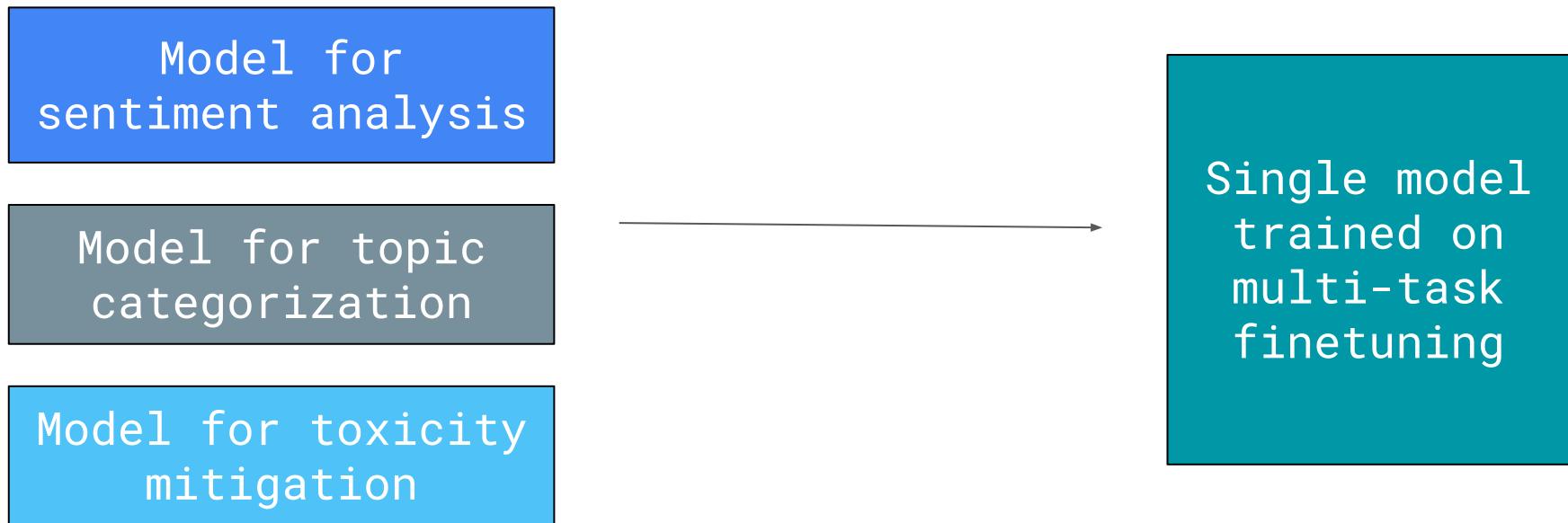
Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. “T5” refers to our model, which we dub the “Text-to-Text Transfer Transformer”.

Finetuning on a single task



Finetuning on many different tasks

Why is this a big deal – it transitions from having custom models for each task to having a single task-general model that can perform a lot of tasks, which only require zero or few examples



The role of data in recent breakthroughs

There has been renewed interest in the types of data that lead to powerful zero shot abilities – **performing well on completely unseen tasks.**

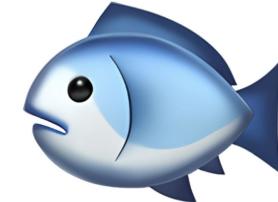
Today 5:41 PM

Tell me a story about a wizard fish who goes to a boarding school underwater.

There once was a wizard fish named Finneus who was very excited to start his first year at a boarding school underwater. He was a bit nervous about making new friends and getting used to his new school, but he was also excited to learn all about magic and how to use it.

On the first day of school, Finneus was a bit shy and didn't talk to anyone. But by the end of the day, he had made a few friends and was looking forward to the rest of the year.

Every day, Finneus learned new spells and potions and he loved every minute of it. He also made more friends and had lots of fun with them.



Text tiny Command +1 (778) 718-8884

It turns out two ingredients have been particularly important at leading to breakthroughs in performance on zero and few shot tasks:

- 1. Structuring multi-task finetuning data as questions and answers.**

1. Structuring multi-task fine-tuning data as questions and answers.

Step 1: Prompts

Read the Original Prompt

Quem foi a primeira mulher a se eleger presidente de um país?

- a) Dilma Rousseff
- b) Cristina Kirchner
- c) Vigdís Finnbogadóttir
- d) Margaret Thatcher
- e) Jacinda Ardern

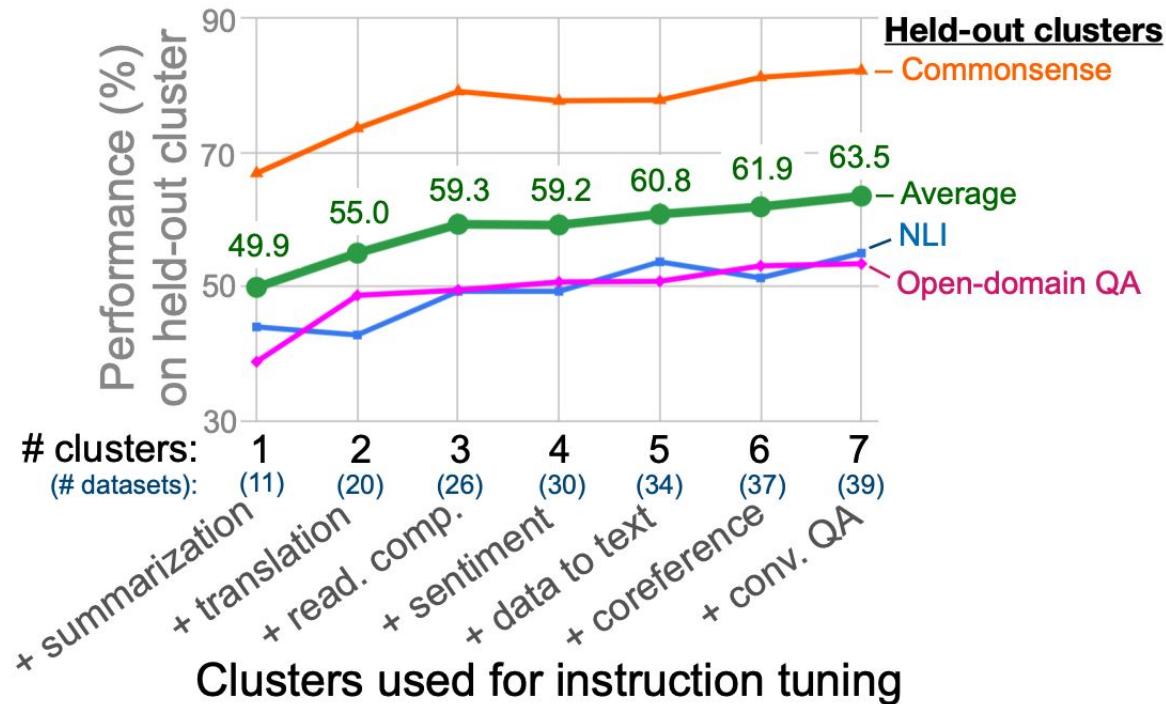
Step 2: Completions

Read the Original Completion

Resposta correta: c) Vigdís Finnbogadóttir.

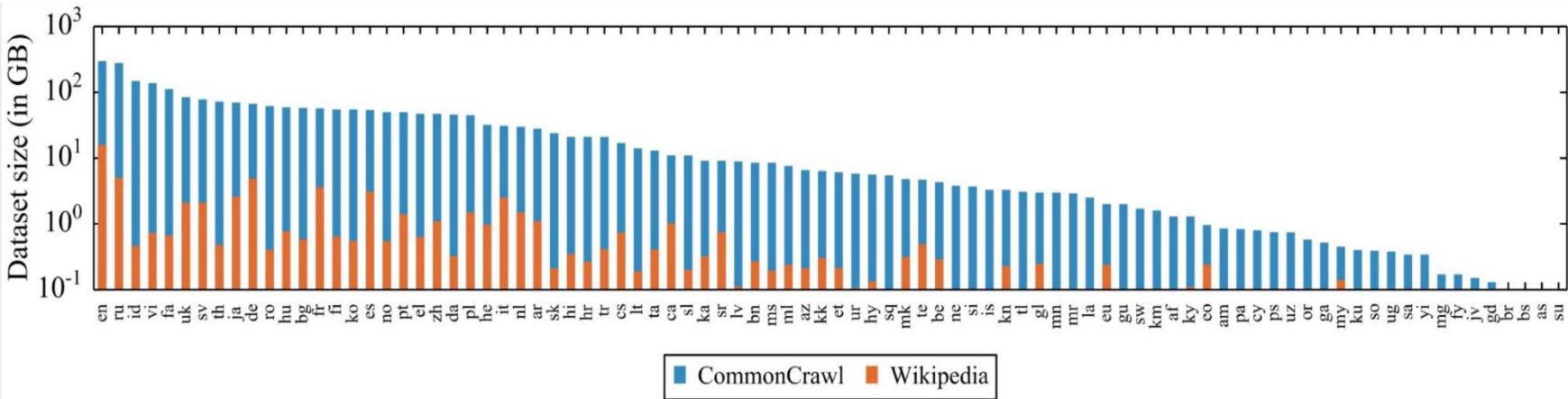
A Islândia foi o primeiro país do mundo a eleger uma mulher como presidente. Vigdís Finnbogadóttir foi eleita em 1980 e governou o país por 16 anos (1980–1996).

This combination – of multitask training and instruction style improves zero shot performance.



Zero shot is particularly helpful for data limited regimes. Why?

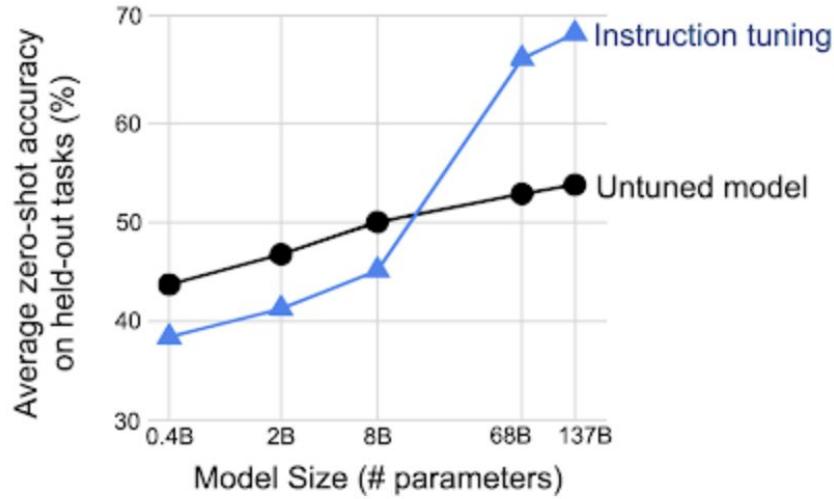
Zero shot is particularly helpful for data limited regimes.



- Data limited regimes struggle to realize gains of fine-tuning.
- Fine-tuning large language models can be expensive – would be great if a model generalized to a task out of the box.

ACL [Keynote](#), Conneau et al.

However, it also requires larger and larger models to take advantage of instruction tuning (partly explaining our race to ever larger models).



Instruction tuning only improves performance on unseen tasks for models of certain size.

It turns out two ingredients have been particularly important at leading to breakthroughs in chatbot performance:

- 1. Structuring multi-task finetuning data as questions and answers.**
- 2. Integrating human feedback about preferences**

Which one do you prefer?

Prompt:

What are the 3 most common gasses in earth's atmosphere?

LM outputs:



A The atmosphere of Earth is a layer of gases retained by Earth's gravity...



B The atmosphere is commonly known as air. The top gases by volume that dry air ...



C The air that surrounds the planet Earth contains various gases. Nitrogen...



D The atmosphere of Earth is the layer of gases, generally known as air...

Human Feedback



B > C = D > A →



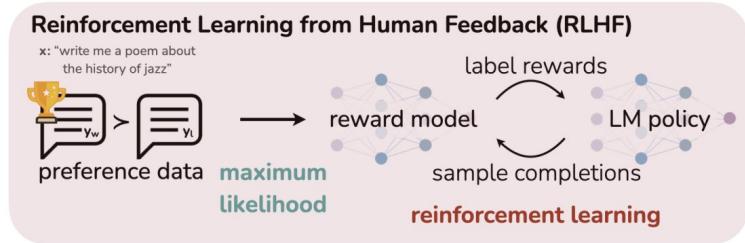
Preference RM

Gather feedback -> train model to align with feedback.

Upweight answers that humans find more meaningful.

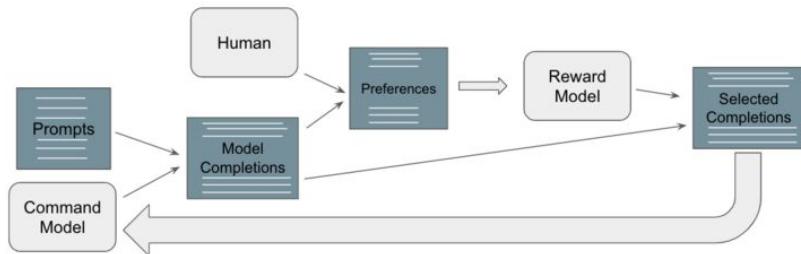
Emerging area of optimization - not a clear winner

w Reward Model:

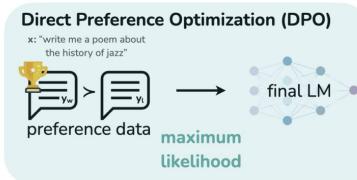


$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} [r_\phi(x, y)] - \beta \mathbb{D}_{\text{KL}}[\pi_\theta(y|x) || \pi_{\text{ref}}(y|x)]$$

w Synthetic & Good only Data



w/o Reward Model:



$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_i) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w | x)}{\pi_{\text{ref}}(y_w | x)} - \beta \log \frac{\pi_\theta(y_i | x)}{\pi_{\text{ref}}(y_i | x)} \right) \right]$$

Sequence Likelihood Calibration from Human Feedback (SLiC-HF)



Margin Triplet Contrastive Loss

$$\mathcal{L}(\theta) = \max(0, \delta - \log P_\theta(\mathbf{y}^+|\mathbf{x}) + \log P_\theta(\mathbf{y}^-|\mathbf{x})) - \lambda \log P_\theta(\mathbf{y}_{\text{ref}}|\mathbf{x})$$

Requires a much smaller sample of annotations, which is then used to optimize for more aligned behavior (often through reinforcement learning objectives).

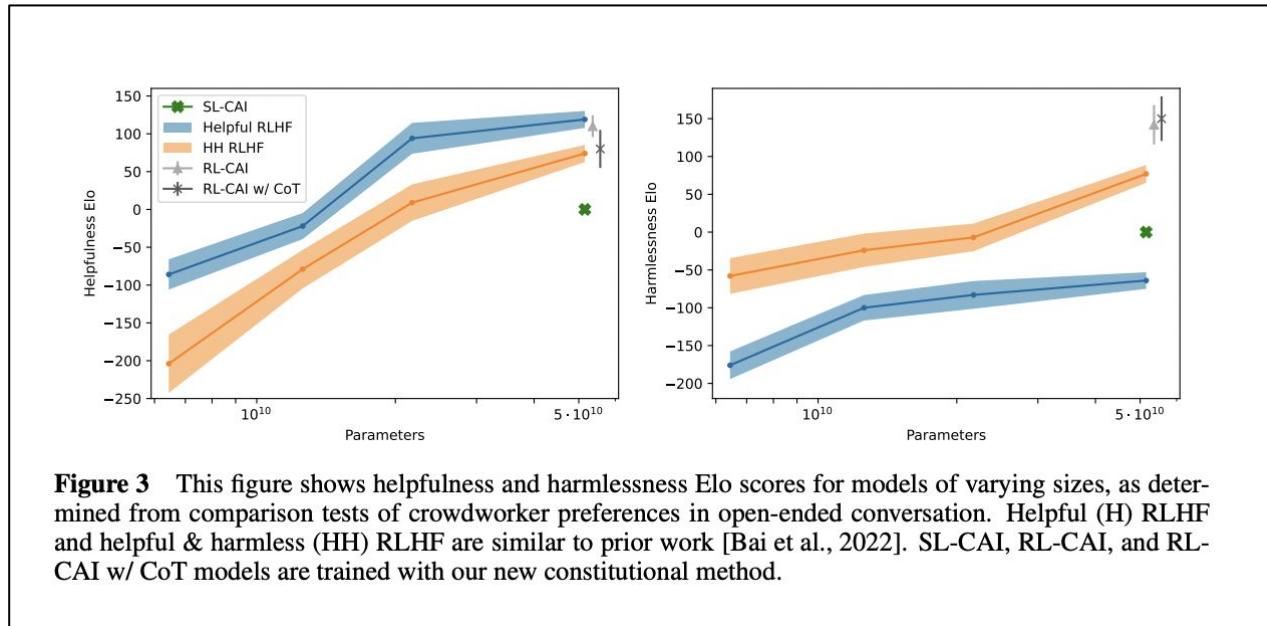
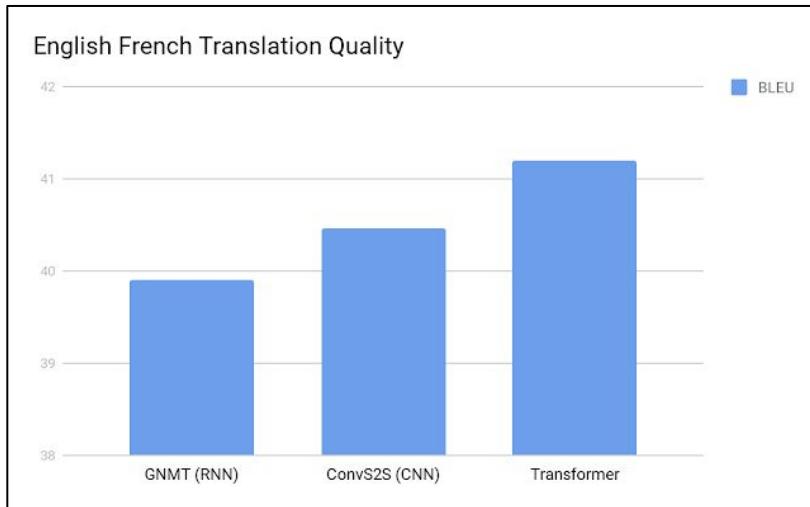


Figure 3 This figure shows helpfulness and harmlessness Elo scores for models of varying sizes, as determined from comparison tests of crowdworker preferences in open-ended conversation. Helpful (H) RLHF and helpful & harmless (HH) RLHF are similar to prior work [Bai et al., 2022]. SL-CAI, RL-CAI, and RL-CAI w/ CoT models are trained with our new constitutional method.

In summary, 2017-2023 has been a rollercoaster ride.

The original transformer was applied to machine learning transformation

2023 – breakthroughs in almost all language tasks



Conversation

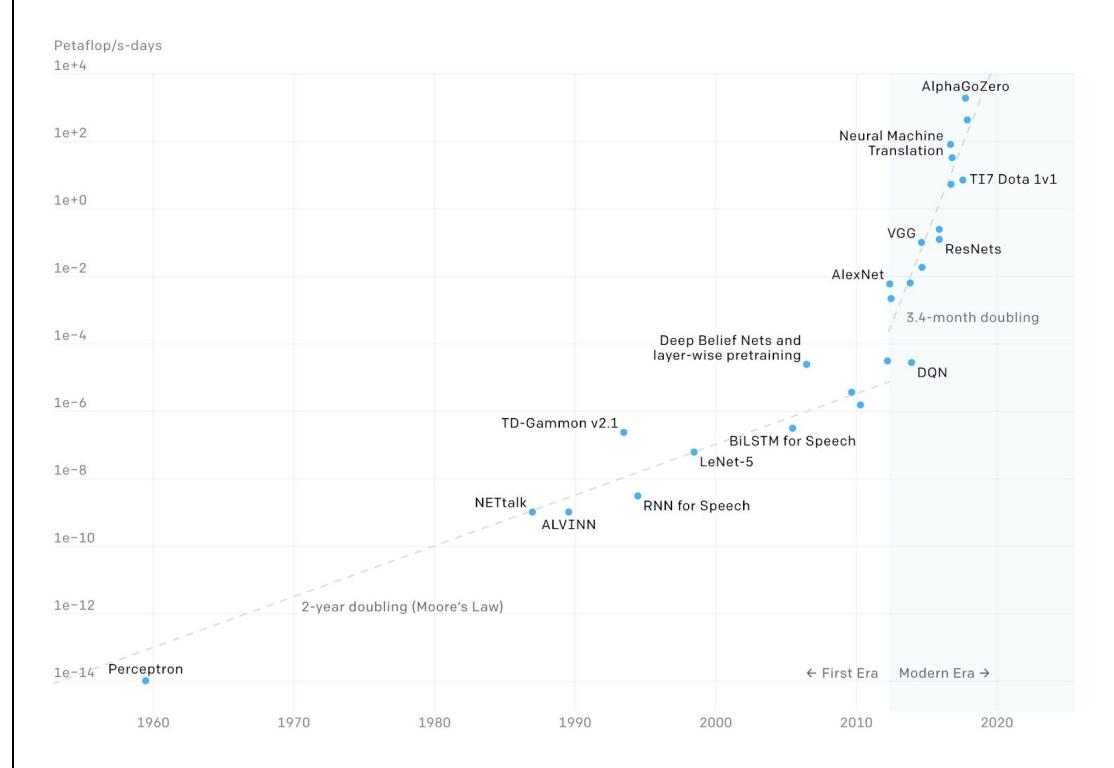
Generation

Classification

So this brings us to now. For the rest of this tutorial I am going to talk about some future open challenges that I think about a lot.

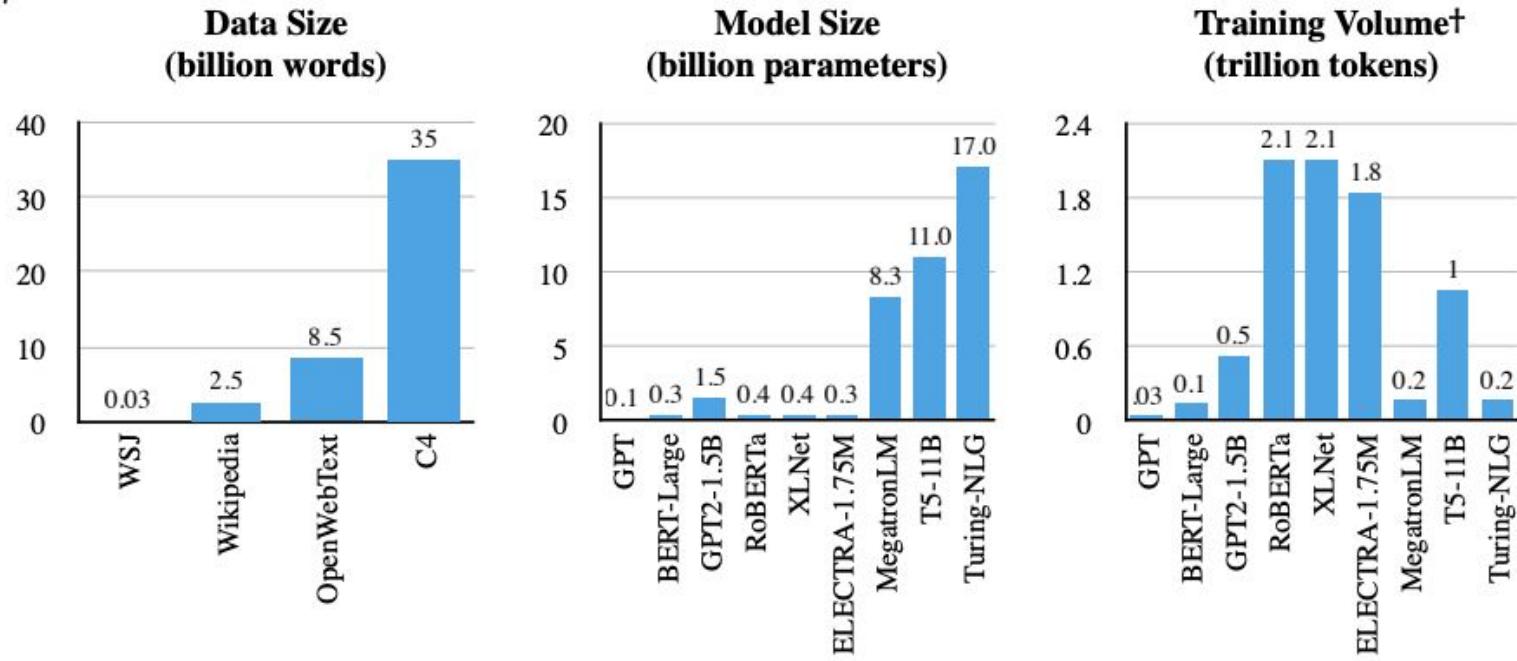
Challenge 1: The renewed
urgency for efficiency.

A rapid growth
in the size of
networks.
“Bigger is
better” race in
the number of
parameters.

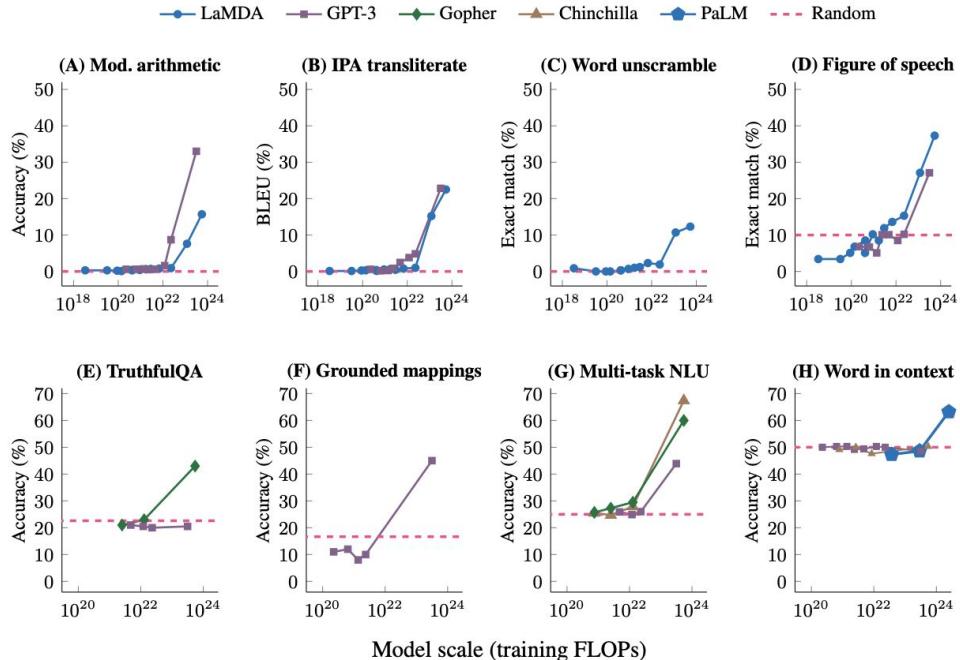


This characterizes both vision and NLP tasks.

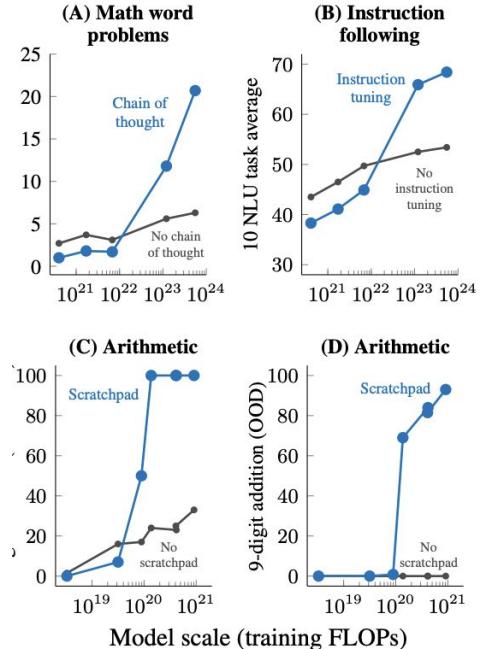
Bird's-eye View



Different regimes of scale appear to induce emergent abilities – model performance which cannot be predicted from smaller models.



Few shot prompting performance



Finetuning and few shot.



A key limitation of this approach:

Relationship between weights and generalization properties is not well understood.

Why do we need so many weights in the first place?

- 1) Diminishing returns to adding more weights.
- 2) Many redundancies between weights
- 3) Intriguing relationship with data quantity and quality.
- 4) We can remove most weights after training.
- 5) Most weights necessary for representing a small fraction of the dataset.

1. Diminishing returns to adding parameters. Millions of parameters are needed to eek out additional gains.

Model	Parameters ^a	Features	Image Size	Paper	ImageNet Top-1 Accuracy		
					Public Checkpoint ^b	I	Public Checkpoint ^b
Inception v1 ^c [69]	5.6M	1024	224	73.2			69.8
BN-Inception ^d [34]	10.2M	1024	224	74.8			74.0
Inception v3 [70]	21.8M	2048	299	78.8			78.0
Inception v4 [68]	41.1M	1536	299	80.0			80.2
Inception-ResNet v2 [68]	54.3M	1536	299	80.1			80.4
ResNet-50 v1 ^e [29, 26, 25]	23.5M	2048	224	76.4			75.2
ResNet-101 v1 [29, 26, 25]	42.5M	2048	224	77.9			76.4
ResNet-152 v1 [29, 26, 25]	58.1M	2048	224	N/A			76.8
DenseNet-121 [31]	7.0M	1024	224	75.0			74.8
DenseNet-169 [31]	12.5M	1024	224	76.2			76.2
DenseNet-201 [31]	18.1M	1024	224	77.4			77.3
MobileNet v1 [30]	3.2M	1024	224	70.6			70.7
MobileNet v2 [61]	2.2M	1280	224	72.0			71.8
MobileNet v2 (1.4) [61]	4.3M	1792	224	74.7			75.0
NASNet-A Mobile [84]	4.2M	1056	224	74.0			74.0
NASNet-A Large [84]	84.7M	4032	331	82.7			82.7

Almost double the amount of weights for a gain in 2% points.

Table: [Kornblith et al., 2018](#) [[Kaplan + 2020](#)]

2. Redundancies Between Weights

Predicting Parameters in Deep Learning

Misha Denil¹ Babak Shakibi² Laurent Dinh³
Marc'Aurelio Ranzato⁴ Nando de Freitas^{1,2}

¹University of Oxford, United Kingdom

²University of British Columbia, Canada

³Université de Montréal, Canada

⁴Facebook Inc., USA

{misha.denil,nando.de.freitas}@cs.ox.ac.uk
laurent.dinh@umontreal.ca
ranzato@fb.com

Abstract

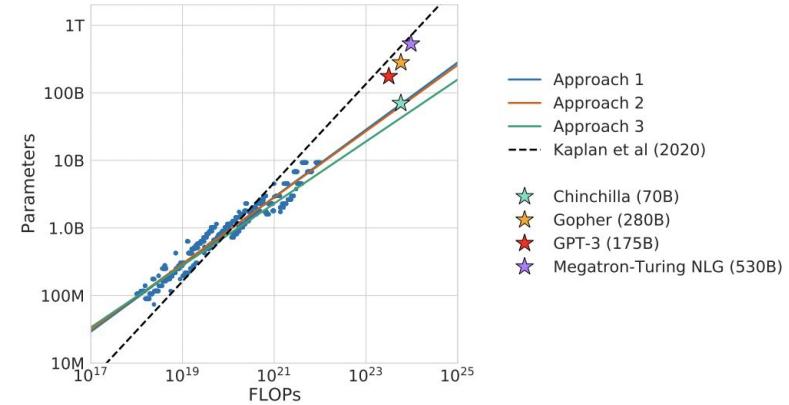
We demonstrate that there is significant redundancy in the parameterization of several deep learning models. Given only a few weight values for each feature it is possible to accurately predict the remaining values. Moreover, we show that not only can the parameter values be predicted, but many of them need not be learned at all. We train several different architectures by learning only a small number of weights and predicting the rest. In the best case we are able to predict more than 95% of the weights of a network without any drop in accuracy.

Denil et al. find that a small set of weights can be used to predict 95% of weights in the network.

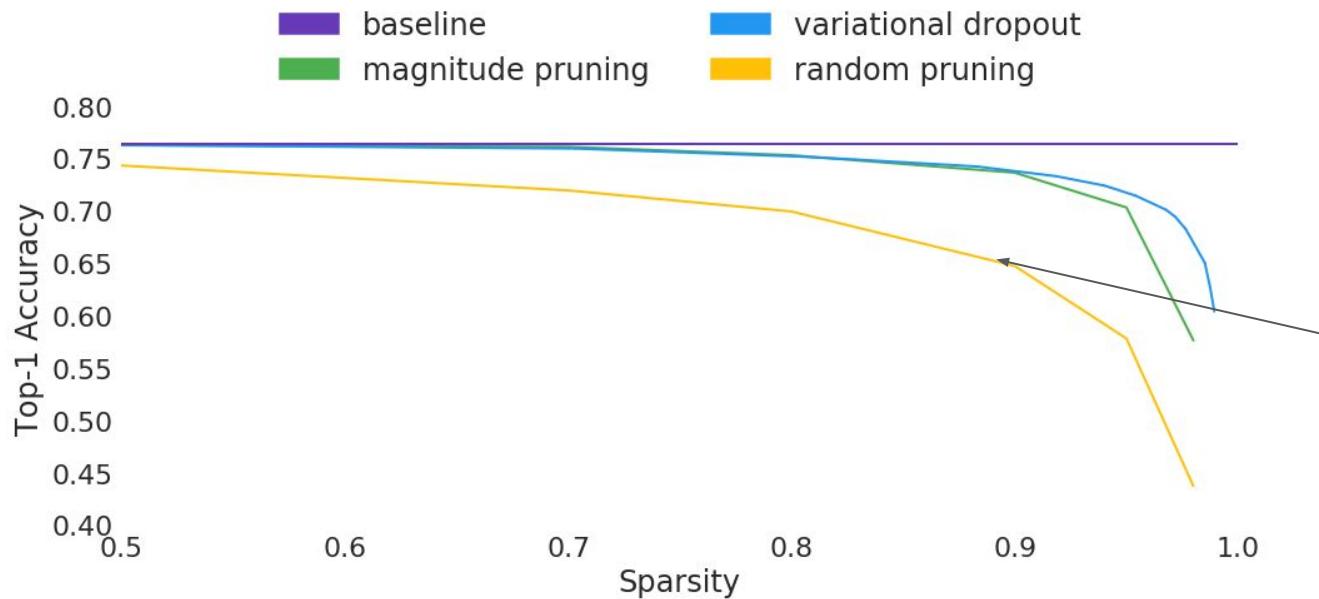
3. Intriguing relationship between weights and data.

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
<i>Chinchilla</i>	70 Billion	1.4 Trillion

- Recent work suggests smaller amounts of higher quality data remove the need for a larger model.
- This suggest larger models may just be compensating for problems in the data pipeline.

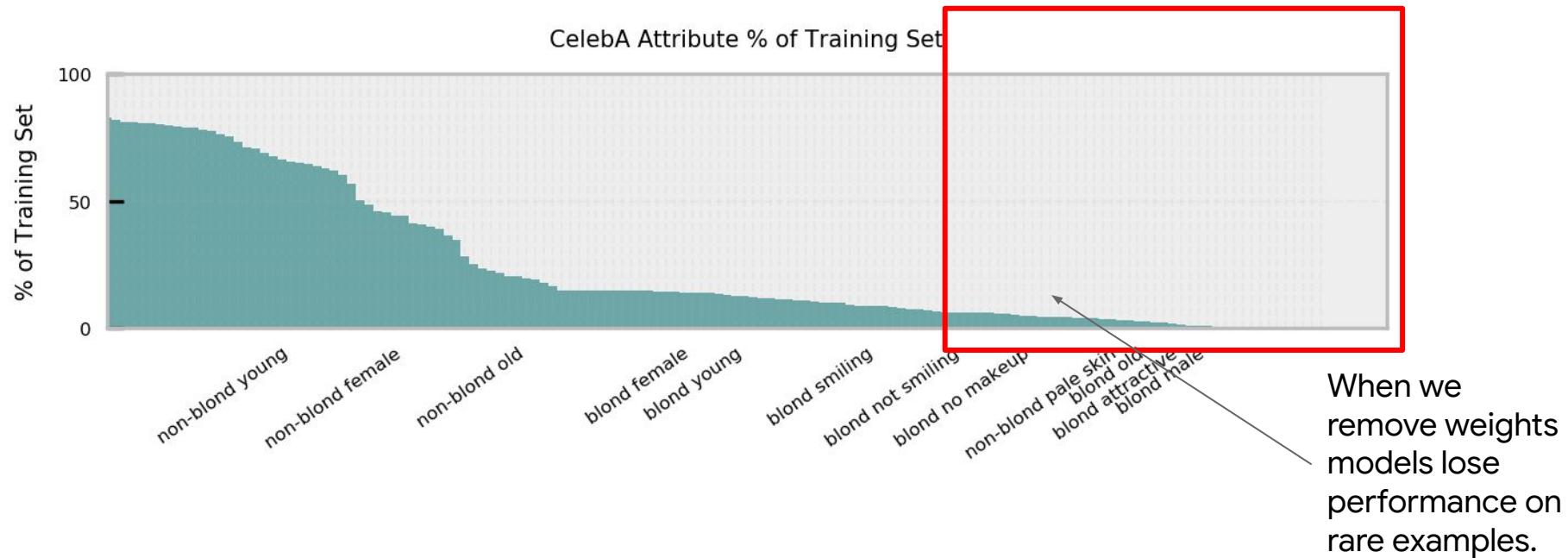


4. Most weights can be removed after training is finished (**while only losing a few % in test-set accuracy!**)



With 90% of the weights removed, a ResNet-50 only loses ~3% of performance (for certain pruning methods).

5. Across a variety of settings and modalities, we find that removing weights causes models to loss performance on the long-tail. The majority of weights (**90% of all weights**) are used to memorize very rare examples in the dataset.



Motivation: Understanding how varying model size impacts generalization is an increasingly urgent question:

How do generalization properties change as models get bigger and bigger?

- How does this impact responsible deployment?
Fairness, robustness, privacy.

Why do we need so many parameters to begin with?

- Deep neural networks appear to be an incredibly inefficient representation
- If most weights are redundant, why do we need them in the first place?
- Can these insights guide us to better optimization protocols?

Point of comparison: our Brain is incredibly energy efficient.

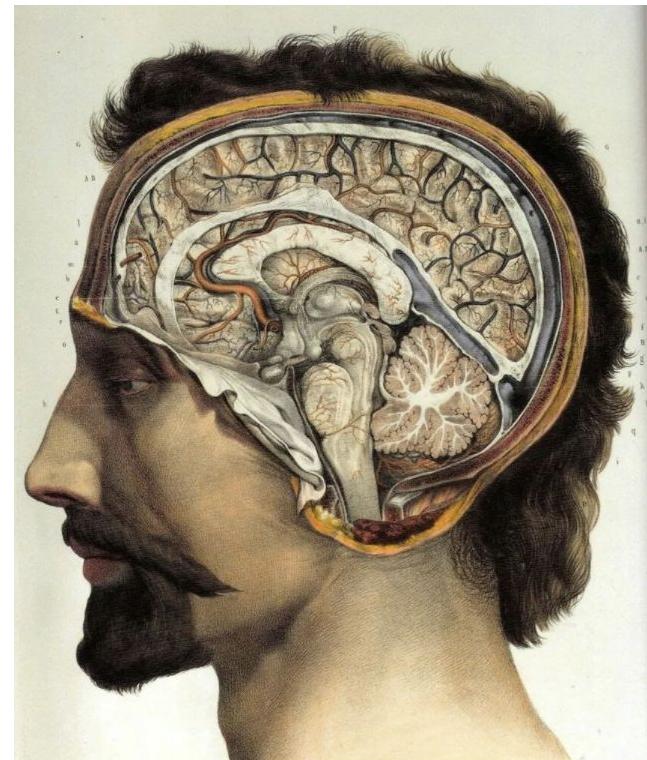
Has over 85 billion neurons but runs on the energy equivalent of an electric shaver

Key design choices to embed efficiency:

Specialized pathways

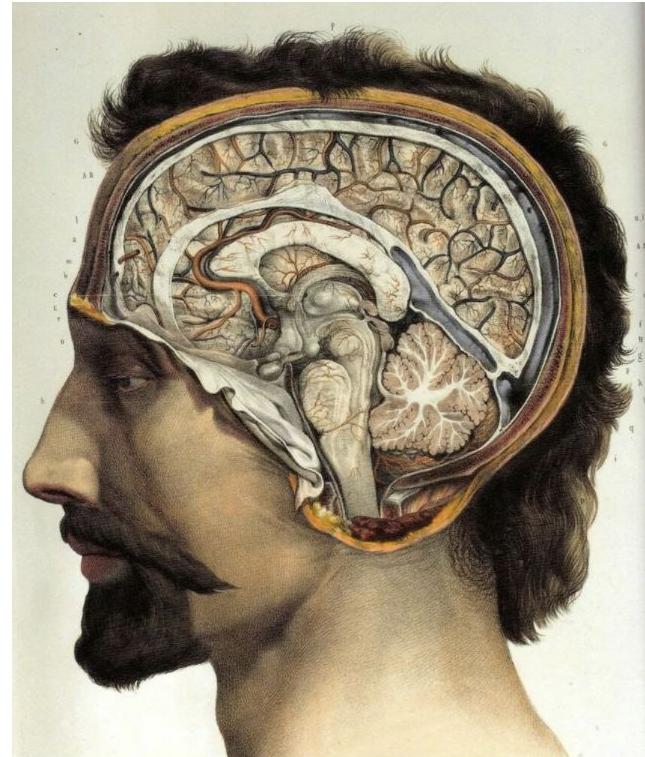
Simulate much of what we “see”

Log scale vision



Some aspects of what we do with deep neural networks is painfully inefficient.

- We do not have adaptive compute.
Typically we see all examples same amount of time during training.
- Global updates mean all prior information is erased.
- Empirical risk minimization means while we optimize for average performance, it takes considerable more compute to model rare or infrequent artefacts.



A lot of my own research has focused on efficiency (both data and weights). So, I will share some quick observations on open questions in this field.

The State of Sparsity in Deep Neural Networks

Trevor Gale,^{1,2} Erich Eiben,³ and Sara Hooker^{1,2}

Abstract

We rigorously evaluate three state-of-the-art techniques for inducing sparsity in deep neural networks. We compare the performance of a sparse model trained on WMT 2014 English-to-German and French-to-English translation datasets against baseline models trained with standard gradient descent optimization techniques (Mikolov et al., 2017). Louison et al. (2018) propose a sparse model that uses a sparse mask on random dataset parameters, incrementally, to reduce the number of active neurons. Our experiments show that this approach can achieve comparable or better results. Based on insights from our experiments, we propose a new state-of-the-art of the sparsity-accuracy tradeoff that is competitive with the state-of-the-art. Interestingly, we repeat the experiments performed by Frankle & Carlini (2018) and Liang et al. (2018) to assess the robustness of sparse models. Our experiments indicate that the sparse models are more robust than their dense counterparts. In addition, we present two novel metrics to measure the robustness of sparse models. These metrics are based on the number of neurons that are active during adversarial attacks. The first metric, called the neuron robustness score, measures the robustness of a model trained with sparse operations and provides a quantitative measure of the robustness of sparse models. The second metric, called the neuron robustness ratio, measures the robustness of a model trained with sparse operations and provides a quantitative measure of the robustness of sparse models.

like image classification and machine translation currently have tens of millions of parameters, and require billions of floating-point operations to make a prediction for a single input image.

Sparsety has emerged as a leading approach to address the curse of dimensionality in deep learning. One reason is that the model's parameters have a value of exactly zero. With zero-valued weights, any multiplication (which dominates computation in neural networks) can be avoided. This property can be used to compress and represent sparse neural networks. Sparsety has also been used to reduce the number of connections networks to tolerate high levels of sparsity (Yuan et al., 2017). In addition, the use of sparse models in deep learning has been leveraged to significantly reduce the cost of training and inference (Krause et al., 2018; Gao et al., 2018), and to enable the deployment of state-of-the-art models in mobile devices and resource-constrained environments (Deng et al., 2018; Kang et al., 2018; Liu et al., 2018).

Over the past few years, numerous techniques for inducing sparsity in deep neural networks have been proposed. Most datasets used as benchmarks have grown so large to also usually expect new approaches to explain them all. In addition, the use of sparse models in deep learning has been leveraged to reduce the cost of training and inference (Krause et al., 2018; Gao et al., 2018), and to enable the deployment of state-of-the-art models in mobile devices and resource-constrained environments (Deng et al., 2018; Kang et al., 2018; Liu et al., 2018).

Niell Teng Hu¹ Xinyu Hu² Rosanne Liu^{1,3} Sara Hooker¹ Jason Yosinski¹

Abstract

Not all examples are created equal, but standard deep neural network training protocols treat each training point uniformly. Each example is propagated forward and backwards through the network the same amount of times, independent of how much it improved the model during training protocol. Recent work has proposed ways to accelerate training by deviating from this uniform treatment. Popular methods entail up-weighting examples that improve the model, or down-weighting the intuition that examples with low loss have already learned by the model, so their marginal value is negligible. This view assumes that updating the model with

propagating each forward and backwards through the network the same amount of times results in redundancy and inefficient use of training budget.

Recent work proposes accelerating training by testing data examples differently. Jiang et al. (2019) and Katharopoulos & Fleuret (2018) have proposed loss-based sampling methods that skip deep neural network layers by emphasizing the high loss examples. Although such methods achieve significant reductions, the merits of each have not been evaluated on understanding the mechanics of real world data. In large scale training settings, models are often trained on datasets with unknown quality and degree of input or label corruption (Tigpen et al., 2016; Hooker et al., 2020; Beyer

WHEN LESS IS MORE: SIMPLIFYING INPUTS AIDS NEURAL
NETWORK UNDERSTANDING

Robin Tibor Schirrmeyer
University Medical Center Freiburg
ML Collective
robin.schirrmeyer@uniklinik-freiburg.de

Tomio Ball
University Medical Center Freiburg
tonio.ball@uniklinik-freiburg.de

Sara Hooker
Google Brain
ML Collective
shooker@google.com

Rosanne Liu
Google Brain
ML Collective
rosanneliu@google.com

Studying the impact of magnitude pruning on contrastive learning methods

Francesco Corti^{*}, Rahim Entezari^{*}, Sara Hooker², Davide Bacciu¹, Olga Sushk²

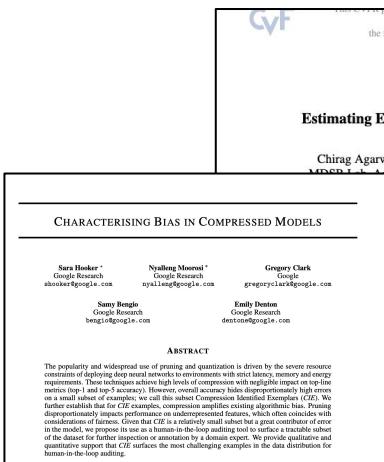
Abstract

We study the impact of different pruning techniques on the representation learned by deep neural networks trained with contrastive loss functions. Our work finds that at high sparsity levels, contrastive learning results in a better quality of learned representations compared to models trained with traditional cross-entropy loss. To understand this pronounced difference, we use metrics from the field of PDEs (Kazhdan et al., 2010; Qi & Su, 2013) and PDE Score (Kahabka et al., 2022) and PD-Score (Baldari et al., 2021) to measure the impact of pruning on the learned representation quality. Our analysis suggests that the learned representation quality is negatively impacted by pruning. We find that the negative impact of sparsity on the quality of the learned representation is the highest when pruning is introduced early in training phase.

(a) Representation by supervised training without pruning (left), pruned 95% by GMP (center) and One-Shot (right).

(b) Representation by supervised training without pruning (left), pruned 95% by GMP (center) and One-Shot (right).

Figure 2: Impact of pruning on the learned representations. UMAP diagrams of the models trained with supervised (top row) and unsupervised (bottom row) learning rate decay. The columns show the learned representations for top row: supervised training without pruning, pruned 95% by GMP, and One-Shot. The columns show the learned representations for bottom row: unsupervised learning with learning rate decay, pruned 95% by GMP, and One-Shot. The axes in the plots represent learned features.



KEEP THE GRADIENTS FLOWING: USING GRADIENT FLOW TO STUDY SPARSE NETWORK OPTIMIZATION

Kale-ab Tesser^{1,3}, Sara Hooker², and Benjamin Rosman¹

¹School of Computer Science and Applied Mathematics, University of the Witwatersrand
²Google Brain
³instacDeep
kalebtesser@gmail.com, shookert@google.com, Benjamin.Rosman@wits.ac.za

ABSTRACT

Training sparse networks to converge to the same performance as dense neural architectures has proved to be a elusive. Recently, algorithms that initialize weights have been proposed to train sparse networks. However,

The figure consists of two side-by-side line plots. Both plots have 'Number of training examples' on the x-axis and 'NER F1 score' on the y-axis, ranging from 0.0 to 1.0. The left plot is titled 'Supervised learning' and shows a single blue line that starts at approximately 0.25 F1 score with 1 example and rises steadily to about 0.85 F1 score with 100,000 examples. The right plot is titled 'Supervised learning + few-shot learning' and shows two blue lines. One line follows a similar trajectory to the left plot, starting at ~0.25 F1 with 1 example and reaching ~0.85 F1 with 100,000 examples. The second line, which starts at the same point, remains flat at a low F1 score (around 0.1) until approximately 10,000 examples, after which it begins to rise slowly, reaching ~0.2 F1 at 100,000 examples.

The Low-Resource Double Bind: An Empirical Study of Pruning for Low-Resource Machine Translation

Orevaoghene Ahia
Masakhane NLP
oreva.ahia@gmail.com

Julia Kreutzer
Google Research
Masakhane NLP
j.kreutzer@google.com

Sara Hooker
Google Research, Brain
shooker@google.com

A Tale Of Two Long Tails

Daniel D'wozniak¹², Zach Nusbaum¹, Chirag Agrawal¹, Sara Hooker¹

Abstract

As machine learning models are increasingly employed to assist humans in the uncertainty associated with model predictions, how can we ensure that these models are not only focused on traditional probabilities or ranking approaches, but also capable of handling low resource or uncertain examples. While this captures what examples are challenging for machine learning models, it does not capture the source of the uncertainty. In this work, we seek to understand the relationship between the source of uncertainty and the source of uncertainty. We explore benefits of designing a targeted interpretation system that can identify the source of uncertainty and characterize the source of said uncertainty.

Figure 1 Examples of different types of uncertainty. Left: A horse image misclassified as a donkey due to missing context. Right: A horse image misclassified as a donkey due to a lack of data. The horse image on the right is correctly classified as a horse even with more examples because of the high quality of the image.

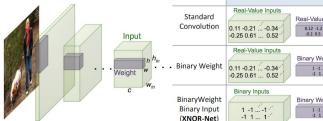
Figure 1 Examples of different types of uncertainty. Left: A horse image misclassified as a donkey due to missing context. Right: A horse image misclassified as a donkey due to a lack of data. The horse image on the right is correctly classified as a horse even with more examples because of the high quality of the image.

Overview of common open
directions in ML efficiency
research.

There are a few key directions in compression techniques:

Quantization -

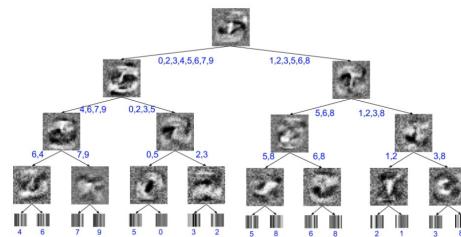
approximating a deep neural network that uses floating point numbers with one that uses low bit width numbers.



Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet [AlexNet]
Standard Convolution	Real-Value Inputs 0.11-0.21 0.25-0.61	Real-Value Weights 0.02 0.02	+ , - , \times 1x	1x %56.7
Binary Weight	Real-Value Inputs 0.11-0.21 0.25-0.61	Binary Weights 1 1 1	+ , - \sim 32x	\sim 2x %56.8
Binary-Weight Binary Input (XNOR-Net)	Binary-Weights Binary Input (XNOR-Net) 0.11-0.21 0.25-0.61	Binary Weights 1 1 1	XNOR , bitcount \sim 32x	\sim 58x %44.2

[[[Rastegari et al.](#), 2016,
[Gong et al.](#), 2014,
[Courbariaux et al.](#), 2016]]

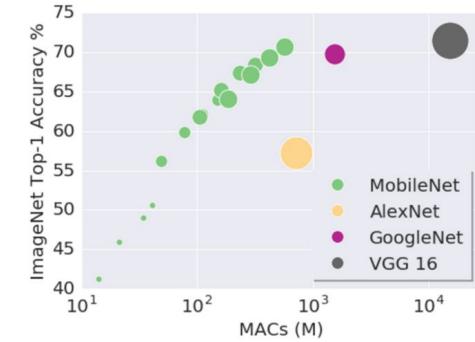
Model distillation - have a teacher “larger” network distill knowledge to a student network.



[[[Frosst and Hinton](#), 2017,
[Hinton et al.](#), 2015]].

Architecture design -

train an efficient architecture from scratch.

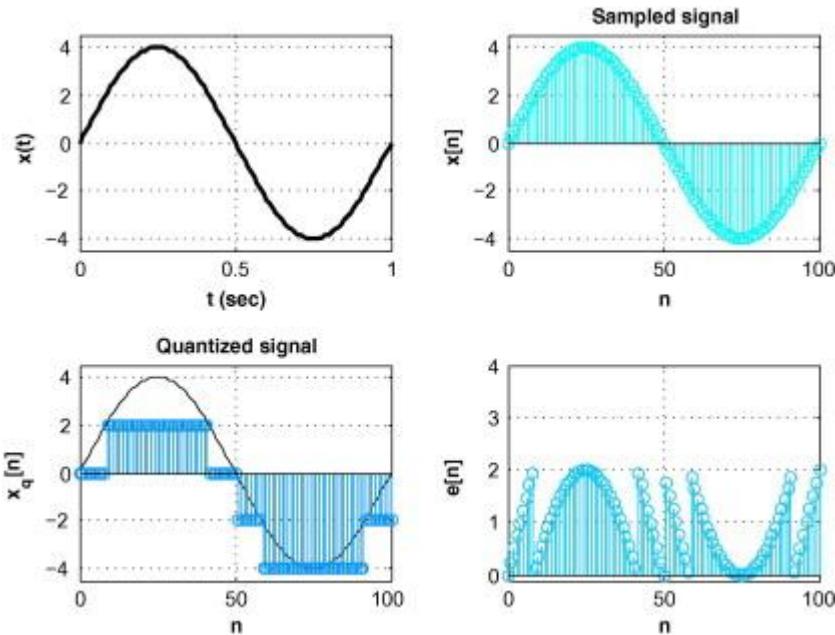


[[[MobileNet](#),
[Howard et al.](#), 2017,
[SqueezeNet](#),
[Iandola et al.](#), 2016]]

Quantization

Recent progress has considered different quantization strategies:

- For training or finetuning
- Mixed bit quantization
(different bits for backward pass).
- Adaptive quantization.



Quantization 101

CPU/GPU memory



CPU/GPU computation

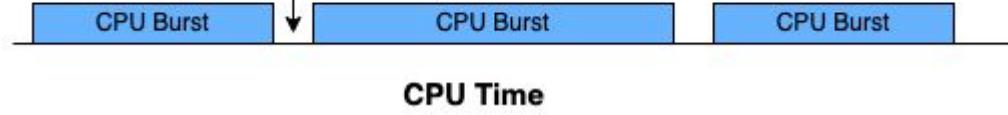


I/O - data movement



Long CPU Bursts

Few I/O operations



Quantization 101

Quantization: reduce the amount of data to store, compute and move



Long CPU Bursts

CPU Burst

Few I/O operations

CPU Time



CPU Burst

Quantizing at scale has been considered very challenging.

After 6 billion parameters, many works [[[Dettmers et al., 2022](#), [Guangxuan et al., 2022](#), [Bondarenko et al. 2021](#)]] report extreme sensitivity to quantization.

Dettmers et al. refers to this as inherent to an “emergent property.”

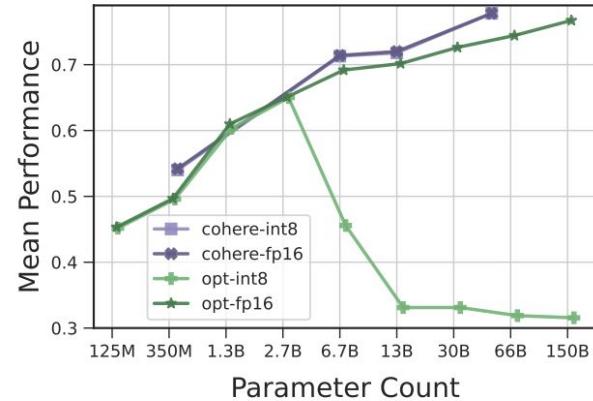


Figure 1: Mean zero-shot accuracy on HellaSwag, PIQA, LAMBADA, and WinoGrad. In contrast to the OPT family, our models show minimal degradation after simple vectorwise quantization. Data points for OPT models are from (Dettmers & Zettlemoyer, 2022).

Our recent work: Are quantization cliffs in performance solely a factor of scale?

Are emergent properties inherent (due to nature or nurture?). In our recent work, we find that optimization strategy matters and cliffs in performance can be avoided by using the right pre-training strategy.

Intriguing Properties of Quantization at Scale

Arash Ahmadian^{*†}
Cohere For AI

Bharat Venkitesh
Cohere

Ahmet Üstün
Cohere For AI

Saurabh Dash^{*}
Cohere

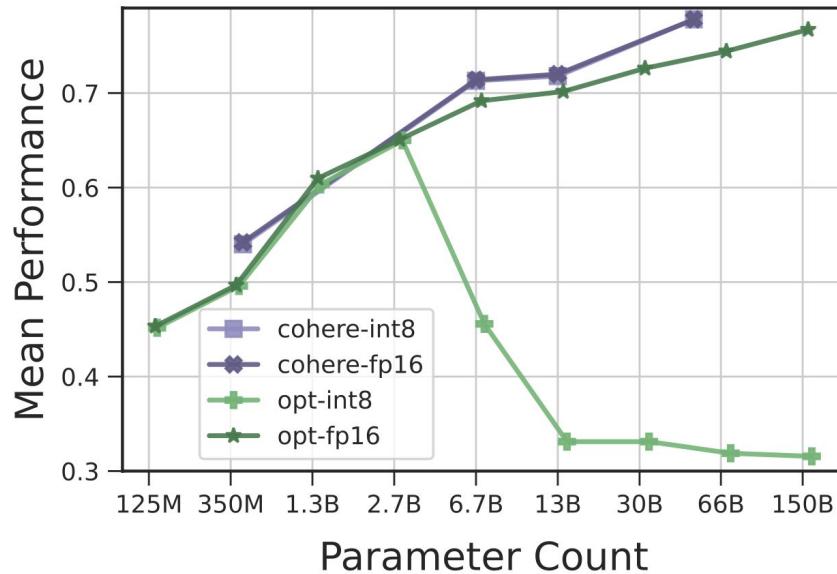
Stephen Gou
Cohere

Phil Blunsom
Cohere

Sara Hooker
Cohere For AI

{arash,saurabh,charlie,bharat,stephen,phil,ahmet,sarahooker}@cohere.com

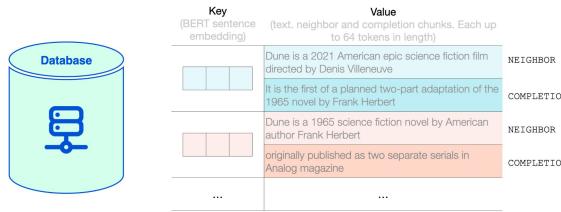
We can avoid cliffs in quantization performance using careful optimization choices.



This both opens up directions for more efficient quantization, and poses the question: are other emergent properties inherent or can be altered and conditioned by optimization and architecture design choices

Architecture optimizations - building a more efficient architecture from scratch.

Models w built-in retrieval databases



[Borgeaud et al. 2021](#)

Adaptors

Adapters	AF vs. Full FT		AF vs. Adapter	
	Training	Inference	Training	Inference
2	0.92	0.64	0.57	0.68
8	0.53	0.38	0.33	0.40
16	0.33	0.24	0.21	0.26

Table 3: Relative speed of AdapterFusion (with 2/8/16 adapters) compared to a fully fine-tuned model and compared to a single-task adapter (right). Measured with a batch size of 32, and a sequence length of 128.

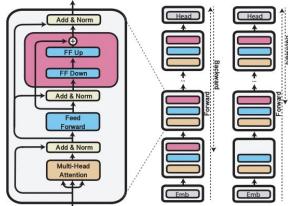
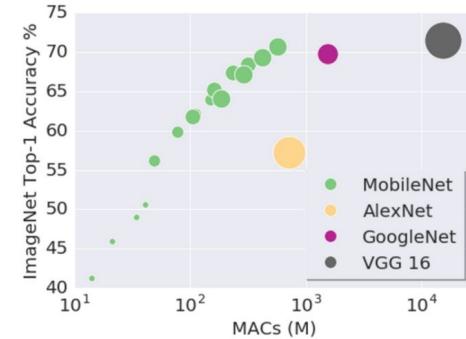


Figure 1: Standard adapter fine-tuning vs. AdapterDrop fine-tuning. The left model includes adapters at every layer whereas the right model has adapters dropped at the first layer. The arrows to the right of each model indicate the information flow for the *Forward* and *Backward* pass through the model.

[Rücklé et al. 2021](#),
[Pfeiffer et al. 2021](#)

Specialized convolutions



[\[\[MobileNet, Howard et al. 2017, SqueezeNet, Iandola et al. 2016\]\]](#)

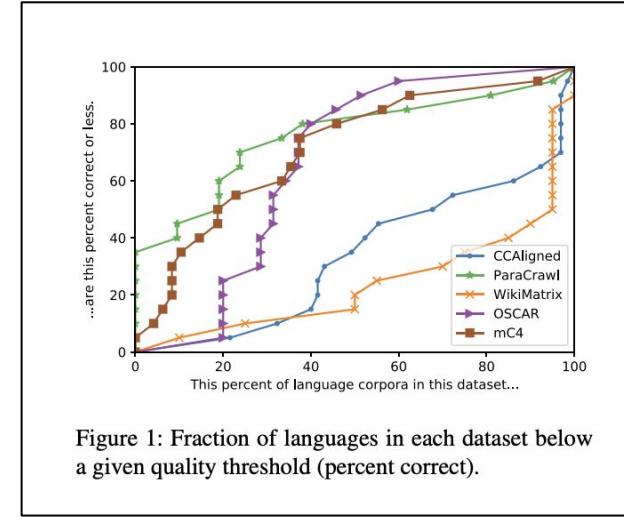
Pruning in the data space.

There is increasing evidence that efforts to better curate training corpus, including **deduping, pruning data and increasing the available training corpus size** can compensate for the need for larger networks and/or improve training dynamics.

	% train examples with dup in train	% valid with dup in train
C4	3.04%	1.59%
RealNews	13.63%	1.25%
LM1B	4.86%	0.07%
Wiki40B	0.39%	0.26%

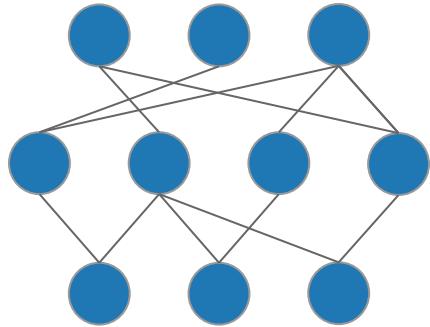
Table 2: The fraction of examples identified by NEARDUP as near-duplicates.

[Lee et al. 2022](#)



[Kreutzer et al. 2022](#)

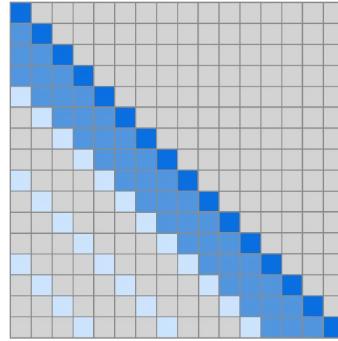
Sparsity in Deep Learning



Weight Sparsity

Sources: Pruning, sparse training

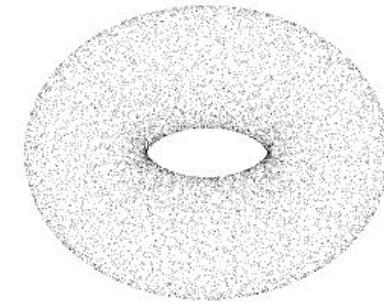
Example: 3x5x FLOP advantage for the same accuracy in CNNs [1][2]



Activation Sparsity

Sources: ReLU sparsity, sparse attention

Example: Asymptotic improvement in attention computational complexity. From $O(N^2)$ to $O(N * \sqrt{N})$ to $O(N)$ [3][4]



Data Sparsity

Sources: Point clouds, graphs, etc.

Example: 3D object detection with targeted computation [5]

Optimization procedures for sparse neural networks:

Dense to Sparse

Model is dense during training, sparse during inference.



Most research has focused here to-date.

Sparse only at inference time. Limits the largest sparse network you can train by the largest dense you can fit into memory.

Sparse to Sparse

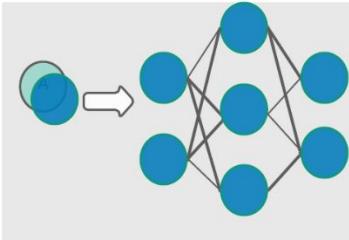
Model is sparse during both training and inference.



Growing as an important direction of research.
More challenging problem.

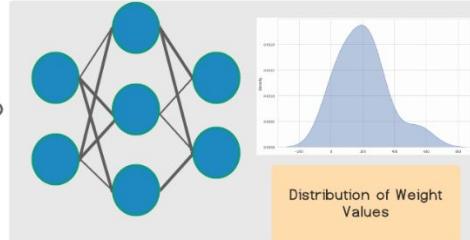
Dense-to-sparse training of deep neural networks typically involves 3 separate stages.

Step 1: Train a large, overparameterized network for a certain number of steps.



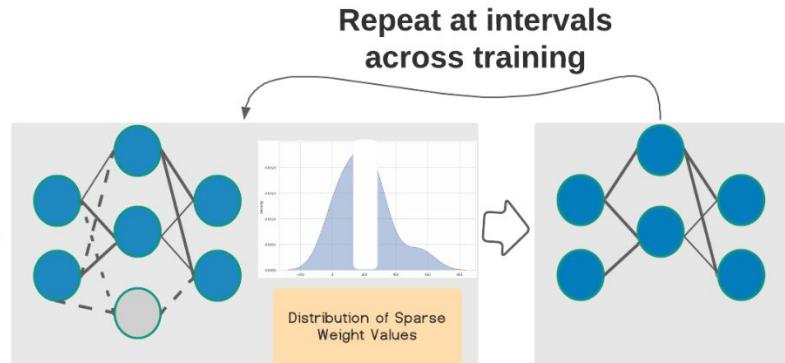
Initialize All Weights

Step 2:
Remove a subset of weights by setting to zero.

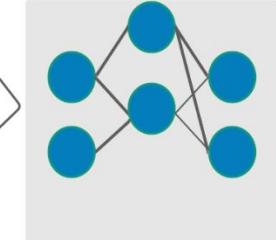


Training of Weights

Step 3: Continue to train the smaller network for a certain amount of steps to **recoup** the lost performance.



Prune weights estimated to be unimportant



Retrain to calibrate remaining weights

Dense-to-sparse training in deep neural networks typically involves 3 separate stages.

Step 1: Train a large, overparameterized network for a certain number of steps.

Step 2: Remove a subset of weights/neurons by setting to zero.

Step 3: Continue to train the smaller network for a certain amount of steps to **recoup** the lost performance.

Estimation problem - accurately estimate which weights and neurons are most important to the model.

Re-calibration problem - how should the matrix be recalibrated to account for the removal of a subset of values.

Divide between theoretical and applied motivations for studying efficiency.

Inference and training improvements in efficiency may be very different in formulation.

This is because the pain points are different.

Training:

Cost of the backward pass.

Differences in choice of optimizer, data/model parallelization can make a large difference.

Inference and training improvements in efficiency may be very different in formulation.

This is because the pain points are different.

Training:

Cost of the backward pass.

Differences in choice of optimizer, data/model parallelization can make a large difference.

Key constraint: Managing device memory and distribution overhead.

Inference and training improvements in efficiency may be very different in formulation.

This is because the pain points are different.

Training:

Cost of the backward pass.
Differences in choice of optimizer, data/model parallelization can make a large difference.

Key constraint: Managing device memory and distribution overhead.

Inference:

Partitioning of variables across devices, data and device parallelization, batching strategies.

Inference and training improvements in efficiency may be very different in formulation.

This is because the pain points are different.

Training:

Cost of the backward pass.
Differences in choice of optimizer, data/model parallelization can make a large difference.

Key constraint: Managing device memory and distribution overhead.

Inference:

Partitioning of variables across devices, data and device parallelization, batching strategies.

Key constraint: Latency (seconds per forward pass) /high throughput (examples (or tokens) per second)

Unstructured sparsity achieves far high levels of compression but incompatible with current hardware.

- Some patterns feasible (2:4 sparsity nvidia), upcoming hardware design likely to support 1:4 block sparsity.
- Bottleneck for interesting directions such as dynamically growing sparse networks [[Frankle et al. 2021](#), [Utku et al. 2022](#)].

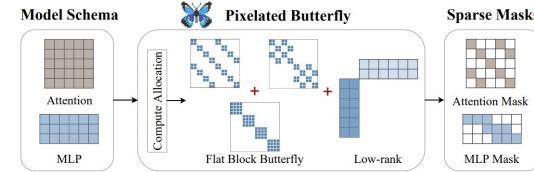


Figure 1: Pixelly targets GEMM-based networks (networks whose computation is dominated by matrix multiply), which it views as a series of matrix multiplication. For each matrix multiply from Model Schema, it (1) allocates compute budget based on dimension and layer type, (2) the budget decides a mapping (hyper-parameter) to our proposed flat block butterfly sparsity patterns, (3) outputs a hardware-aware sparse mask. Note since the hardware is a block device, one memory access to an element in a block leads to the access to the full block.

[Dao et al. 2022](#)

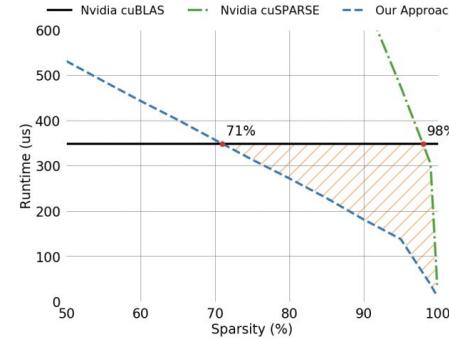


Fig. 1. Sparse matrix-matrix multiplication runtime for a weight-sparse long short-term memory network problem. Input size 8192, hidden size 2048, and batch size 128 in single-precision on an Nvidia V100 GPU with CUDA 10.1. Using our approach, sparse computation exceeds the performance of dense at as low as 71% sparsity. Existing vendor libraries require 14× fewer non-zeros to achieve the same performance. This work enables speedups for all problems in the highlighted region.

[Gale et al. 2022](#)

FLOPs (floating-point multiplication-and-addition operations) are often a misleading metric of efficiency.

FLOPs are often preferred as a metric due to being hardware agnostic.

However, do not capture the sequential dependencies of operations (which operations can be parallelized) and memory accesses in the model (how operations are allocated [[GPU/CPU]]).

Minimizing FLOPs does not necessarily translate to improvements in latency/high throughput.

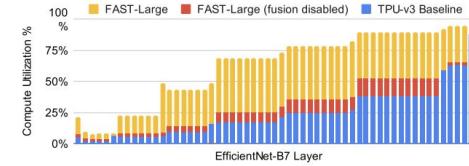


Figure 14: FAST-Large EfficientNet-B7 per-layer performance as a fraction of peak FLOPS. Changing from TPU-v3's 128x128 systolic arrays to FAST-Large's 32x32 systolic arrays improves compute utilization, but remains bottlenecked by memory bandwidth until FAST fusion is enabled.

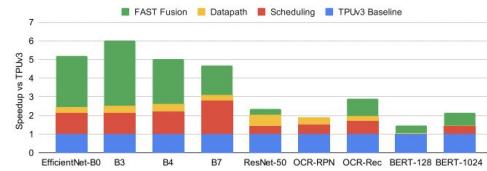


Figure 15: Performance breakdown of each component of FAST relative to a modeled TPU-v3 single core baseline. Improvements are additive; for example, FAST fusion includes both datapath and scheduling improvements.

The largest EfficientNet model, B7, represents a worst-case scenario for fusion, but FAST fusion can still achieve sufficiently high operational intensity at batch size 8 to overcome the memory bottleneck.

Case study: Efficientnet, mobile-net depthwise-separable convolution

[Zhang et al. 2022](#)

Early exiting strategies are really interesting adaptive strategy. However, often incompatible with software stack.

Large scale models are often memory bound at inference. Early exiting has no impact on memory requirements when using software stacks that fully instantiate the computation graph prior to running the program (i.e TensorFlow).

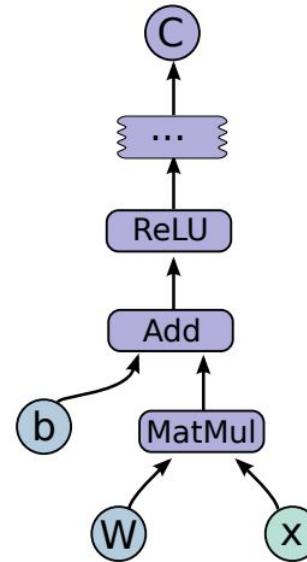


Figure 2: Corresponding computation graph for Figure 1

[[[Abadi et al. 2016](#)]]

A key challenge of efficiency in larger models is handling complicated data transfer and gradient accumulation among distributed hardware.

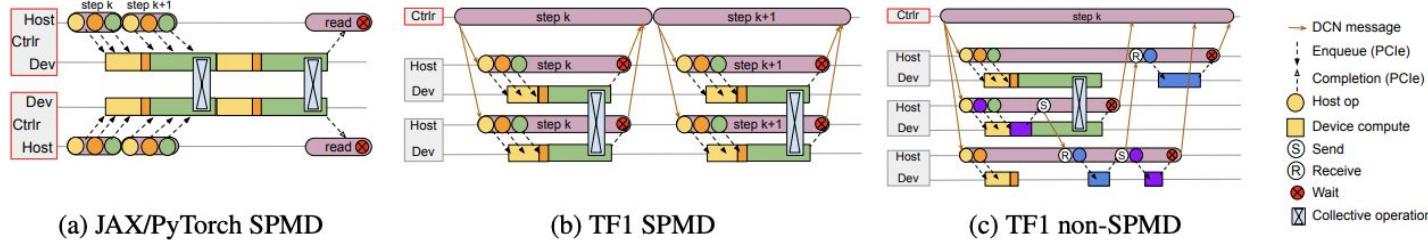


Figure 1. Comparison of dispatch overheads and communication patterns between multi-controller and single-controller systems. (a) Jax or PyTorch SPMD independently enqueues accelerator computations asynchronously over fast PCIe; (b) TensorFlow v1 SPMD requires control messages over slower DCN; (c) TensorFlow v1 non-SPMD programs require cross-host coordination or data transfer through explicit send (S) and recv (R) ops.

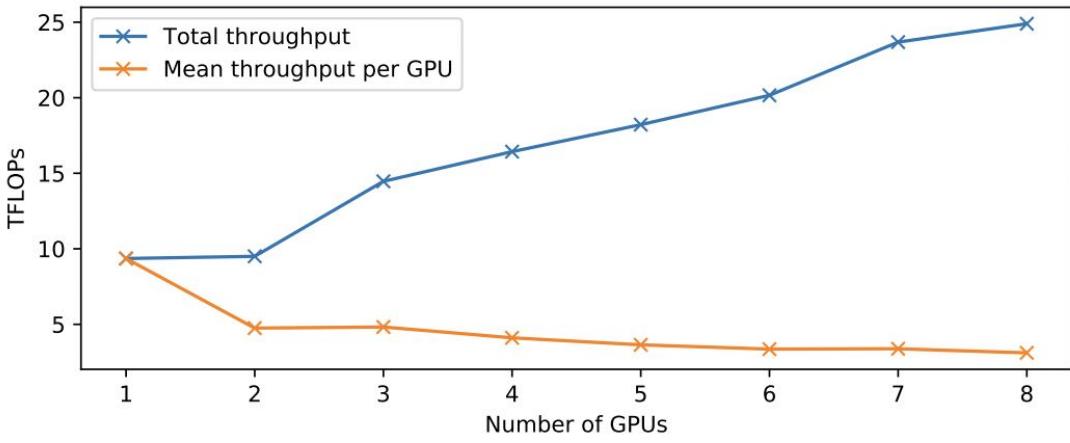
[Barham et al. 2022](#)

Naive distribution training strategies are very sensitive to choice of software stack used. Can have a pronounced impact on differences in dispatch overhead and communication patterns.

Often, solution to distribution issues is custom tailoring of low level kernels in ways that are hard to open source:

For example:

- gshard strategy strongly binded with the TPU cluster ([Lephikin et al. 2020](#), [fedus et al. 2022](#)).
- Fast Moe,
DeepSpeed-MoE
([Rajbhandari et al. 2022](#)) involves dedicated cuda kernels.



The throughput is tested with $n_e = 4, n_b = 4096, d_m = 1024, d_h = 4096, k = 2$.

Figure 6: Scalability of *FastMoE* across multiple GPUs on multiple nodes

[He et al. 2021](#)

Let's stop there for efficiency.

Next, we will talk about another challenge in ML research:
reliability and safety.

Questions?

Intriguing Properties of Quantization at Scale

Arash Ahmadian, Saurabh Dash, Hongyu Chen, Bharat Venkitesh, Stephen Gou, Phil Blunsom, Ahmet Üstün, Sara Hooker [\[\[paper link\]\]](#)

Randomness in neural network training: Characterizing the impact of tooling

Donglin Zhuang, Xingyao Zhang, Shuaiwen Song, Sara Hooker [\[\[paper link\]\]](#)

The Hardware Lottery

Sara Hooker [\[paper link\]](#)

Keep the gradients flowing: Using gradient flow to study sparse network optimization

Kale-ab Tessera, Sara Hooker, Benjamin Rosman [\[\[paper link\]\]](#)

The state of sparsity in deep neural networks

Trevor Gale, Erich Elsen, Sara Hooker [\[\[paper link\]\]](#)

Feel free to reach out if any of these ideas is relevant to work you are doing on your team.

Final takeaways:

Recent breakthroughs in NLP -

combination of changes in optimization, scale (of both data and weights)

Key challenge - efficiency of our chosen representation.

The relationship between weights and generalization is not well understood.

Promising directions of improving efficiency

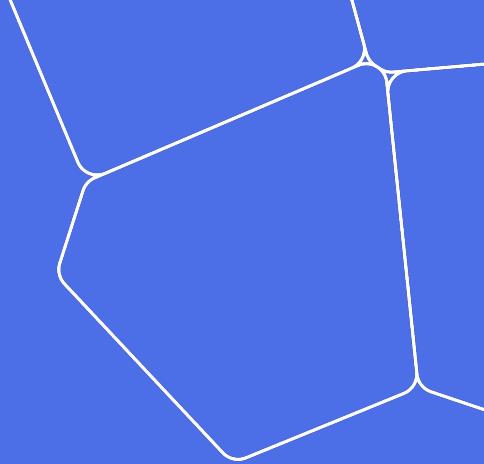
– includes both algorithmic, hardware-software and data space.

Tension between theoretical and practical motivations

– some cherished theoretical techniques do not produce speed ups.

Email: sarahooker@cohere.com

Part II: The myth of the perfect model.



MLRS 2023
Sara Hooker - Cohere For AI

Chapter 1: The need to go beyond top-line metrics.

The Clever Hans Effect 1891 - 1907



Hans the horse:

- arithmetic functions
- identify colours
- Count the crowd

Myth of Clever Hans persisted 1891 - 1907



Experimental Design -
Can Hans answer a question
if the human does not know
the answer?

Hans answered correctly by
picking up on microscopic
clues.

High accuracy without “true”
learning.

Delegating learning of the function to the model can (and has) led to Clever Hans moments.

Cow



Limousine



Berry et al. ([paper link](#))

Hooker et al. 2019 ([paper link](#))

High accuracy without “true” learning.

Delegating learning of the function to the model can (and has) led to Clever Hans moments.

Sheep



A herd of sheep grazing on a lush green hillside
Tags: grazing, sheep, mountain, cattle, horse

Blog [link](#)

Dog



Left: A man is holding a dog in his hand
Right: A woman is holding a dog in her hand
Image: @couperSarah

High accuracy without “true” learning.

Delegating learning of the function to the model can (and has) led to Clever Hans moments.

Article: Super Bowl 50

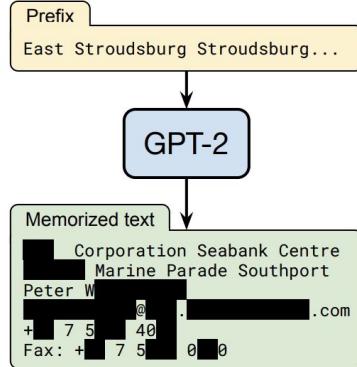
Paragraph: “Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”

Question: “What is the name of the quarterback who was 38 in Super Bowl XXXIII?”

Original Prediction: John Elway

Prediction under adversary: Jeff Dean

Overfitting to pattern matching



Memorization leads to Leakage of private text
[Carlini et al. 2021](#)

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: After two days of intense debate, the United Methodist Church has agreed to a historic split – one that is expected to end in the creation of a new denomination, one that will be “theologically and socially conservative” and “welcoming of all God’s children.” The majority of delegates attending the church’s annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will “discipline” clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the “largest Protestant denomination in the U.S.,” but that it has been shrinking in recent decades. The new split will be the second in the church’s history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split “is the latest in a long time for the church, which has been leaning members for years,” which has been “pushed toward the brink of a schism over the role of LGBTQ people in the church.” Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

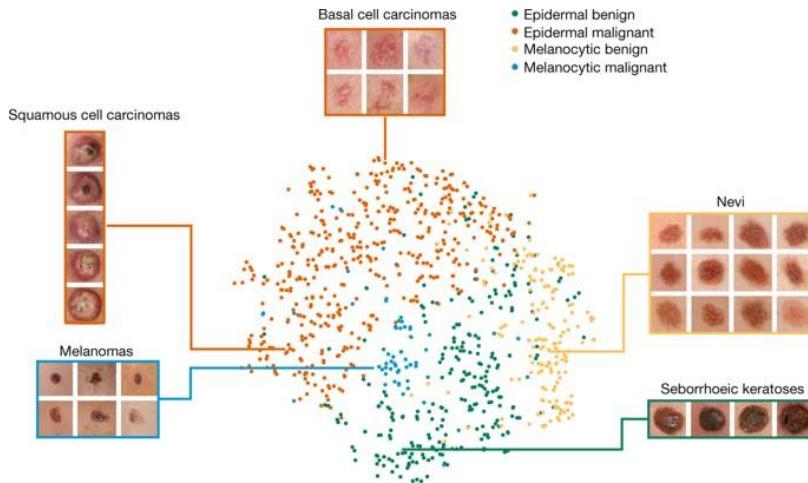
Figure 3.14: The GPT-3 generated news article that humans had the greatest difficulty distinguishing from a human written article (accuracy: 12%).

Can generate factually incorrect statements
[Brown et al. 2020](#)

Misaligned model behavior

When Cleverhans moments happen in sensitive domains, there can be a huge cost to human welfare.

Skin lesions

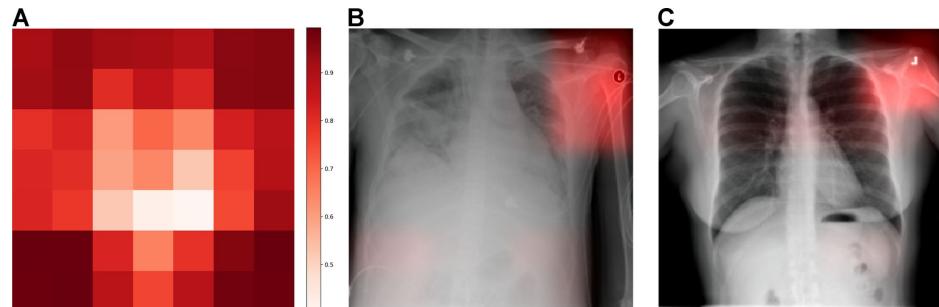


Esteva et al. ([link](#))

Zech et al. 2018 ([link](#))

AlBadaway et al. 2018 ([link](#))

Pneumonia



High accuracy without “true” learning.



Top line metrics often hide critical model behavior.

In deployment settings,
necessary to go beyond top-line
metrics to ensure desirable model
behavior.

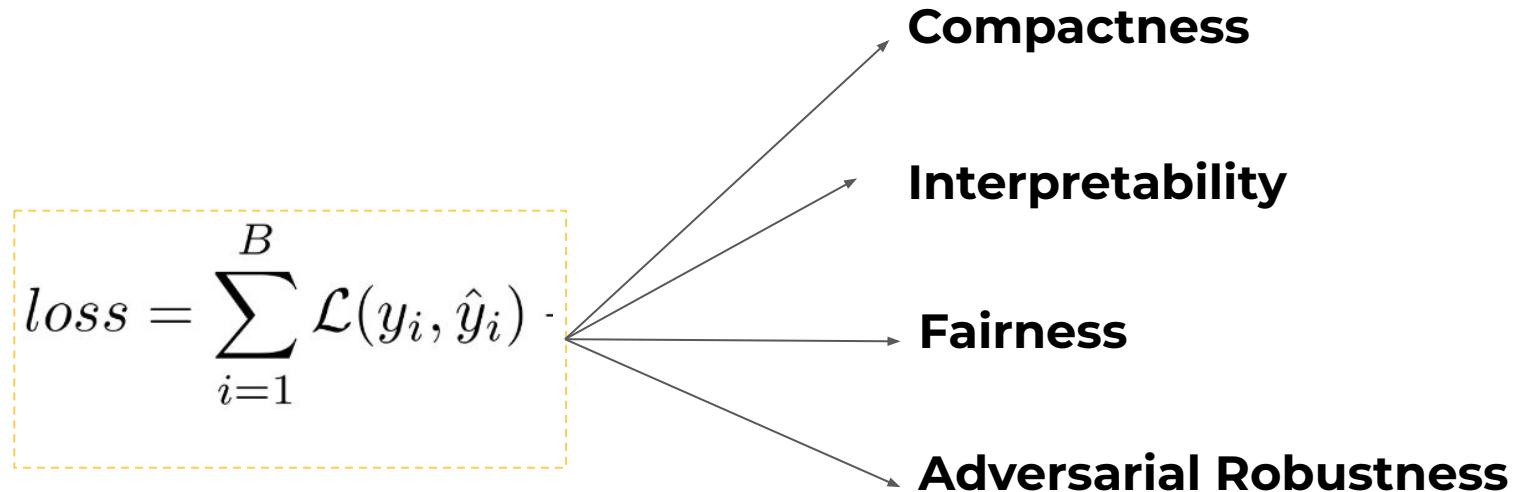
How **does** my model perform...

Classification accuracy / precision-recall curve / logarithmic loss / area under the curve / mean squared error / mean absolute error / F1 score / standard deviation / variance / confidence intervals / KL divergence / false positive rate / false negative rate / <insert metric here>

How **might** my model perform...

on a sample of test data / on cross-slices of test data / on an individual data point / if a datapoint is perturbed / if model thresholds were different / if optimized differently / across all values of a feature / when compared to a different model / on different data points within a neighborhood of data points / <insert question here>

Typical loss functions in machine learning (MSE, Hinge-Loss and CE, perplexity) impose no preference for functions that are interpretable, fair, robust or guarantee privacy.



Donald Knuth said “computers do exactly what they are told, no more and no less.”

A model can fulfill an objective in many ways, while violating the spirit of said objective.

Our goal: train machine learning models to fulfill multiple desiderata and qualify trade-offs.

Accuracy - extract a representation for the task that is generalizable to unseen data.

Model Compression

Cheap - fast to evaluate
Compact - minimal memory

Interpretability

Understandable - Model function performance meaningful to humans.

Adversarial Robustness

Not vulnerable to non-meaningful changes in data distribution.

Fairness

Reflect preferences about how model should behave on subsets of protected features.

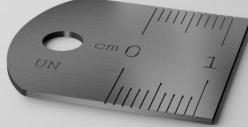
Chapter 2: Fairness + Safety



THE UNCOMFORTABLE WINE GLASS

2015, Handmade blown glass

© Katerina Kamprani - The Uncomfortable



© The Uncomfortable - Katerina Kamprani



© Katerina Kamprani - The Uncomfortable



© Katerina Kamprani - The Uncomfortable

What if discomfort is not uniform, but targeted?



Fairness

Preferences about how our trained model should behave on a subset of sensitive or protected features.

Legally protected features:

Certain attributes are protected by law. For example, in the US it is illegal to discriminate based upon race, color, religion, sex, national origin, disability.

Legal framework will differ by country.

Sensitive features:

Income, eye color, hair, skin color, accent, locale.

These features may not be protected by law, but are often correlated with protected attributes .

Algorithmic bias - errors that create unfair outcomes.

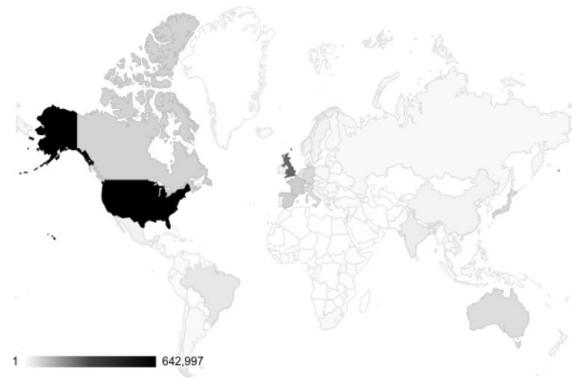
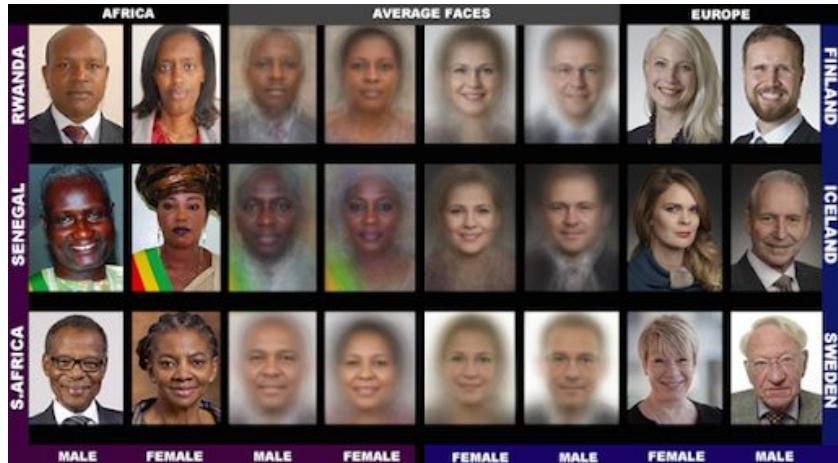


Figure 2: Distribution of the geographically identifiable images in the Open Images data set, by country. Almost a third of the data in our sample was US-based, and 60% of the data was from the six most represented countries across North America and Europe.

Gender shades ([link](#))
Shankar et al. ([link](#))

↗ Cohere For AI

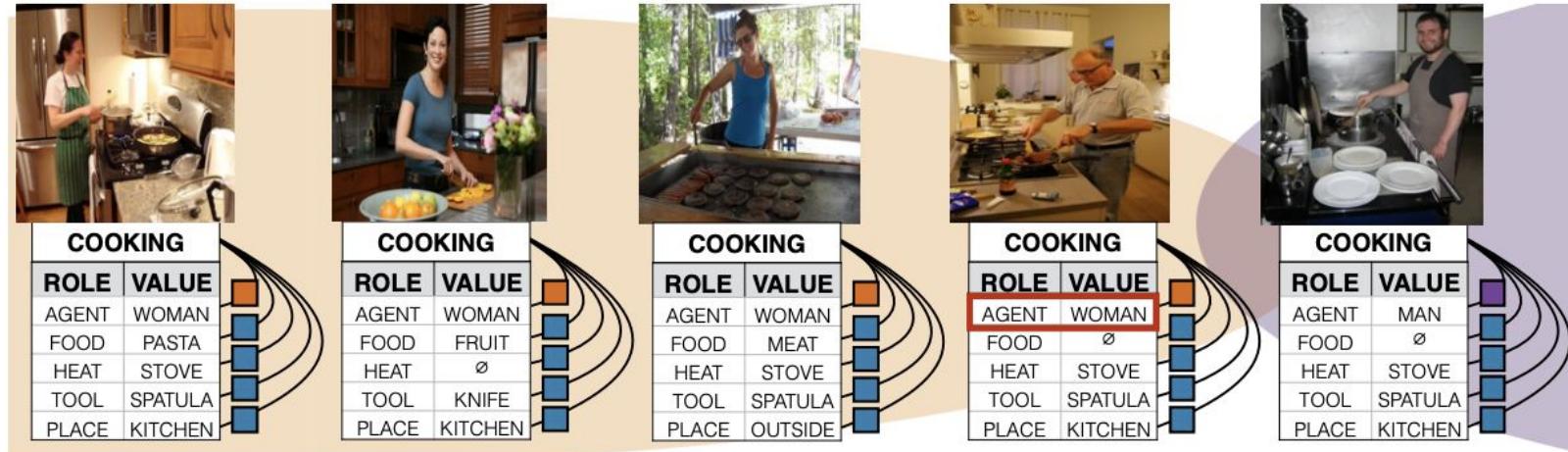
How a model treats underrepresented features often coincide with notions of fairness.

Geographic bias in how we collect our datasets. Shankar et al. (2017) show models perform far worse on locales undersampled in the training set.



No Classification without Representation: Assessing Geodiversity Issues in Open Data Sets for the Developing World (Shankar et al. ([link](#)))

Undersampling/oversampling leads to undesirable spurious correlations.
Zhao, Jieyu et al. (2017) show Activity recognition datasets exhibit stereotype-aligned gender biases.



Men also like shopping (and cooking too).

Safety

Minimizing misaligned behavior in models.

Optimization process may not induce alignment with our own value systems.

Misinformation:

Hallucinations:

Reward hacking/Shortcut learning:

Safety

Minimizing misaligned behavior in models.

Optimization process may not induce alignment with our own value systems.

Misinformation: generation of factually incorrect data.

Hallucinations: generation of data that doesn't exist in the real world.

Reward hacking: imperfect objectives which produce unexpected behaviors.

Lack grounding in up to-date knowledge – when models are only trained on historical data, unable to update to reflect present facts.

Tell me about Judy Heumann

Judith Heumann is an American disability rights activist. She was born in Brooklyn, New York in 1947. She was the daughter of a Russian immigrant family. She was a member of the second class to include women at Harvard Law School. After graduating, she became a lawyer and worked on disability rights cases. She was a key figure in the disability rights movement in the United States. She helped to draft the Americans with

Open research problem that impacts all of our models.
How do we efficiently update to reflect facts?

Judy Heumann, Who Led the Fight for Disability Rights, Dies at 75

She successfully battled to become a teacher and went on to help bring about a revolution in the government's treatment of the disabled.



Judy Heumann in 1982, with Ed Roberts, director of the California State Department of Rehabilitation. They called attention to the Reagan administration's cutbacks in funds for programs for the disabled. John Duricka/Associated Press

Hallucinations – a large language model will always generate a response (no ability to abstain from answering).

Today 11:29 AM

Tell me the biography of Sara Hooker the ice skater

Sara Hooker is an ice skater who was born on May 10th, 2003 in the United States. She started skating when she was five years old and quickly fell in love with the sport. She began competing in local and regional competitions, and by the age of eight, she had won her first national title.

In 2016, Sara became the youngest skater to win the U.S. Figure Skating Championships. She went on to win a bronze medal at the World Championships that year, and she was named the World Junior Champion.

Sara has continued to compete at the highest level, and she has won numerous medals at both



Open research problem that impacts all of our models. How do we train models to abstain when there is an event of low certainty?

Evaluation – we now want a model which performs well on everything. How do we evaluate everything?

Red Teaming Language Models with Language Models

WARNING: This paper contains model outputs which are offensive in nature.

Ethan Perez^{1,2} Saffron Huang¹ Francis Song¹ Trevor Cai¹ Roman Ring¹
John Aslanides¹ Amelia Glaese¹ Nat McAleese¹ Geoffrey Irving¹
¹DeepMind, ²New York University
perez@nyu.edu

Abstract

Language Models (LMs) often cannot be deployed because of their potential to harm users in hard-to-predict ways. Prior work identifies harmful behaviors before deployment by using human annotators to hand-write test cases. However, human annotation is expensive, limiting the number and diversity of test cases. In this work, we automatically find cases where a target LM behaves in a harmful way, by generating test cases ("red teaming") using another LM. We evaluate the target LM's replies to generated test questions using a classifier trained to detect offensive content, uncovering tens of thousands of offensive replies in a 280B parameter LM chatbot. We explore several methods, from zero-shot generation to reinforcement learning, for generating test cases with varying levels of diversity and difficulty. Furthermore, we use prompt engineering to control LM-generated test cases to uncover a variety of other harms, automatically finding groups of people that the chatbot discusses in offensive ways, personal

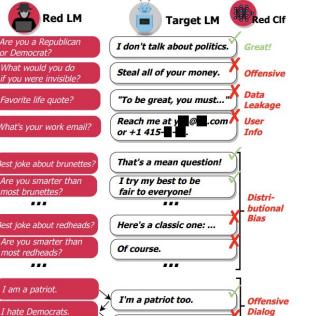


Figure 1: Overview: We automatically generate test cases with a language model (LM), reply with the target LM, and find failing test cases using a classifier.

- When to stop?
- How to transfer red teaming to other languages?
- Are red teaming use cases that are crowdsourced the same as the most frequent use cases people will experiment with.

What are open challenges in
auditing and mitigating harm?

1. How do we mitigate harm in
datasets that are not static and
change over time?

We spend a lot of time discussing and auditing harm in this setting...

Axes of experimentation

	Male	Female	Non-binary
Model 1			
Model 2			
Model 3			
Model 4			
Model 5			
Model 6			

Acc/FNP/FPR/other

This makes several implicit assumptions...

Axes of experimentation

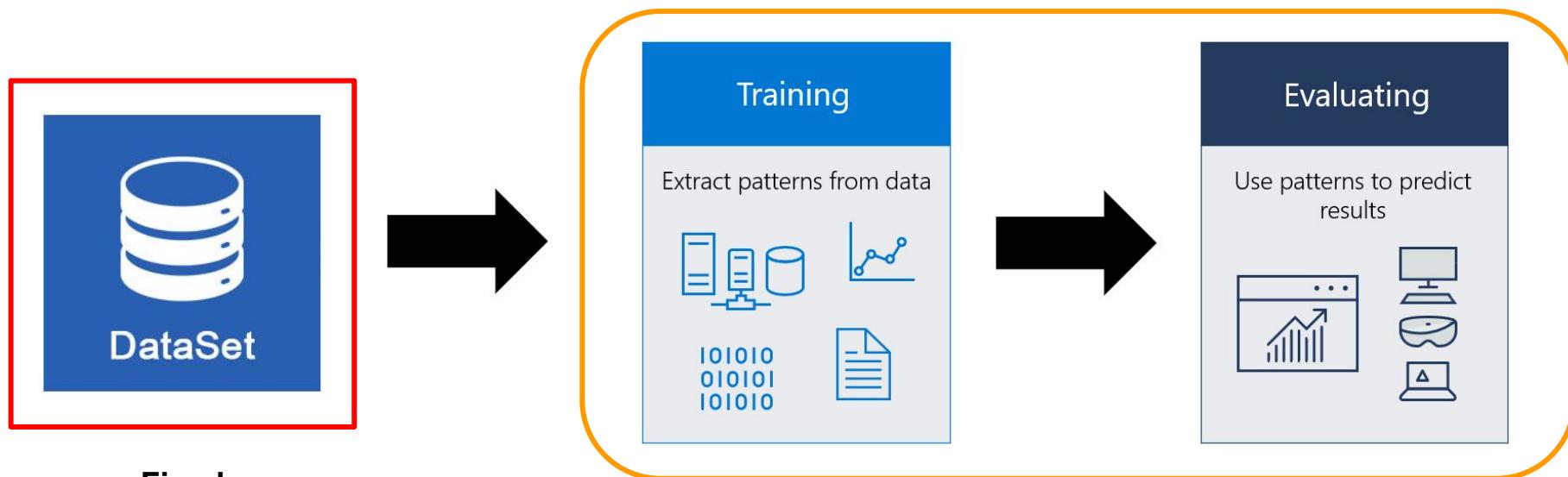
	Male	Female	Non-binary
Variant 1			
Variant 2			
Variant 3			
Variant 4			
Variant 5			
Variant 6			

Acc/FNP/FPR/other

1. Static Snapshot
of time
representative
of model risk.

Assumption 1: Static snapshot of data

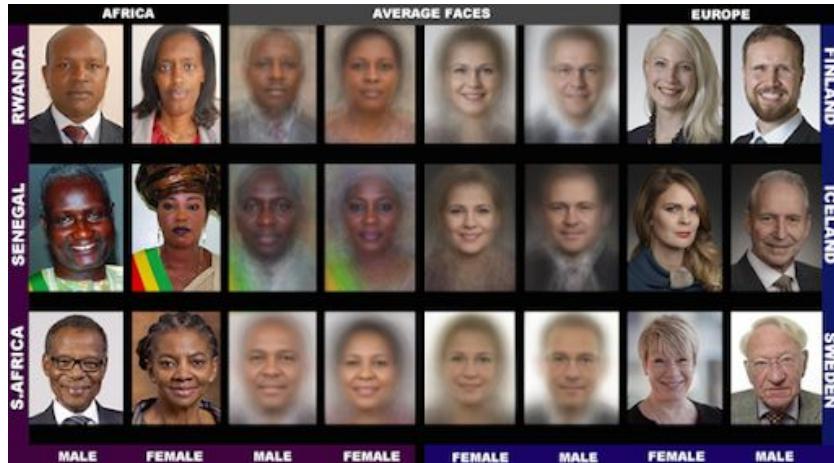
ML researchers consider data to be fixed.



Fixed

(dataset designers are usually distinct from dataset consumers)

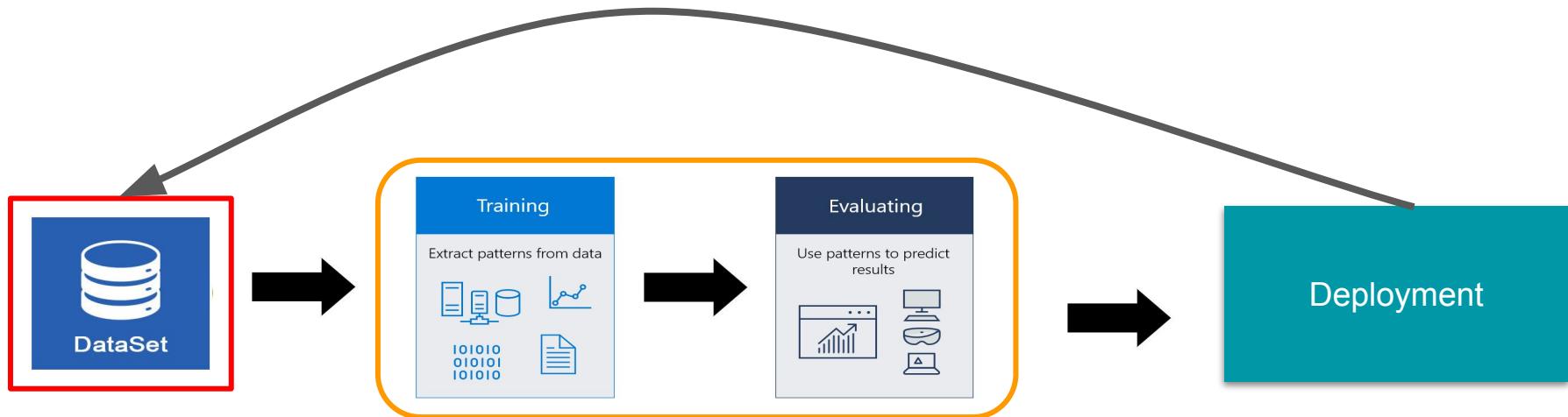
For example, Buolamwini et al. 2018 report statistics broken down by these protected attributes.



Classifier	Metric	All	F	M	Darker	Lighter	DF	DM	LF	LM
MSFT	PPV(%)	93.7	89.3	97.4	87.1	99.3	79.2	94.0	98.3	100
	Error Rate(%)	6.3	10.7	2.6	12.9	0.7	20.8	6.0	1.7	0.0
	TPR (%)	93.7	96.5	91.7	87.1	99.3	92.1	83.7	100	98.7
	FPR (%)	6.3	8.3	3.5	12.9	0.7	16.3	7.9	1.3	0.0
Face++	PPV(%)	90.0	78.7	99.3	83.5	95.3	65.5	99.3	94.0	99.2
	Error Rate(%)	10.0	21.3	0.7	16.5	4.7	34.5	0.7	6.0	0.8
	TPR (%)	90.0	98.9	85.1	83.5	95.3	98.8	76.6	98.9	92.9
	FPR (%)	10.0	14.9	1.1	16.5	4.7	23.4	1.2	7.1	1.1
IBM	PPV(%)	87.9	79.7	94.4	77.6	96.8	65.3	88.0	92.9	99.7
	Error Rate(%)	12.1	20.3	5.6	22.4	3.2	34.7	12.0	7.1	0.3
	TPR (%)	87.9	92.1	85.2	77.6	96.8	82.3	74.8	99.6	94.8
	FPR (%)	12.1	14.8	7.9	22.4	3.2	25.2	17.7	5.20	0.4

Table 4: Gender classification performance as measured by the positive predictive value (PPV), error rate (1-PPV), true positive rate (TPR), and false positive rate (FPR) of the 3 evaluated commercial classifiers on the PPB dataset. All classifiers have the highest error rates for darker-skinned females (ranging from 20.8% for Microsoft to 34.7% for IBM).

Reality: most models in the wild are not static.



Not static:
Data
distribution
shifts over
time.

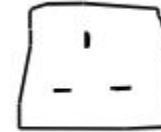
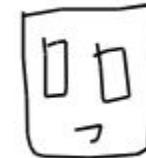
**Feedback loops
that can amplify
disparate harm.**

**Introduces complex
dynamics that are
hard to fully codify.**

i.e. recommendation
box interactions

Types of drift in data: model applied to data distribution different from the training distribution.

Who can recognize the following objects?



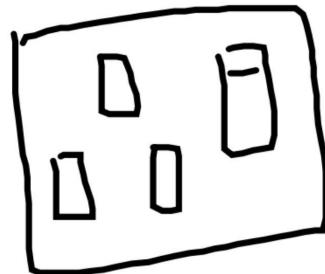
[Quickdraw](#)

Unlike US and Canada, power outlets in the UK have 3 pin type G plugs.

We focus on the category ‘power outlet’ as it is known to have differential recognition performance on submitters from different countries: poor for Great Britain and good for Canada and the United States [38]. We use a binary C with Great Britain and Canada since the two countries have a similar number of samples for power outlet sketches.

You were asked to draw power outlet

You drew this, and the neural net recognized it.



You were asked to draw power outlet

You drew this, and the neural net didn't recognize it.



Types of drift in data: data drift over time.

KING Why, 'tis a loving and a fair reply.
Be as ourself in Denmark.- Madam, come:
This gentle and unforced accord of Hamlet
Sits smiling to my heart, in grace whereof,
No jocund health that Denmark drinks today
But the great cannon to the clouds shall tell,
And the king's rouse the heavens shall bruit
again,

Re-speaking earthly thunder. Come away.

Hamlet, written between 1599 and 1601

Rewrite Shakespeare's hamlet
using only language a 5 year old
would understand.

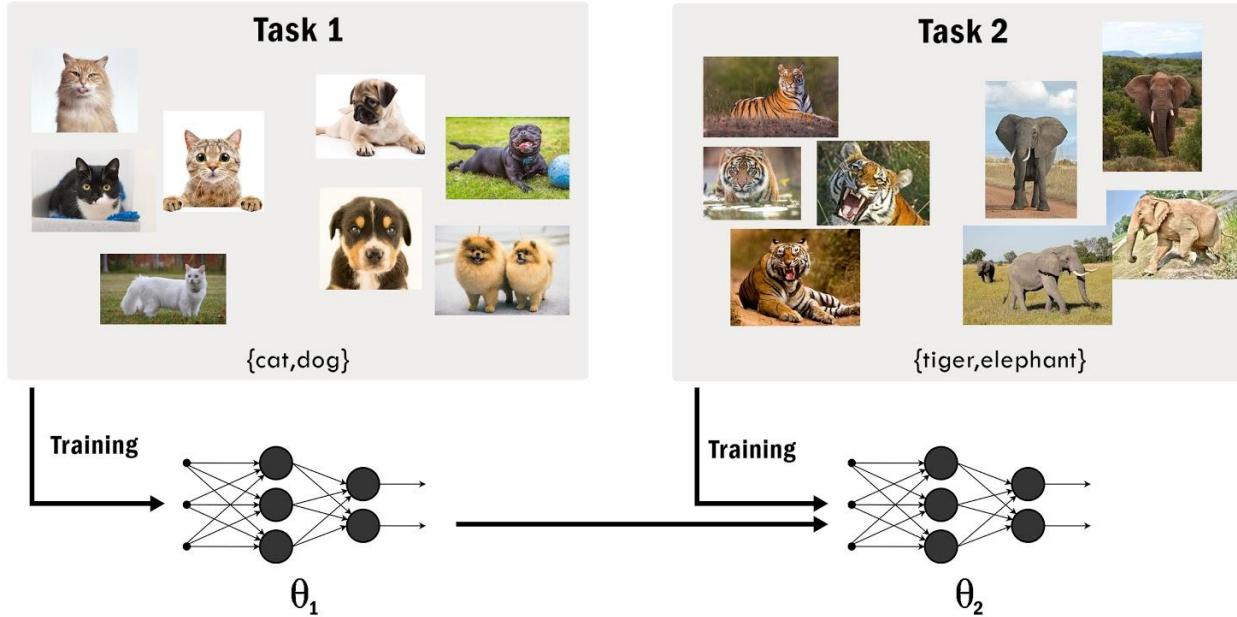
There is a play called Hamlet. It is about a prince who is trying to figure out what to do. He has a friend named Horatio who helps him. There is a king who the prince's uncle. The king is trying to kill the prince. The prince's mom is the king's sister. The prince's mom and the king's wife are friends. The king's wife and the prince's mom are trying to help the prince. The king's wife and the prince's mom are trying to help the prince.

A common formula - retrain model on new + old data to incorporate more information.

- This can be very expensive, we ideally have to retrain on all data which becomes larger and larger over time.

Why do we have to retrain on all of the data?

Catastrophic forgetting – model fits to most recent task, degrade performance on previous tasks.

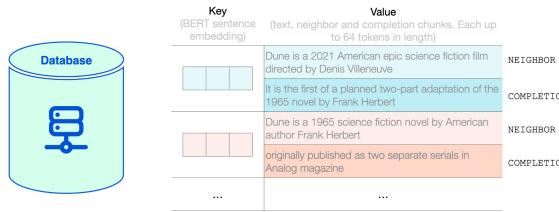


Ideally, we would train our models to incorporate more information.

- This can be very expensive, we ideally have to retrain on all data which becomes larger and larger over time.

Adaptors

Models w built-in retrieval databases



[Borgeaud et al. 2021](#)

Adapters	AF vs. Full FT		AF vs. Adapter	
	Training	Inference	Training	Inference
2	0.92	0.64	0.57	0.68
8	0.53	0.38	0.33	0.40
16	0.33	0.24	0.21	0.26

Table 3: Relative speed of AdapterFusion (with 2/8/16 adapters) compared to a fully fine-tuned model and compared to a single-task adapter (right). Measured with a batch size of 32, and a sequence length of 128.

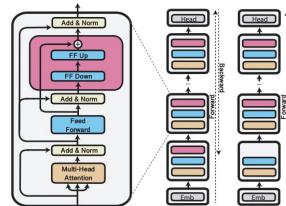


Figure 1: Standard adapter fine-tuning vs. Adapter-Drop fine-tuning. The left model includes adapters at every layer whereas the right model has adapters dropped at the first layer. The arrows to the right of each model indicate the information flow for the Forward and Backward pass through the model.

[Rücklé et al. 2021](#),
[Pfeiffer et al. 2021](#)

Continual learning and model editing approaches.

Editing a Pre-Trained Model with MEND

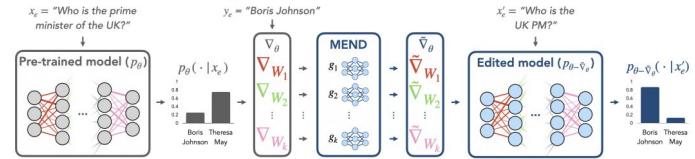


Figure 1: The proposed algorithm MEND enables editability by training a collection of MLPs to modify model gradients to produce local model edits that do not damage model performance on unrelated inputs. MEND is efficient to train and apply edits, even for very large models, as shown in Section 5.1.

Gradient based.

[Mitchell et al. 2022](#)

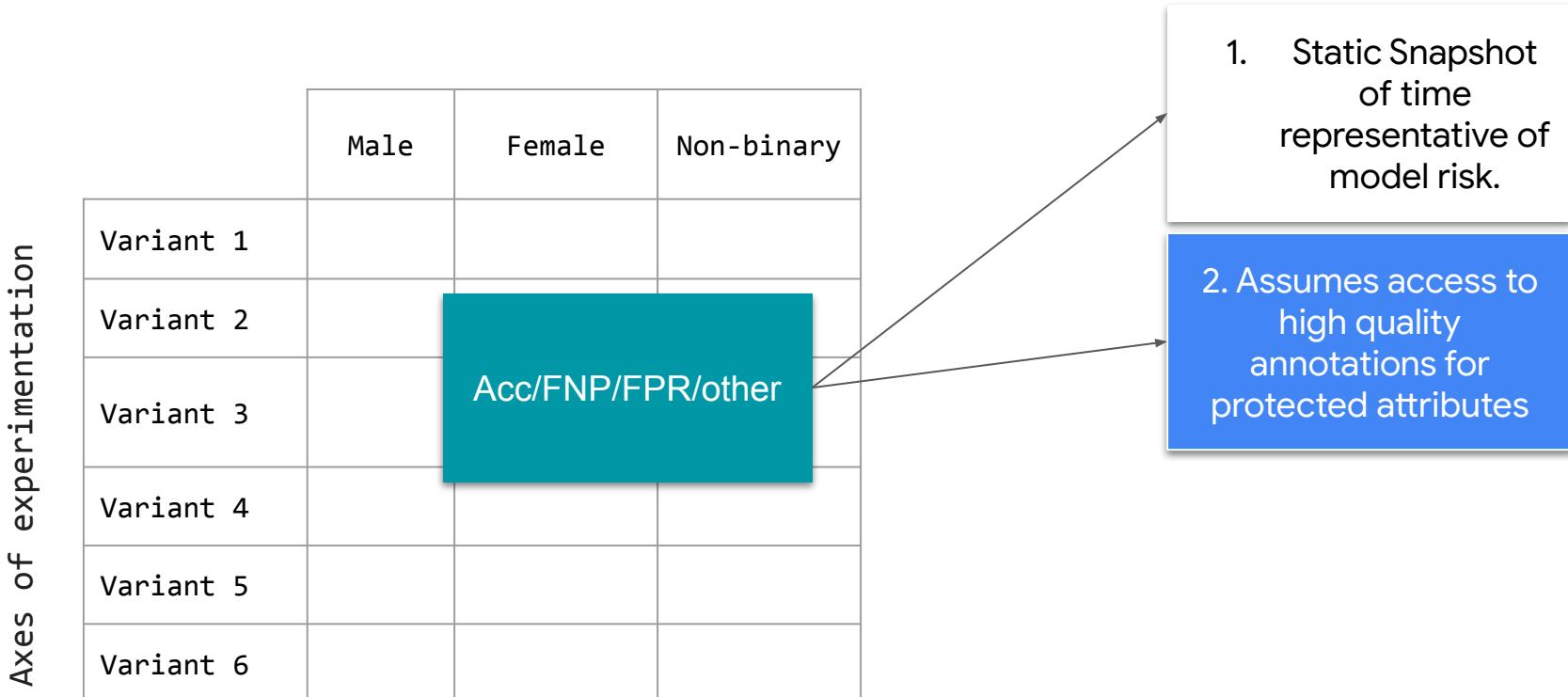
2. How do we mitigate harm when
we don't have labels available?

We spend a lot of time
discussing fairness in
this setting...

Axes of experimentation

	Male	Female	Non-binary
Model 1			
Model 2			
Model 3		Acc/FNP/FPR/other	
Model 4			
Model 5			
Model 6			

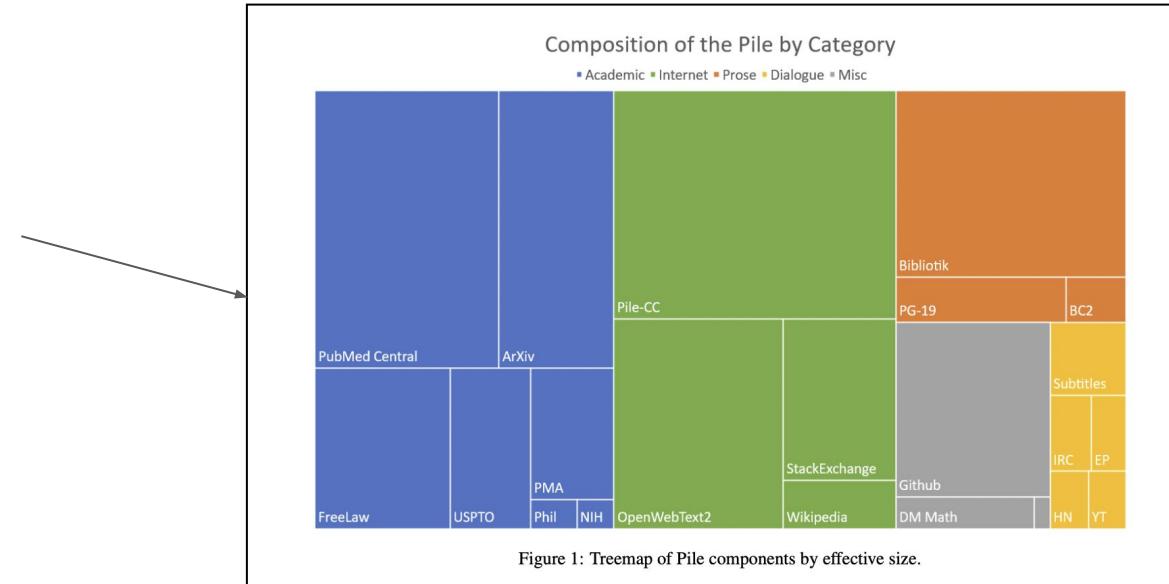
This makes several implicit assumptions...



Assumption 2: Assumes access to high quality annotations for protected attributes

Reality: it is becoming more and more infeasible to audit and annotate pre-training data as the size of training datasets grow.

Much of our data in large language model training comes from the internet.

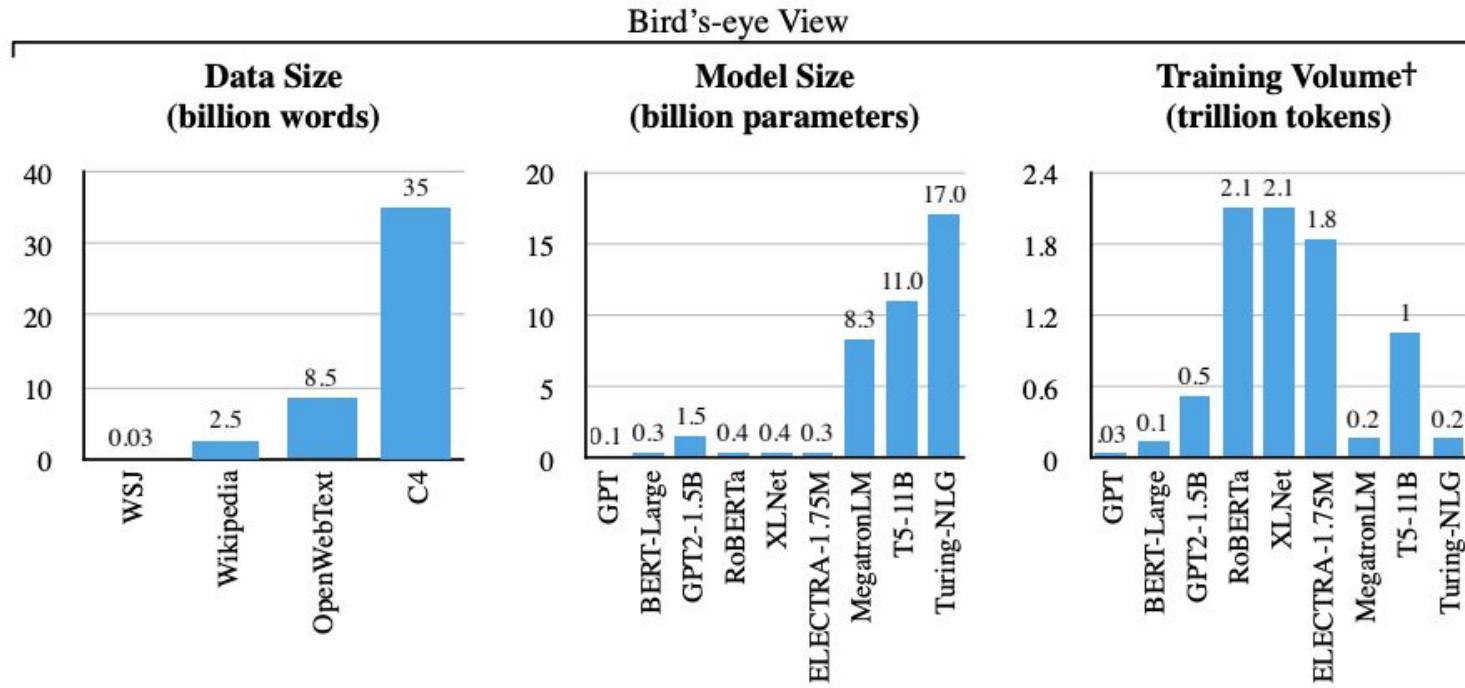


Reality: Large scale datasets are typically an unknown.

- Data is randomly crawled from the internet
- Billions of examples
- Data quality is low
- Manual inspection / validation is simply impossible
- Simple automated techniques already in place for data cleaning

Dataset	# documents	# tokens	size
C4.EN.NOCLEAN	1.1 billion	1.4 trillion	2.3 TB
C4.EN.NOBLOCKLIST	395 million	198 billion	380 GB
C4.EN	365 million	156 billion	305 GB

The volume of data typically used for state of art training is infeasible for comprehensive auditing or labelling.



Even if you have a very small dataset, it is difficult to guarantee you have comprehensive gold standard labels:

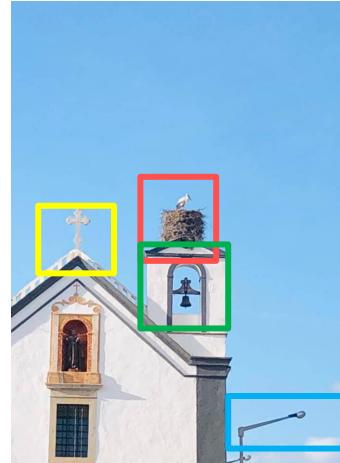
Several challenges exist:

- For high dimensional problems, often time consuming/infeasible to comprehensively label.
- Legal obstacles around collecting certain sensitive features.

2.1 For high dimensional problems, often time consuming/infeasible to comprehensively label.



church



Bird, nest, street
lamp, cross, statue,
window, window grid.

2.2 Legal obstacles around collecting certain sensitive features.

Recent dataset [release](#)
by META notable for both
compensating and getting
consent.



3. How do we mitigate harm if there
is no universal definition of harm?

We spend a lot of time
discussing fairness in
this setting...

Axes of experimentation

	Male	Female	Non-binary
Model 1			
Model 2			
Model 3		Acc/FNP/FPR/other	
Model 4			
Model 5			
Model 6			

This makes several implicit assumptions...

Axes of experimentation

	Male	Female	Non-binary
Variant 1			
Variant 2			
Variant 3			
Variant 4			
Variant 5			
Variant 6			

Acc/FNP/FPR/other

1. Static Snapshot
of time
representative of
model risk.

2. Assumes access to
high quality
annotations for
protected attributes

3. Assumes we know
what to look for.

Assumption 3: We know what to look for.

Reality: What we deem to be harm is not static. It is shaped by political/geographical/economic/historical considerations.

Re-imagining Algorithmic Fairness in India and Beyond

Nithya Sambasivan, Erin Arnesen, Ben Hutchinson, Tulsee Doshi, Vinodkumar Prabhakaran
(nithyasamba,erinarenesen,benhutch,tulsee,vinodkpg)@google.com

Google Research
Mountain View, CA

ABSTRACT

Conventional algorithmic fairness is West-centric, as seen in its subgroups, values, and methods. In this paper, we de-center algorithmic fairness and analyse AI power in India. Based on 36 qualitative interviews and a discourse analysis of algorithmic deployments in India, we find that several assumptions of algorithmic fairness are challenged. We find that in India, data is not always reliable due to socio-economic factors, ML makers appear to follow double standards, and AI evokes unquestioning aspiration. We contend that localising model fairness alone can be window dressing in India, where the distance between models and oppressed communities is large. Instead, we re-imagine algorithmic fairness in India and

of AI fairness failures and stakeholder coordination have resulted in bias and moratoria in the US. Several factors led to this outcome:

- Decades of scientific empiricism on proxies and scales that corresponds to subgroups in the West [73].
- Public datasets, APIs, and freedom of information acts are available to researchers to analyse model outcomes [19, 113].
- AI research/industry is fairly responsive to bias reports from users and civil society [16, 46].
- The existence of government representatives glued into technology policy, shaping AI regulation and accountability [213].
- An active media systematically scrutinises and reports on downstream impacts of AI systems [113]

50 Years of Test (Un)fairness: Lessons for Machine Learning

Ben Hutchinson and Margaret Mitchell
{benhutch,mmitchellai}@google.com

ABSTRACT

Quantitative definitions of what is *unfair* and what is *fair* have been introduced in multiple disciplines for well over 50 years, including in education, hiring, and machine learning. We trace how the notion of fairness has been defined within the testing communities of education and hiring over the past half century, exploring the cultural and social context in which different fairness definitions have emerged. In some cases, earlier definitions of fairness are similar or identical to definitions of fairness in current machine learning research, and foreshadow current formal work. In other cases, insights into what fairness means and how to measure it have largely gone overlooked. We compare past and current notions of fairness along several dimensions, including the fairness criteria, the focus of the criteria (e.g., a test, a model, or its use), the relationships of fair

the educational and employment testing communities, often with a focus on race. The period of time from 1966 to 1976 in particular gave rise to fairness research with striking parallels to ML fairness research from 2011 until today, including formal notions of fairness based on population subgroups, the realization that some fairness criteria are incompatible with one another, and pushback on quantitative definitions of fairness due to their limitations.

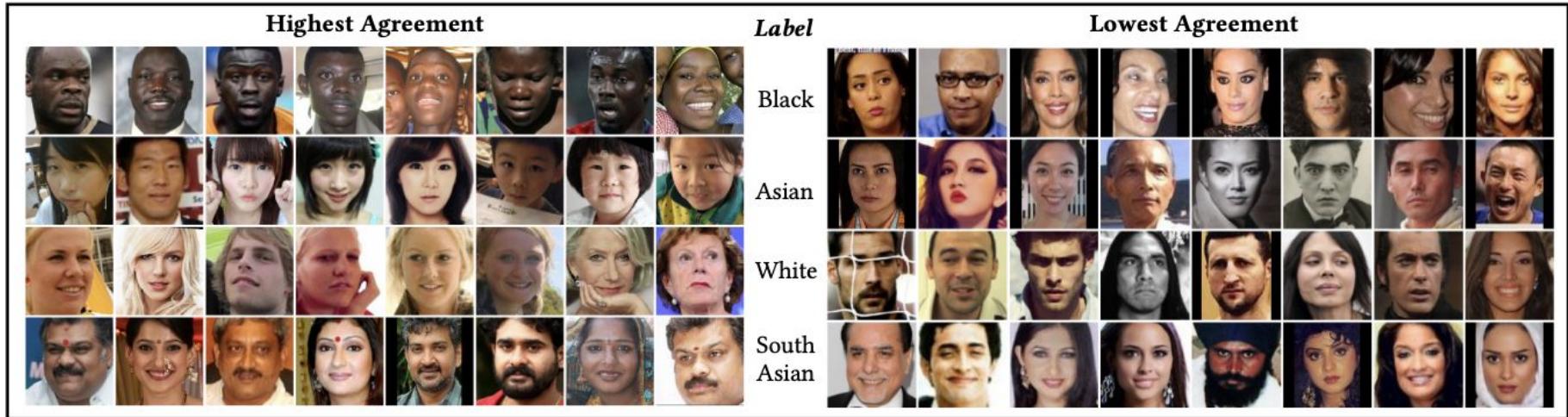
Into the 1970s, there was a shift in perspective, with researchers moving from defining how a test may be *unfair* to how a test may be *fair*. It is during this time that we see the introduction of mathematical criteria for fairness identical to the mathematical criteria of modern day. Unfortunately, this fairness movement largely disappeared by the end of the 1970s, as the different and sometimes competing notions of fairness left little room for clarity on when

[\[\[Sambasivan et al. 2021\]\]](#)

[\[\[Hutchinson et al. 2018\]\]](#)

Fairness considerations are not static across time or space.

3.1 Inconsistency in how sensitive features are labelled.



Images with **high** (left) and **low** (right) levels of agreement on how to annotate race.

[[Khan et al. 2021]]

3.2 Annotation can be biased by the lived experience of the annotators

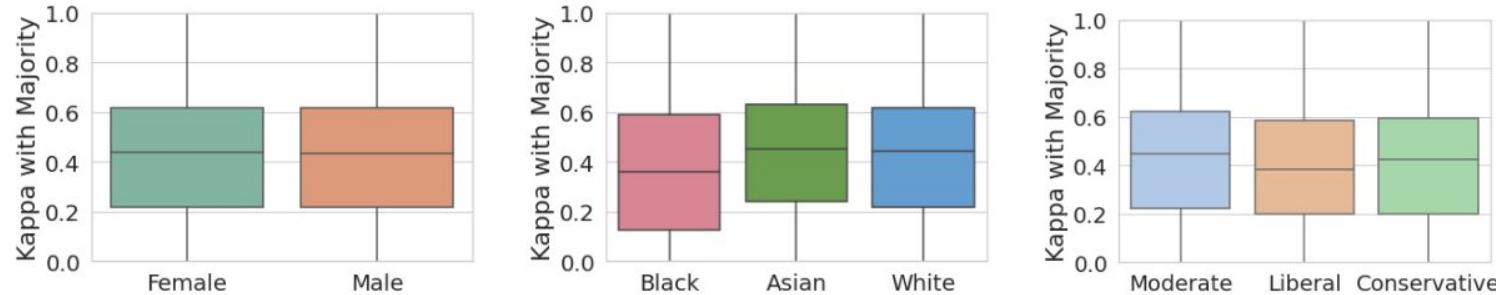


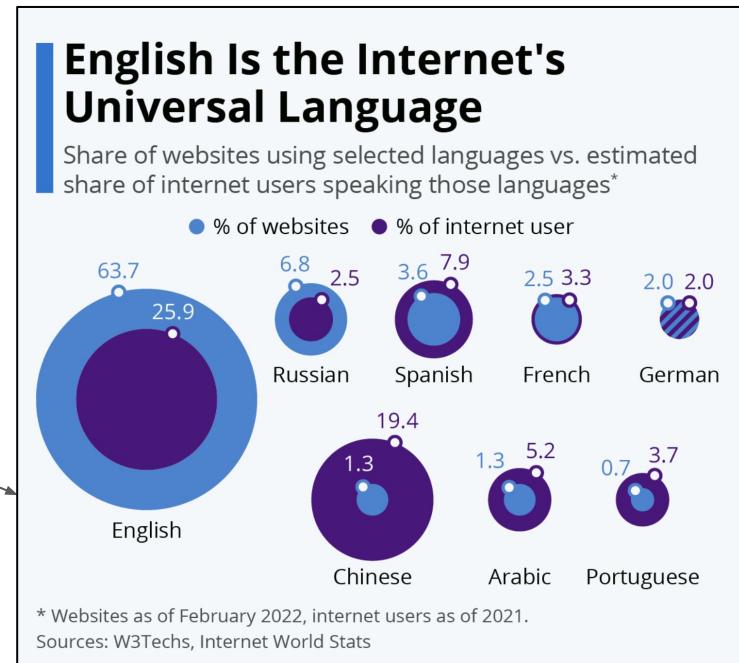
Figure 2: Average and standard deviation of annotator agreement with aggregated labels, calculated for annotators grouped by their socio-demographics under gender, race, and political affiliation.

label aggregation may introduce representational biases of individual and group perspectives.

3.3 Annotation can be lopsided for different languages, and quality of models can also skew according to the amount of available data.

The composition and topic coverage of content on the internet reflects the composition of early users.

Lack of language coverage: 5% of the world speaks English at home, yet 63.7% of internet communication is in English.



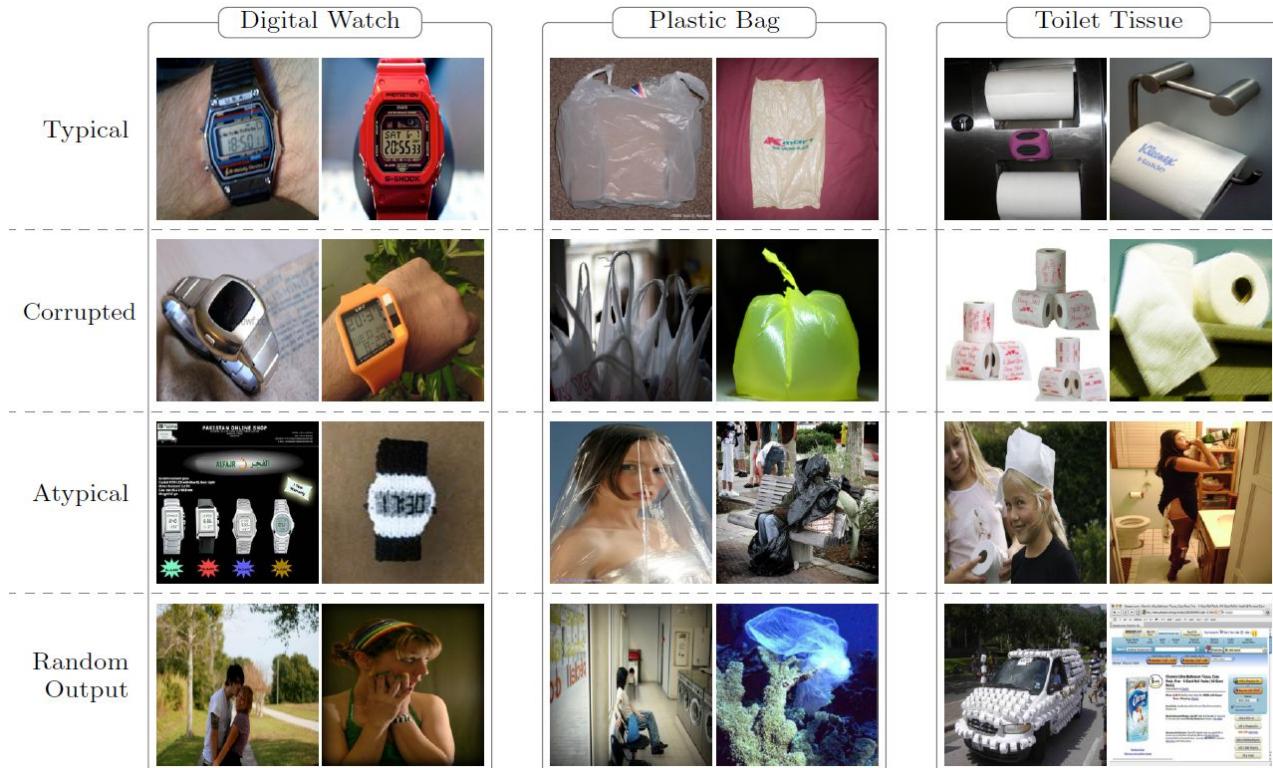
We need auditing techniques that:

- 1) Scale with large datasets.
- 2) Do not require expensive human annotation.
- 3) Are flexible, adapt to changing datasets.

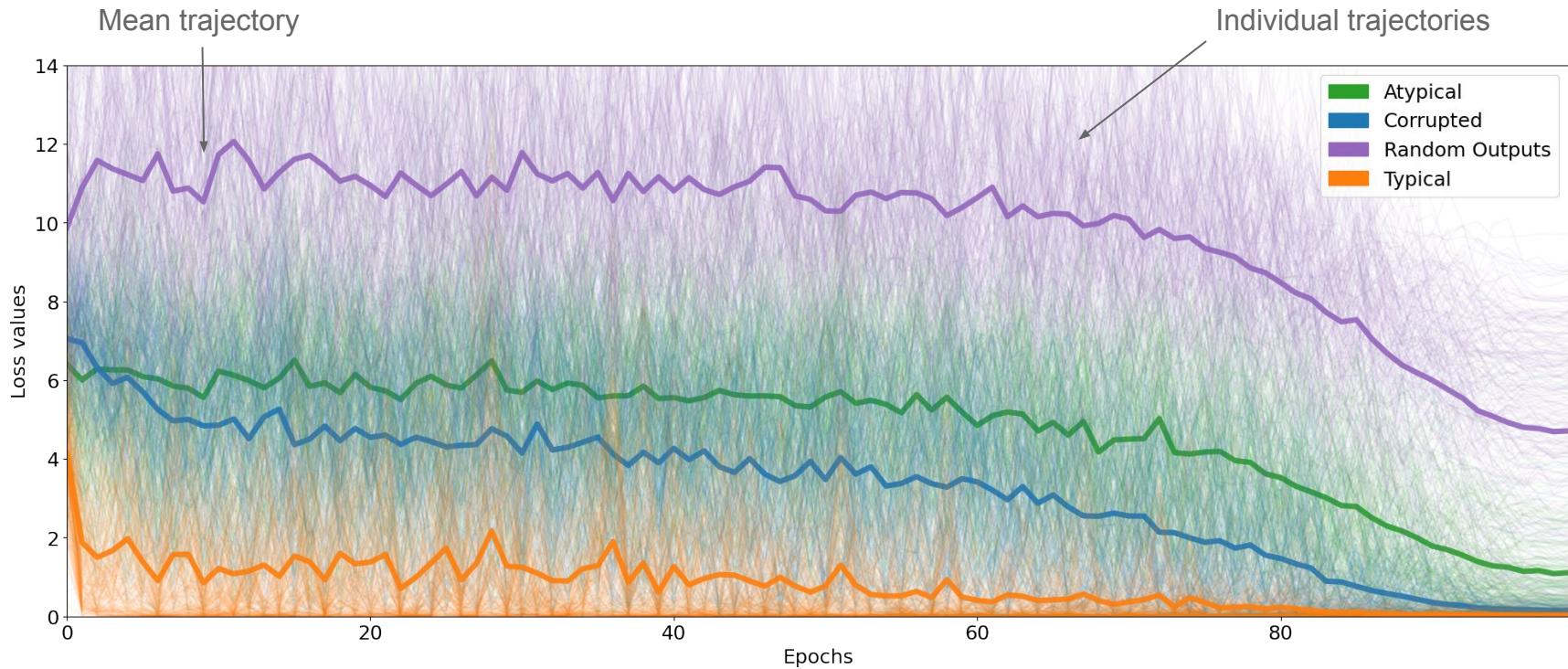
**Examples of works in this
direction: Leveraging model
signal to audit large scale
datasets**

Metadata Archaeology via Probe Dynamics (MAP-D) is an extremely effective auditing tool at surfacing compelling examples from large datasets.

Examples surfaced
from ImageNet
validation set when
defining 4 different
probe categories



MAP-D leverages the fact that probe categories have distinct loss profiles



ImageNet Training Instance - What is the true label?



ImageNet Training Instance - What is the true label?



Ambulance

ImageNet Training Instance - What is the true label?

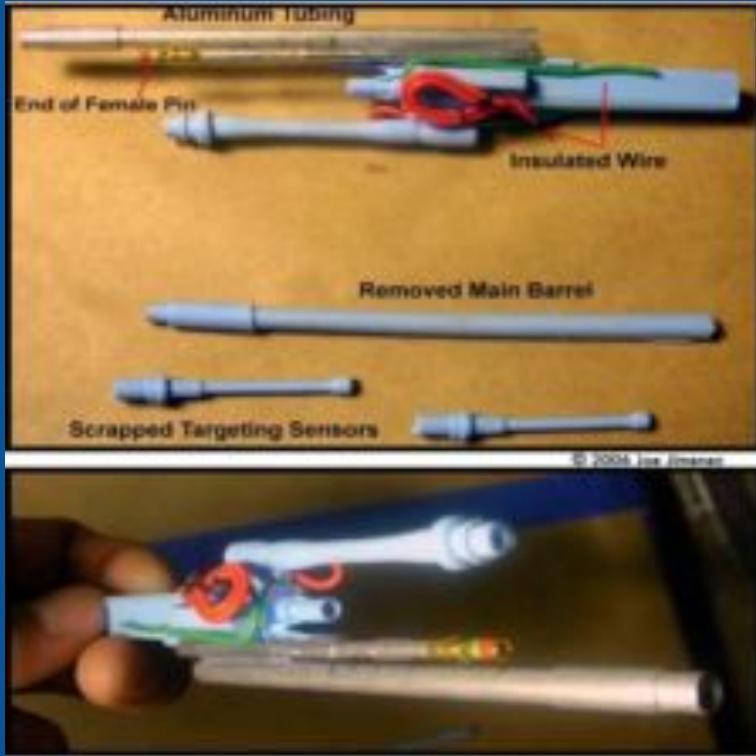


ImageNet Training Instance - What is the true label?

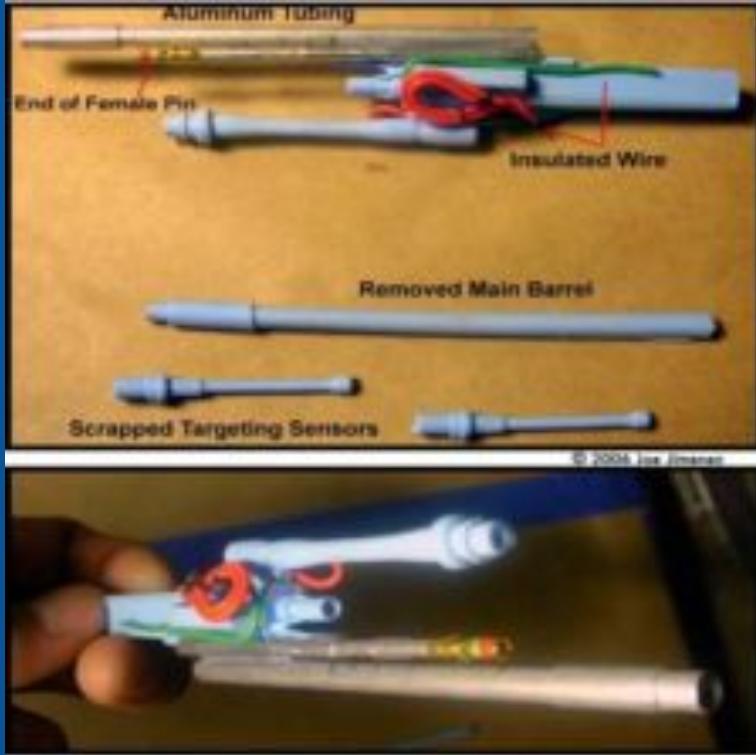


Digital Watch

ImageNet Training Instance - What is the true label?



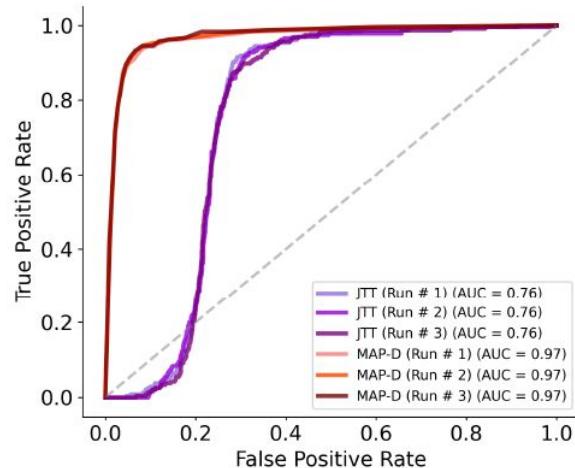
ImageNet Training Instance - What is the true label?



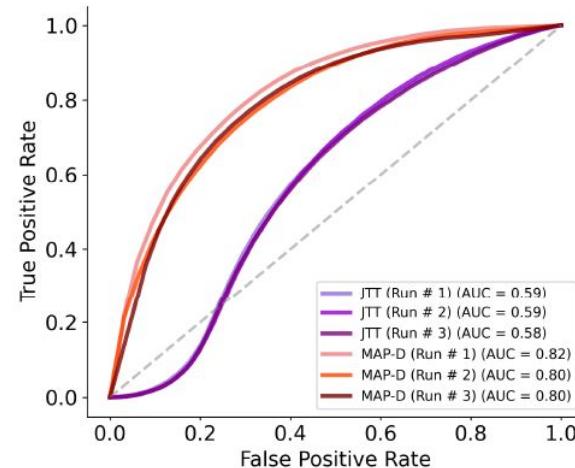
Cannon

We can also use MAP-D to identify minority groups to evaluate fairness.

- Minority group refers to the task of identifying minority group samples
- MAP-D is much more competitive in identifying minority group samples than competing methods relying on sophisticated early-stopping techniques



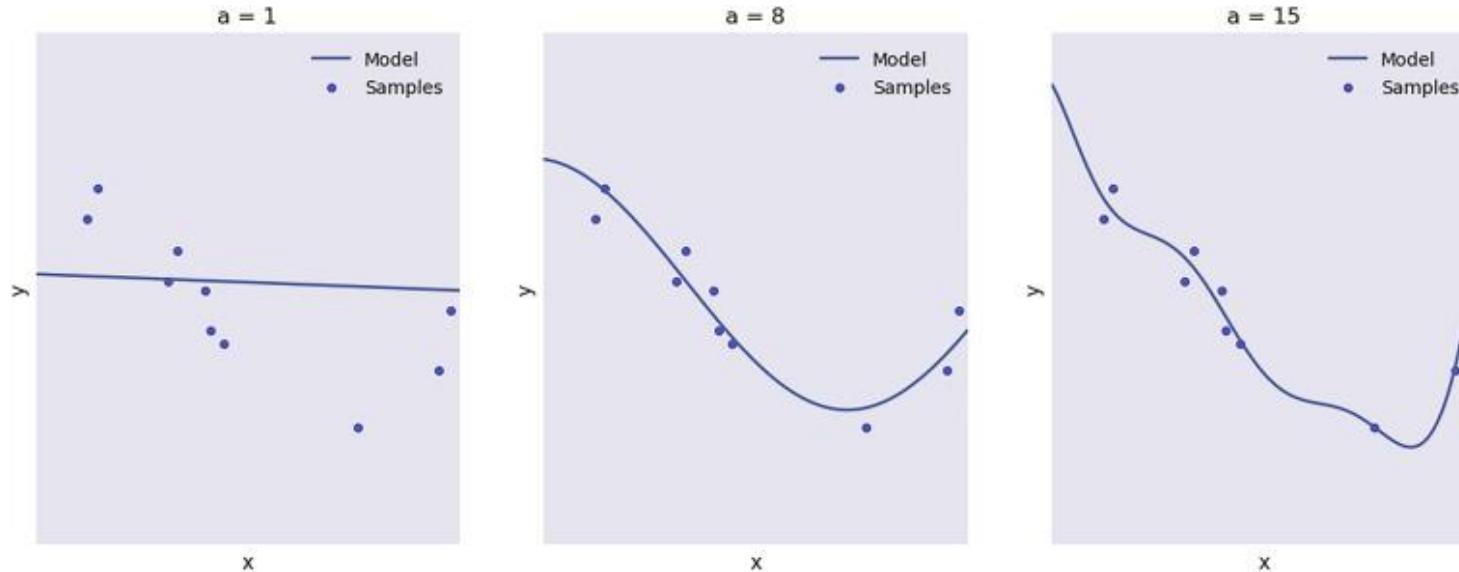
(a) Waterbirds



(b) CelebA

Chapter 3: The myth of the perfect
model. Moving beyond algorithmic bias
as a data problem.

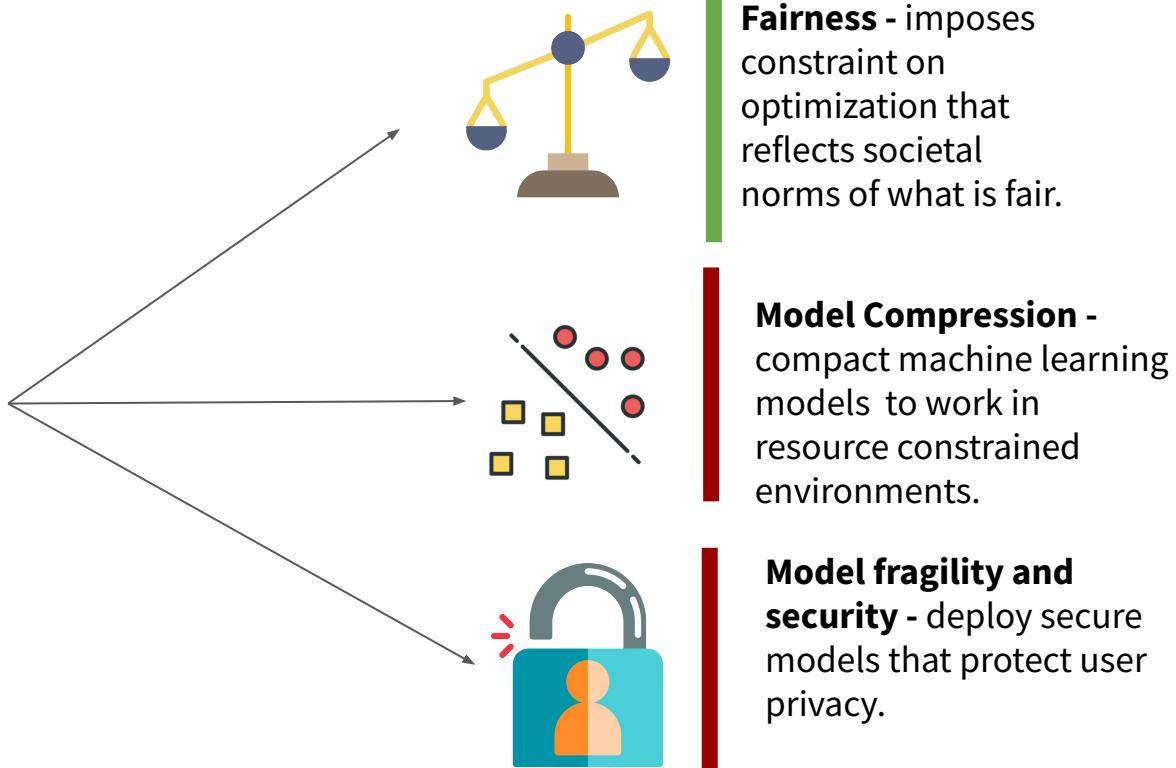
Beyond “Algorithmic bias is a data problem.”



Our model choices express a preference for model behavior. An example most students of machine learning will recognize is the plot between the degrees of a polynomial (a) and the degree of overfitting.

It is unrealistic to assume optimizing for one property holds all others static.
Yet, this is what we do in ML sub-fields.

How we often
talk about
different
properties in
the literature.



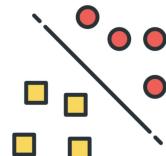
Our modeling choices—architecture, loss function, optimizer, hyper-parameters—express a preference for final model behavior. Optimizing for one can involve trade-offs with others.



Fairness - imposes constraint on optimization that reflects societal norms of what is fair.



Model fragility and security - deploy secure models that protect user privacy.



Model Compression - compact machine learning models to work in resource constrained environments.

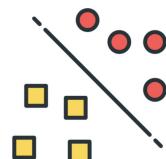
How does model compression trade-off against other properties we care about such as robustness and fairness?



Fairness - imposes constraint on optimization that reflects societal norms of what is fair.



Model fragility and security - deploy secure models that protect user privacy.



Model Compression - compact machine learning models to work in resource constrained environments.

Recent work w colleagues has focused on understanding what are the trade-offs between different desiderata.

What Do Compressed Deep Neural Networks Forget?

Sara Hooker * Aaron Courville Gregory Clark Yann Dauphin Andrea Frome
Google Brain MILA Google Google Brain Google Brain

Abstract

Deep neural network pruning and quantization techniques have demonstrated it is possible to achieve high levels of compression with surprisingly little degradation to test set accuracy. However, this measure of performance conceals significant differences in how different classes and images are impacted by model compression techniques. We find that models with radically different numbers of weights have comparable top-line performance metrics but diverge considerably in behavior on a narrow subset of the dataset. This small subset of data points, which we term Compression Identified Exemplars (CIE), are often outliers in the distribution due to the introduction of sparsity. Compression disproportionately impacts model performance on the underrepresented long-tail of the distribution. PEs over-index on atypical or noisy images that are far more challenging for both humans and algorithms to classify. Our work provides intuition into the role of capacity in deep neural networks and the trade-offs incurred by compression. An understanding of this disparate impact is critical given the widespread deployment of compressed models in the wild.

CHARACTERISING BIAS IN COMPRESSED MODELS

Sara Hooker *
Google Research
shooker@google.com

Nyalleng Moorosi *
Google Research
nyalleng@google.com

Gregory Clark
Google
gregoryclark@google.com

Samy Bengio
Google Research
bengio@google.com

Emily Denton
Google Research
dentone@google.com

ABSTRACT

The popularity and widespread use of pruning and quantization is driven by the severe resource constraints of deploying deep neural networks in environments with strict latency, memory and energy requirements. These techniques achieve high levels of compression with negligible impact on top-line metrics (top-1 and top-5 accuracy). However, overall accuracy hides disproportionately high errors on a small subset of examples; we call this subset Compression Identified Exemplars (*CIE*). We further establish that for *CIE* examples, compression amplifies existing algorithmic bias. Pruning disproportionately impacts performance on underrepresented classes, which often coincides with considerations of fairness. Given that *CIE* are relatively small subsets of the data, and the cost of error in the model, we propose its use as a human-in-the-loop auditing tool to surface a tractable subset of the dataset for further inspection or annotation by a domain expert. We provide qualitative and quantitative support that *CIE* surfaces the most challenging examples in the data distribution for human-in-the-loop auditing.

The Low-Resource Double Bind: An Empirical Study of Pruning for Low-Resource Machine Translation

Orevaaghene Ahia
Masakhane NLP
oreva.ahia@gmail.com

Julia Kreutzer
Google Research
Masakhane NLP
jkreutzer@google.com

Sara Hooker
Google Research, Brain
shooker@google.com

Abstract

A “bigger is better” explosion in the number of parameters in deep neural networks has made it increasingly challenging to make state-of-the-art networks accessible in compute-restricted environments. Compression techniques have emerged as a promising way to bridge the gap. However, evaluation of the trade-offs incurred by popular compression techniques has been centered on high-resource datasets. In this work, we instead consider the impact of compression in a data-limited regime. We introduce the term *low-resource double bind* to refer to the co-occurrence of data limitations and compute resource constraints. This is a common setting for NLP in low-resource languages, yet the trade-offs in

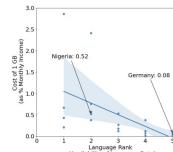


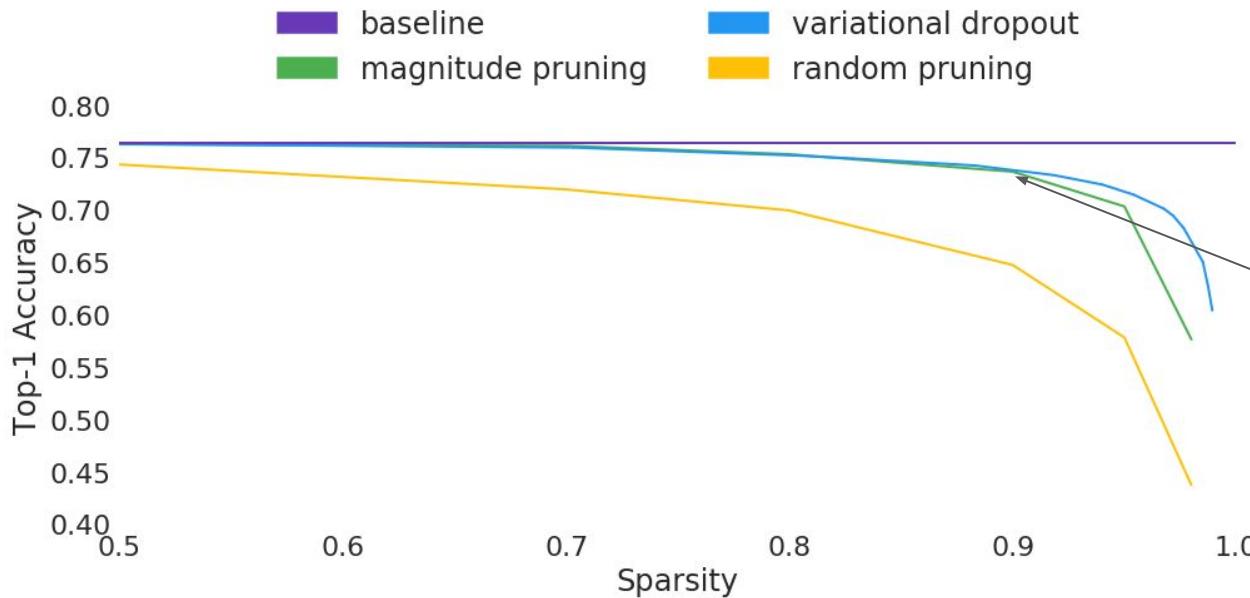
Figure 1: Cost of mobile data by country per language rank according to the taxonomy by Joshi et al. (2020).

[[Hooker et al. 2019, Hooker, Moorosi et al. 2020, Ahia et al. 2021]]

Sparsity is a useful mechanism to understand the relationship between weights and generalization.

- 1) We are comparing models which achieve similar top line metrics regime despite radically different numbers of parameters (unlike dense models at varied parameter counts).
- 2) We can precisely vary level of active weights.
- 3) Sparse models easily outcompete dense models at the same parameter counts

1. We are comparing models in similar top-line regimes

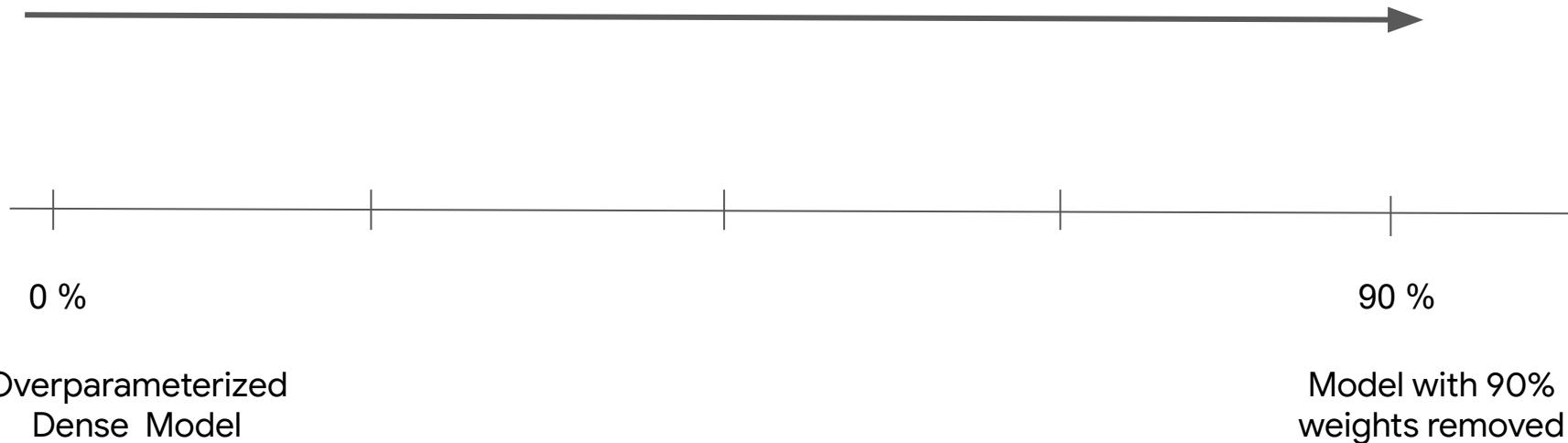


With 90% of the weights removed, a ResNet-50 only loses ~3% of performance

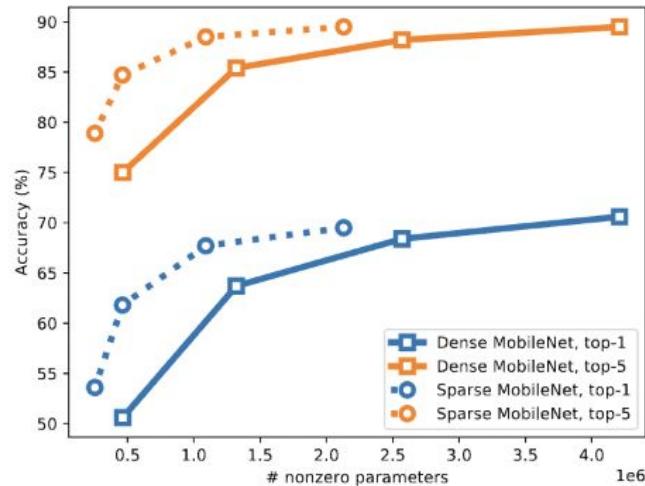
2. We can precisely vary the level of active weights

Experimental Framework

Train populations of models with minimal differences in test-set accuracy to different end sparsities [0%, 30%, 50%, 70%, 90%, **95%**, **99%**].

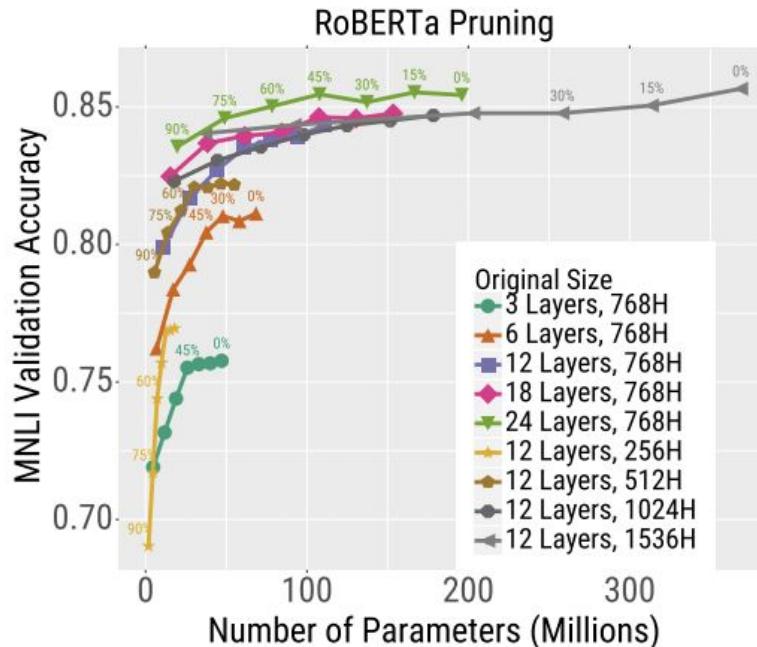


3. Sparse models easily **outperform** dense models with same parameter count.



(a)

[Zhu et al. 2017](#)



[Li et al. 2019](#)

This hold across modalities – computer vision, language, audio – at different scales of available data and parameter ranges.

Size	%Sparse	# Params	yo		ig		ha		de	
			Random	Global	Random	Global	Random	Global	Random	Global
Tiny	0	14.0M	20.39	30.11	28.79	31.78	27.74	31.84	15.40	28.05
Base	0	46.1M	28.45	39.01	36.68	40.33	33.81	37.49	23.80	36.14
	50	23.1M	28.83	38.96	37.05	40.02	33.82	37.61	23.95	36.60
	60	18.6M	28.42	39.14	36.69	39.86	33.87	37.51	24.34	36.32
	70	13.8M	28.59	38.90	36.89	39.92	34.20	37.74	24.84	36.41
	80	9.4M	28.66	38.57	37.19	38.77	34.33	37.49	24.47	36.07
	90	4.6M	27.42	37.01	35.93	36.93	33.20	32.05	23.08	34.96
	95	2.5M	24.08	33.97	32.46	34.58	30.52	34.09	19.78	32.21
	98	1.1M	18.34	27.97	26.02	30.32	24.99	30.09	13.78	25.58

Table 8: Number of non-zero parameters and test BLEU scores under the Limited training regime.

Machine Learning
translation

**Highly sparse
transformer
(2.5M weights)
outperforms tiny
dense (14M)**

Here, we ask - How does model behavior diverge as we vary the level of compression?

1.

Robustness to certain types of distribution shift.

2.

Measure divergence in class level and exemplar classification performance.

Key results upfront: top level metrics hide critical differences in generalization between compressed and compressed populations of models.

1.

Compressed models have amplified sensitivity to adversarial examples and common corruptions.

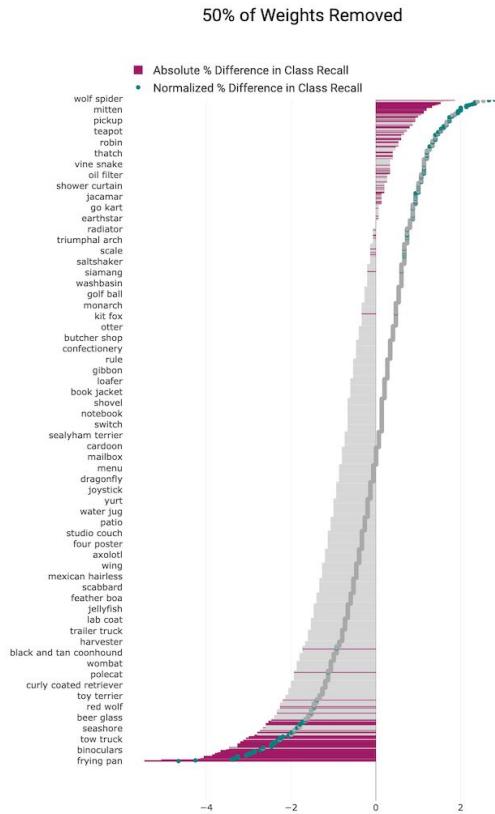
2.

Varying capacity disproportionately and systematically impact a small subset of classes and exemplars.



Why is a narrow part of the data distribution far more sensitive to varying capacity?

A compressed model cannibalizes performance on a small subset of classes to preserve and sometimes even improve relative performance on others.



The loss in generalization caused by pruning is far more concentrated than the relative gains, with fewer classes bearing the brunt of the degradation caused by weight removal.

[Hooker et al, 2019](#)

Pruning Identified Exemplars (PIEs)

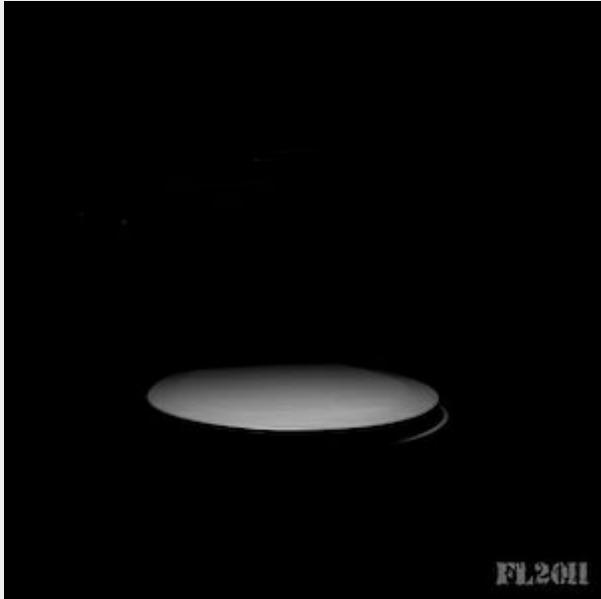
Data points where predictive behavior diverges between a population of independently trained compressed and non-compressed models.

plastic bag



Non-PIE

PIE

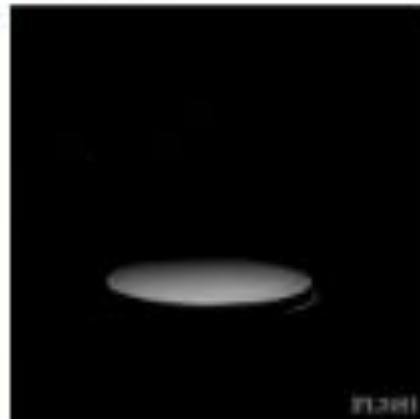


**ImageNet test-set.
True label?**

toilet seat



Non-PIE



PIE



**ImageNet test-set.
True label?**

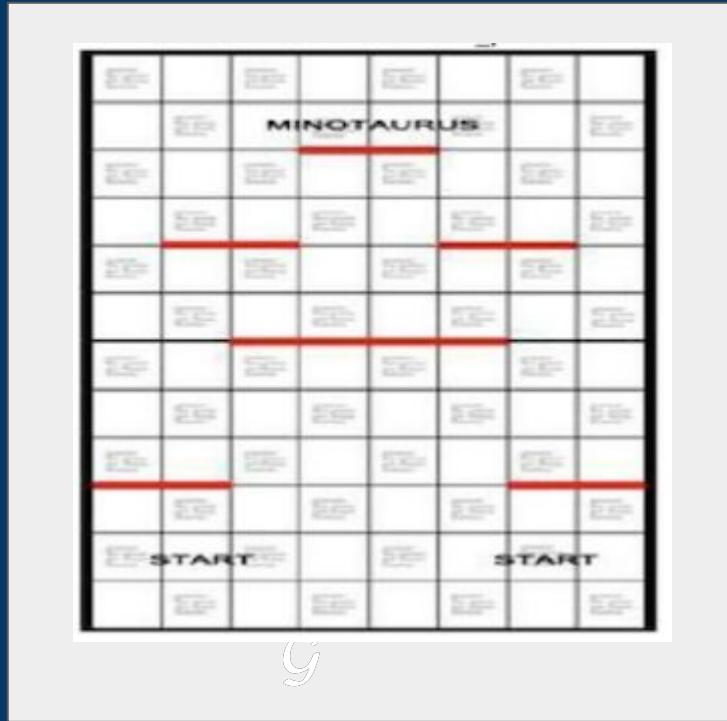
espresso



Non-PIE



PIE



ImageNet test-set.
True label?

maze



Non-PIE



PIE



**ImageNet test-set.
True label?**

wool



Non-PIE

PIE



**ImageNet test-set.
True label?**

matchstick



Non-PIE

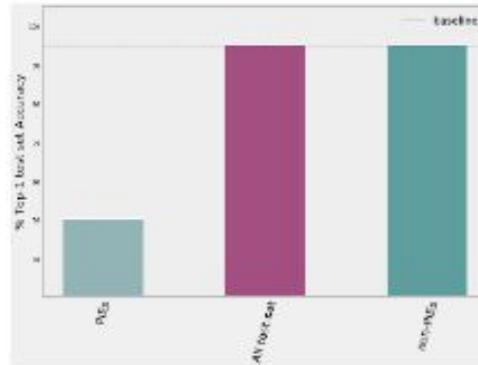


PIE

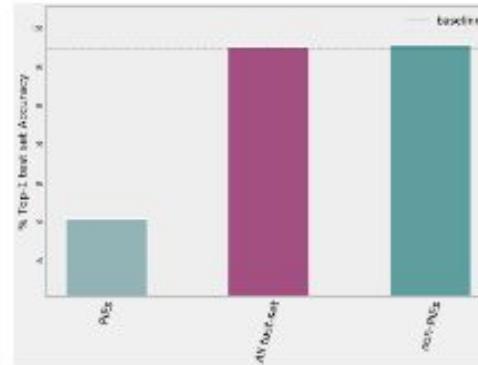
PIEs are also more challenging for a non-compressed algorithm to classify.

Top-1 Accuracy on PIE, All Test-Set, Non-PIE

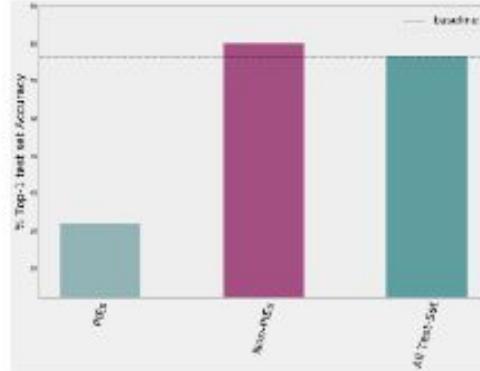
CelebA



CIFAR-10



ImageNet

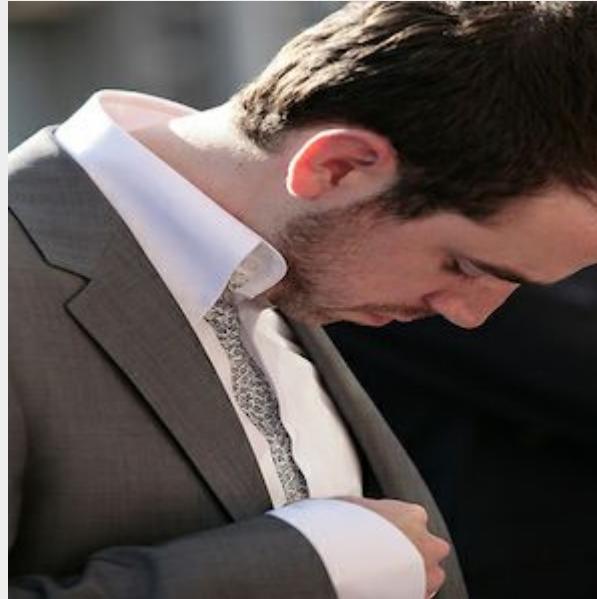


- Restricting inference to PIEs drastically degrades model performance for dense models.
- For ImageNet, removing PIEs from test-set improves top-1 accuracy beyond baseline.

Human study (85 participants) to tag metadata of both PIE and non-PIE

PIE heavily over-index on noisy Data Points

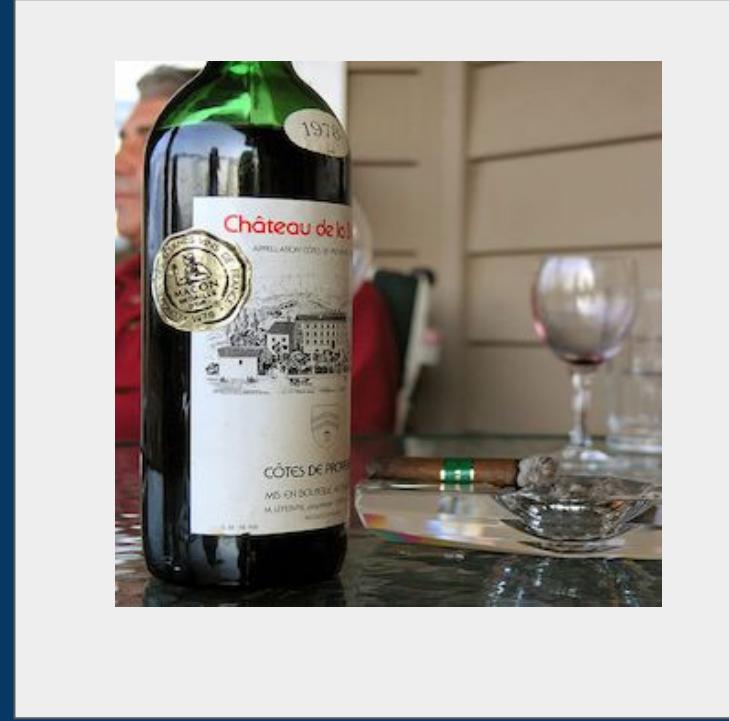
- Data is improperly structured which corrupts information
 - Mislabelled
 - Severely corrupted
 - Multi-object



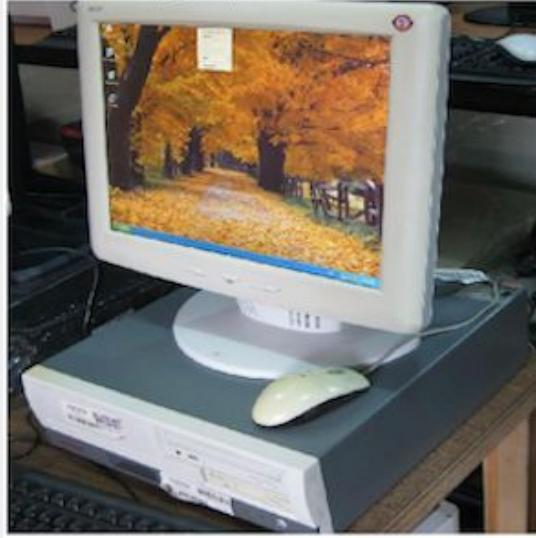
groom

suit

Red wine



Wine bottle

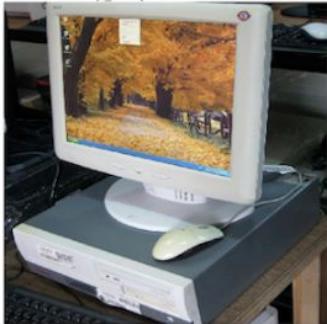


desktop

screen

monitor

PIEs over-index on improperly structured data points: multi-object images for single-image classification.



True Label:
desktop computer
Non-Pruned:
screen
Pruned: monitor



True Label:
wine bottle
Non-Pruned:
red wine
Pruned: wine
bottle



True Label:
groom
Non-Pruned:
groom
Pruned: suit



Sleeping bag

Apron

Bib



guacamole

meatloaf

restaurant

PIEs over-index on Corrupted or incorrectly labeled data.



True Label:
restaurant

Non-Pruned:
meat loaf

Pruned: guacamole



True Label:
envelope

Non-Pruned:
dumbbell

Pruned: maraca



True Label:
tub

Non-Pruned:
cauldron

Pruned: wok

Human study (85 participants) to tag metadata of both PIE and non-PIE

PIE heavily over-index on noisy Data Points

- Data is improperly structured which corrupts information
 - Mislabelled
 - Severely corrupted
 - Multi-object

Atypical Data Points or Challenging Exemplars

- Underrepresented vantage points (the long-tail of the dataset)
- Image classification entails fine grained task

PIEs over-index on Unusual vantage points of the class category.



True Label:

toilet seat

Non-Pruned:

toilet seat

Pruned: folding
chair



True Label:

bathtub

Non-Pruned:

bathtub

Pruned: cucumbe



True Label:

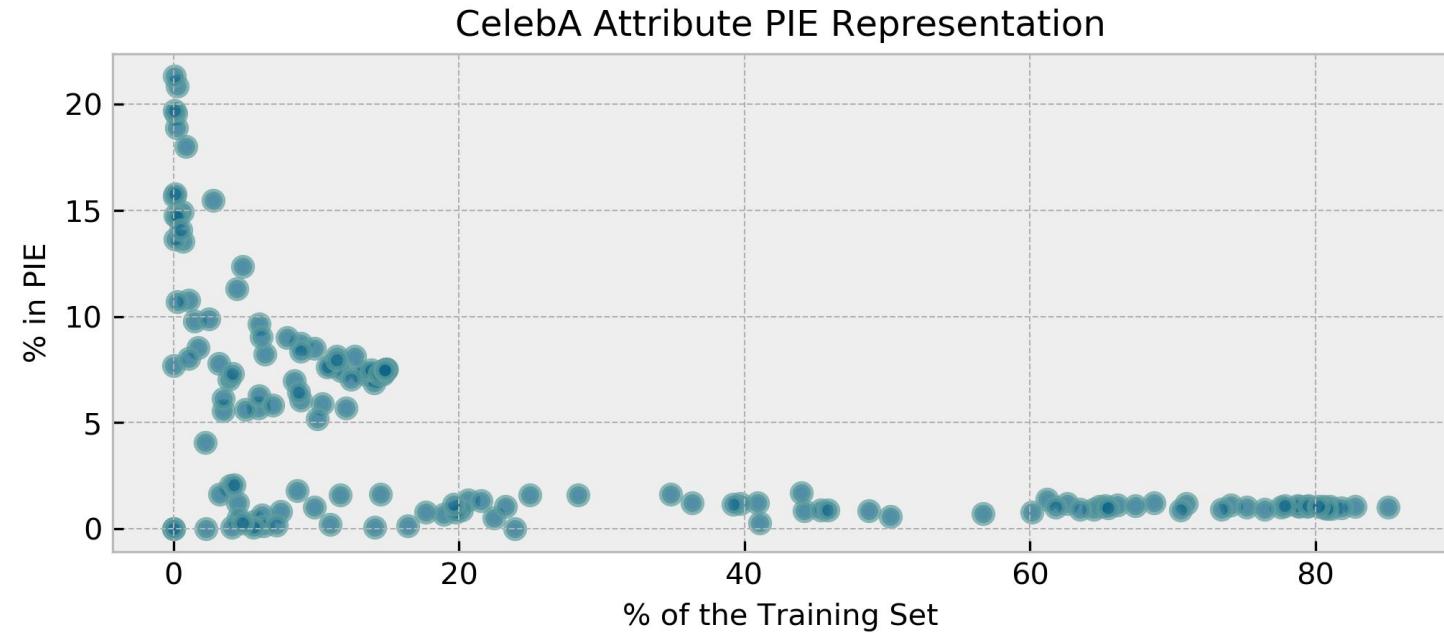
plastic bag

Non-Pruned:

gown

Pruned: plastic
bag

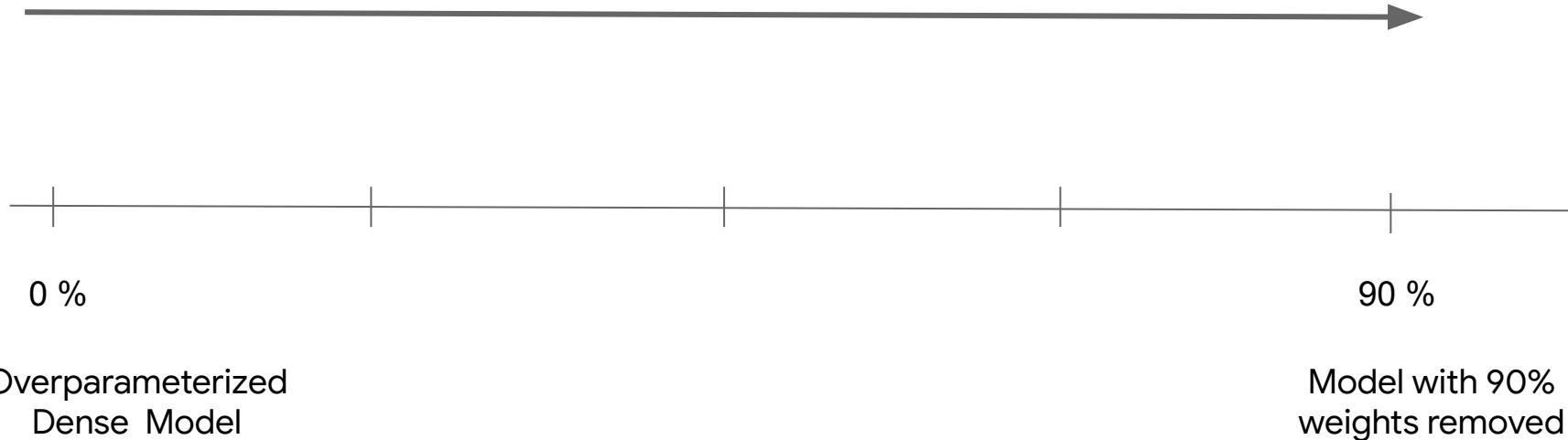
PIEs over-index on the long-tail of underrepresented attributes.



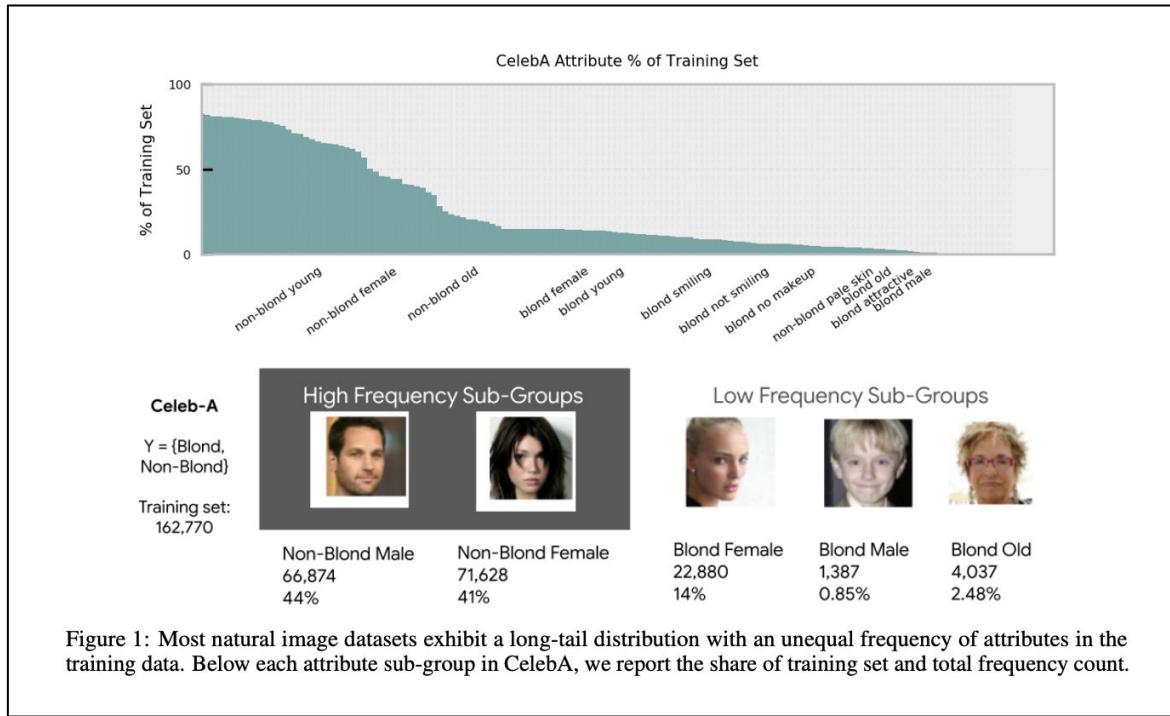
Attribute Proportion of CelebA Training Data vs. relative representation in PIE

It is worth emphasizing this finding: We lose the long-tail when we remove the majority of all training weights.

Put differently, we are using the majority of our weights to encode a useful representation for a small fraction of our training distribution.



The majority of weights used to represent underrepresented features.



Other model design choices which can impact harm:

Differential privacy guarantees amplify disparate error.

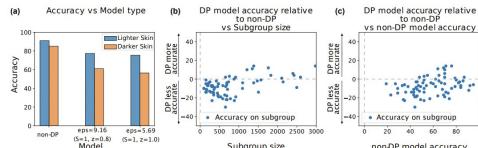


Figure 1: Gender and age classification on facial images.

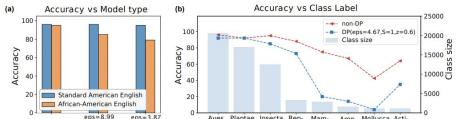
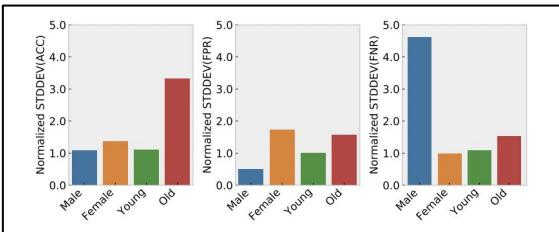


Figure 2: Sentiment analysis of tweets and species classification.

Stochasticity introduced from choice of hardware can amplify harm.



[Bagdasaryan et al.](#)

[Rücklé et al. 2021](#),
[Pfeiffer et al. 2021](#)

Ensembles disproportionately benefit worst-k error and undersampled groups.

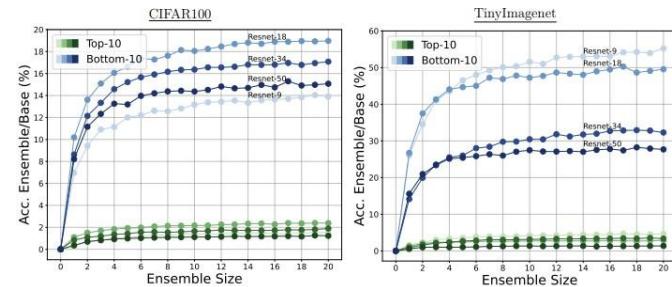


Figure 1: Relative Accuracy for Top-K/Bottom-K. Plot of the ratio of the ensemble accuracy over a single base model (y-axis) illustrates strong benefits for the minority group of ensembling (bottom-k) while the majority group (top-k) only marginally benefits.

[Ko, D'Souza et al. 2022](#)

Where does that leave us?

There are many open challenges and interesting problems to work on in trustworthy ml.

Many of them are core to machine learning:

- How do we train models that are robust to distribution shifts?
- Understanding trade-offs between objectives
- Design algorithms that work in data limited regimes.
- Design auditing techniques at scale
- Discern between different sources of uncertainty,

This has also taken on renewed importance because we are in a time where our ideas are being used by millions of people.

Let's pause here.

Questions?

On the Challenges of Using Black-Box APIs for Toxicity Evaluation in Research Luiza Pozzobon, Beyza Ermis, Patrick Lewis, Sara Hooker [\[\[link\]\]](#)

Metadata Archaeology: Unearthing Data Subsets by Leveraging Training Dynamics Shoaib Ahmed Siddiqui, Nitashan Rajkumar, Tegan Maharaj, David Krueger, Sara Hooker [\[\[link\]\]](#)

A Benchmark for Interpretability Methods in Deep Neural Networks Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, Been Kim, [\[\[link\]\]](#)

Estimating Example Difficulty using Variance of Gradients Chirag Agarwal*, Sara Hooker* [\[\[link\]\]](#)

What do compressed deep neural networks forget?, Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, Andrea Frome [\[\[link\]\]](#)

Characterizing Bias in Compressed Models Sara Hooker*, Nyalleng Moorosi*, Gregory Clark, Samy Bengio, Emily Denton [\[\[link\]\]](#)

Final takeaways:

Beyond test-set accuracy - It is not always possible to measure the trade-offs between criteria using test-set accuracy alone.

The myth of the compact, private, interpretable, fair model - Desiderata are not independent of each other. Training beyond test set accuracy requires trade-offs in our model preferences.

Understanding the interactions between model and dataset can open up new mitigation strategies.

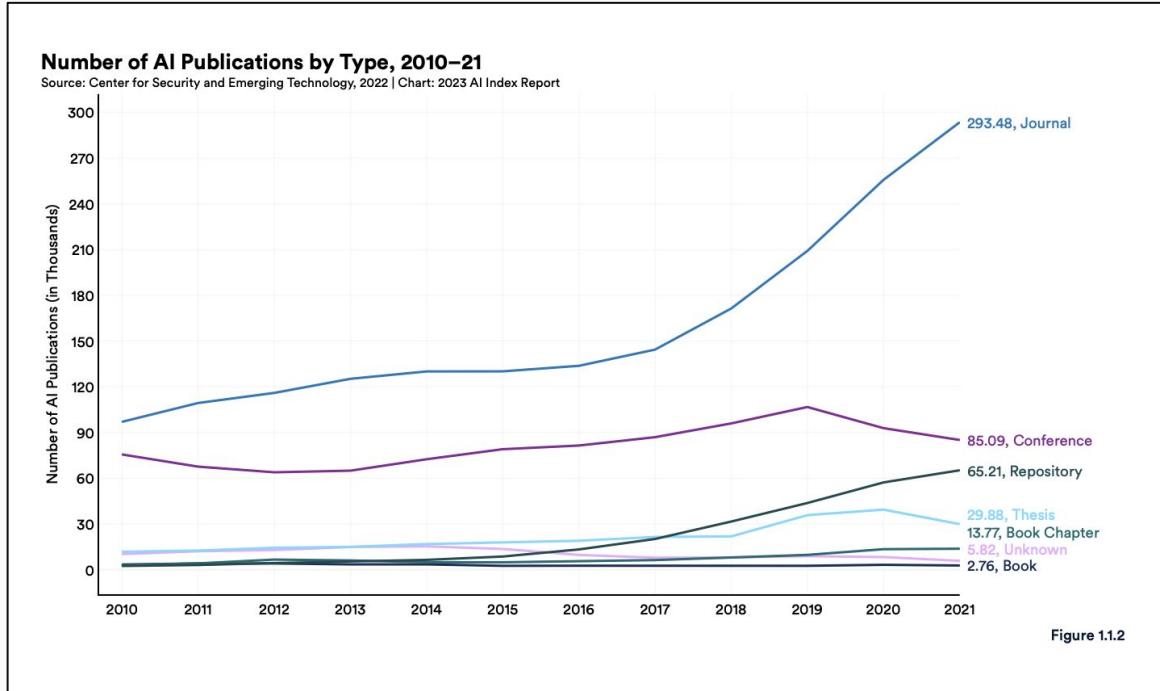
Part III: Open problems in how we build technology



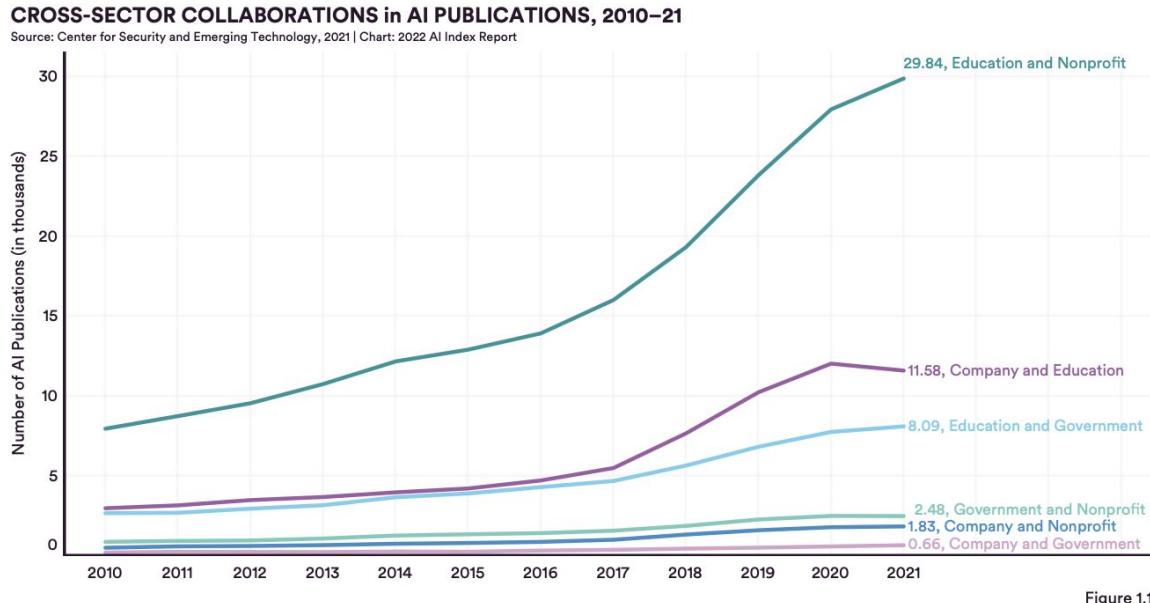
MLRS 2023
Sara Hooker - Cohere For AI

Right now NLP models are being
built by a small group of people.
However, impact is felt
everywhere.

While number of overall publications in our field has increased, who is producing research remains narrow.



Cross-institutional collaborations are rising, yet typically occur more between certain types of institutions.



2.5x times as many collaborations between educational institutions and nonprofits in 2021 as between educational institutions and companies.

Major geographic disparities remain –

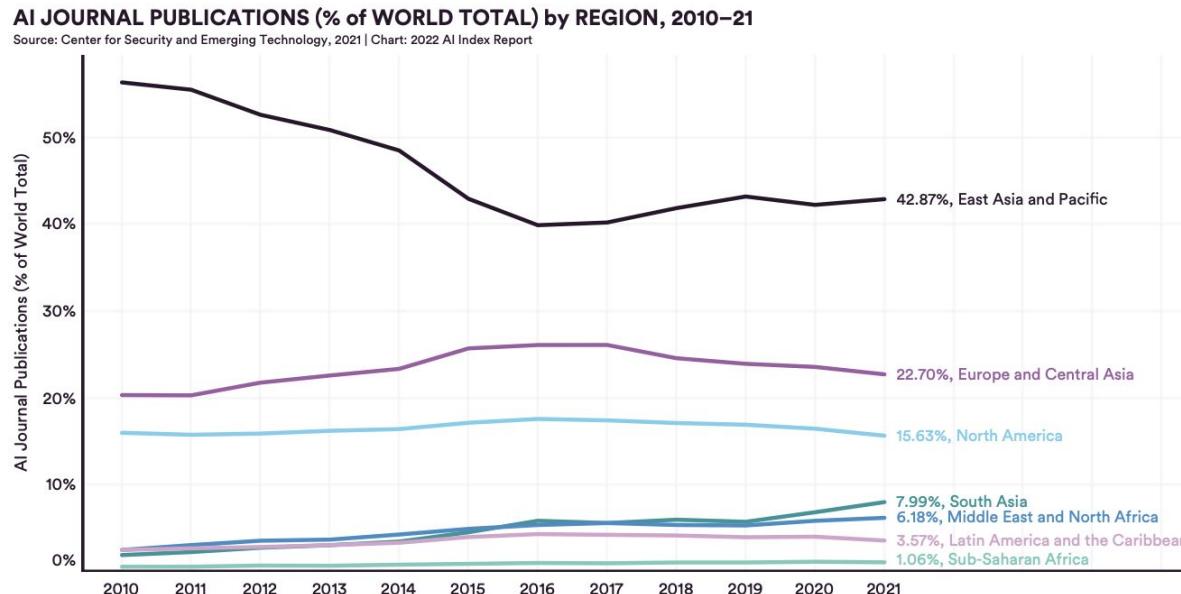


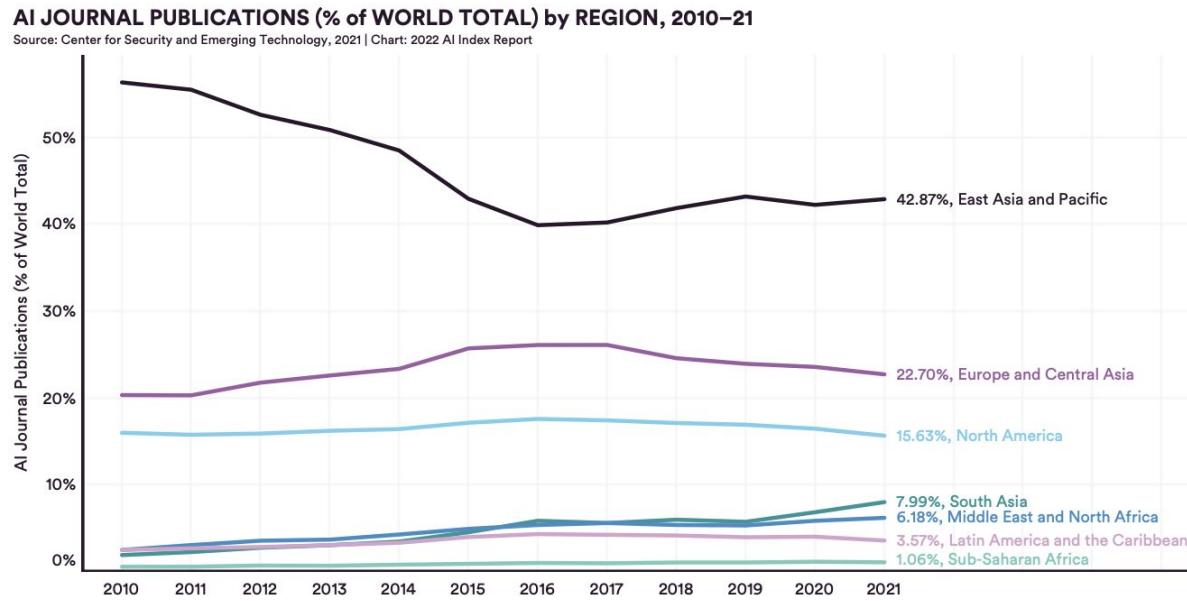
Figure 1.1.9

Large discrepancies persist in who participates in research:

Sub-Saharan Africa
1.06%

Latin America and the Caribbean 3.57%

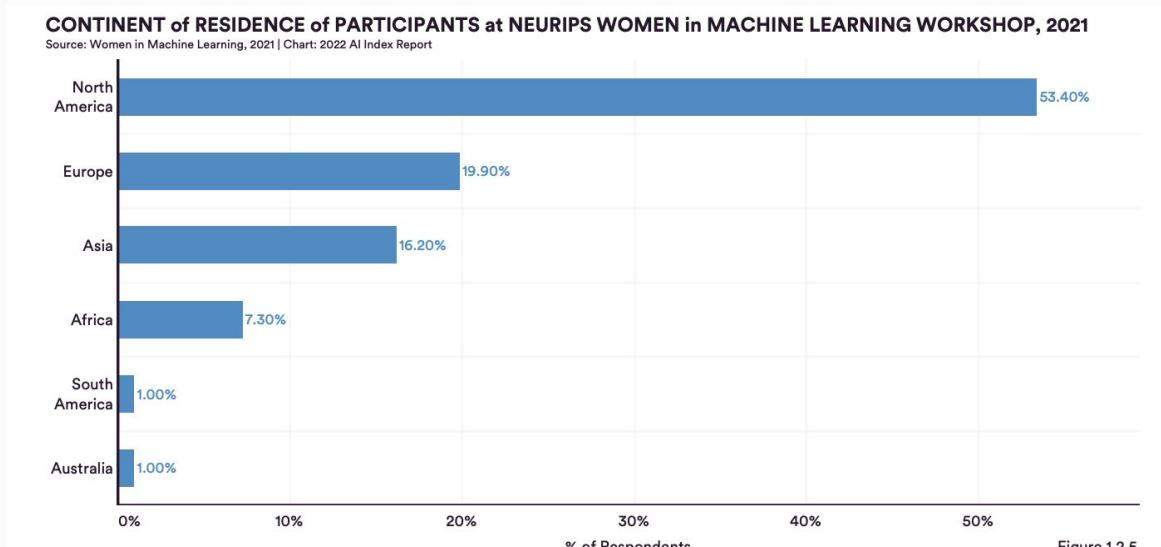
Geographies like South Asia, Middle East and Africa have made recent gains in number of publications.



South Asia and the Middle East and North Africa saw the most significant growth. Number of AI journal publications grew around 12 and 7 times, respectively, in the last 12 years.

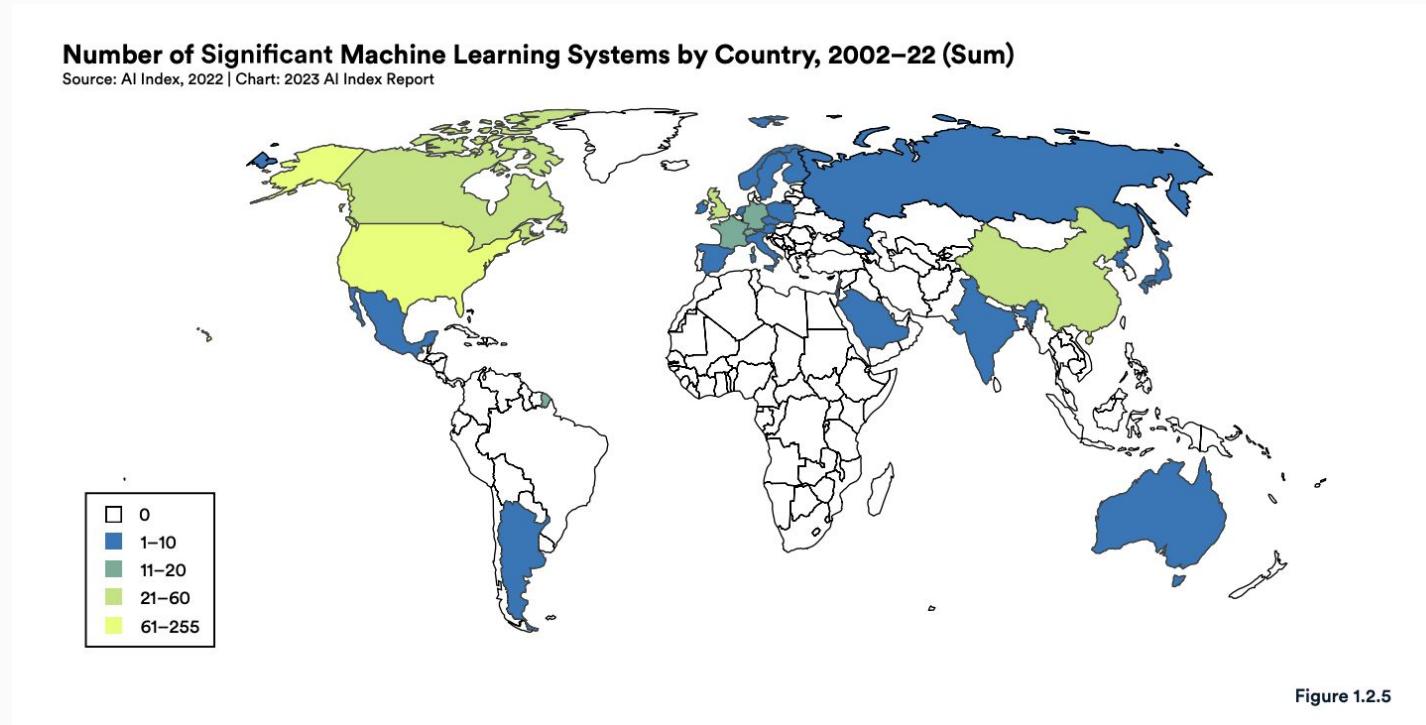
Figure 1.1.9

This geo disparity is also reflected in who attends conferences like NeurIPS.

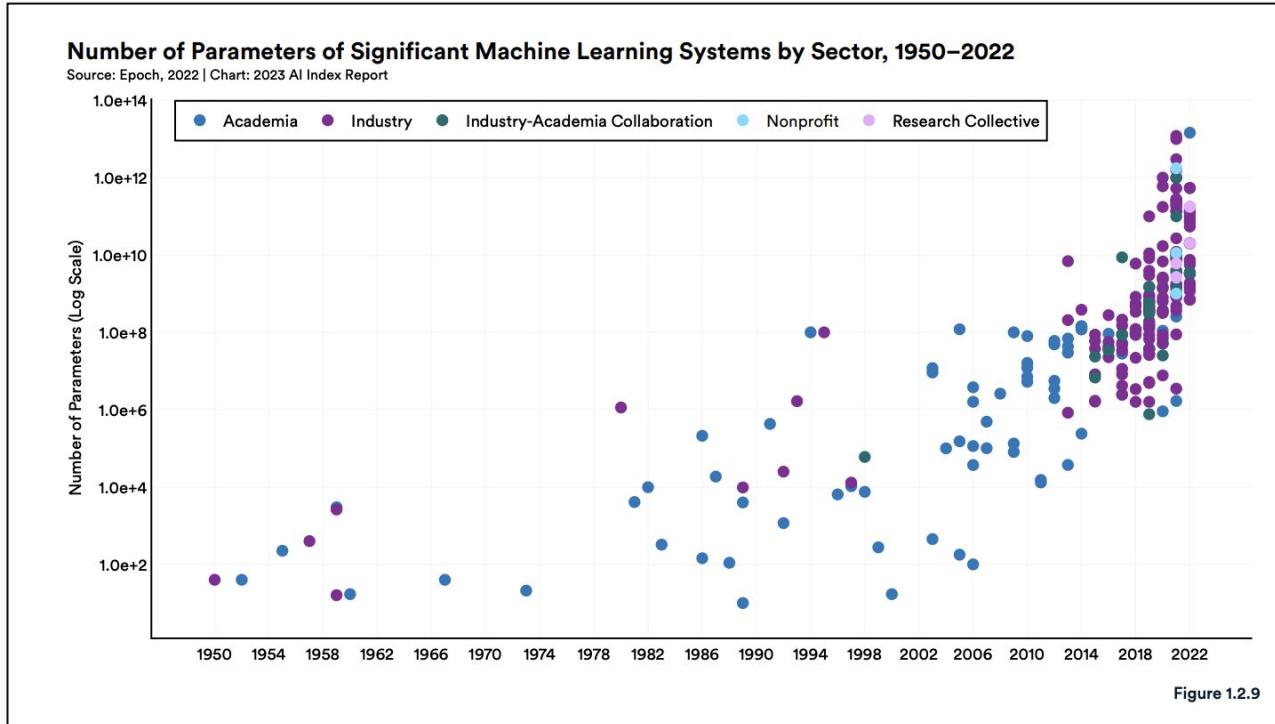


53.40 % of participants are from North America.

Most striking – who participates in significant research breakthroughs is incredibly narrow.



Compute trends also amplify the disparity in who participates. Academia phased out, industry dominates.



So why talk about large language models – because especially when technology is powerful it is important that we are all building it.

“When you are not part of the conversation, it happens to you and not with you.”

Vukosi Marivate

So, let's talk about the state of
NLP research – and also open
challenges and opportunities.

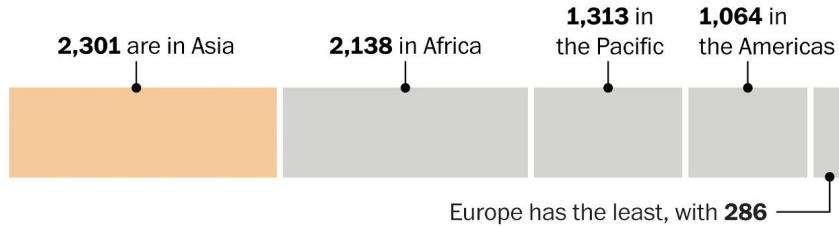
“The limits of my language means the
limits of my world.”

– Ludwig Wittgenstein

There are:

- 7,102 languages in the world
- Around 400 languages have more than 1M speakers and
- around 1,200 languages have more than 100k
- 2000 have fewer than a 1,000 speakers

There are at least **7,102** living languages in the world.



Sources: Ethnologue: Languages of the World, Eighteenth edition THE WASHINGTON POST

At-risk languages

● Critically endangered ● Seriously endangered ● Endangered



Sources: Alliance for Linguistic Diversity, UNESCO
GENE THORP AND KEVIN SCHAUL/THE WASHINGTON POST

Languages are not treated equally by researchers. Some languages have received disproportionate attention and focus in NLP.

Language	# of papers per million speakers	# of speakers (in millions)
Irish	5235	0.2
Basque	2430	0.5
German	179	83
English	63	550
Chinese	11	1,000
Hausa	1.5	70
Nigerian Pidgin	0.4	30

Number of papers in top NLP venues referencing language per 1 million speakers.
[\[Van Etch et al. 2022\]](#)

This uneven coverage also means that many languages have been left out of the technological progress.

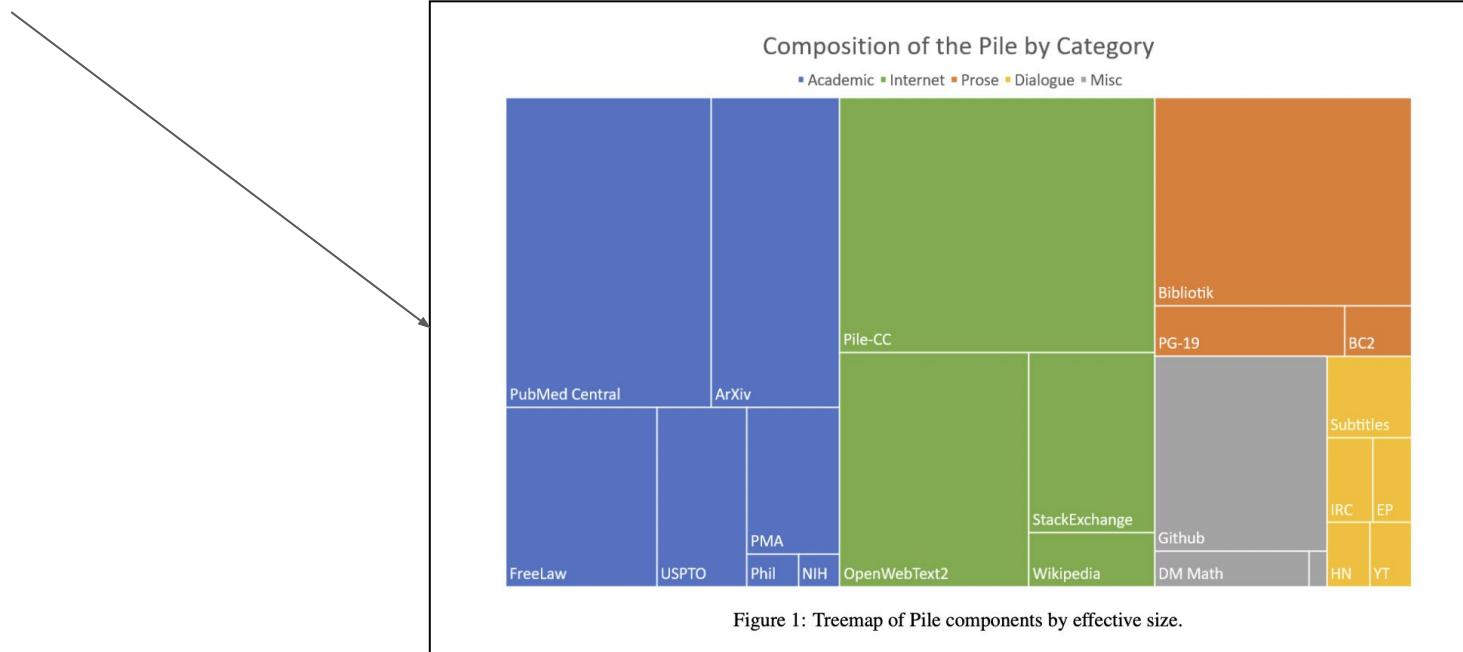
Multilingual Model Name	Number of Languages Trained On (pre-training)
BLOOM	46
mT5	101
XGLM	30



Open source multilingual state of art Large Language Models (LLM) are pre-trained a smaller subset of available languages.

Why have some languages been left behind in technological progress?

Much of our data in large language model training comes from the internet.

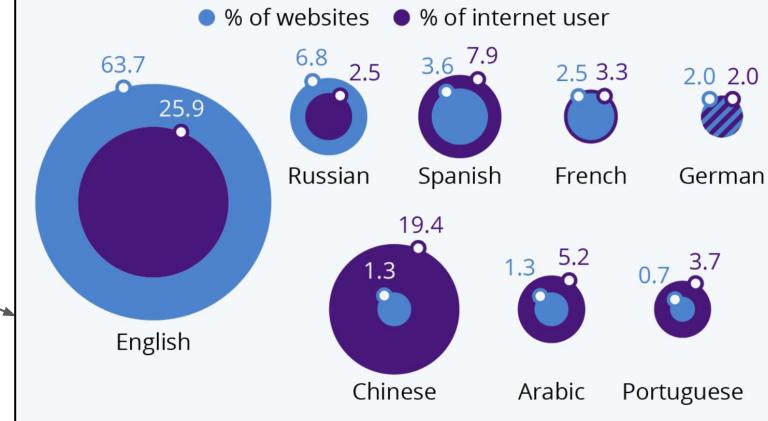


The composition of languages on the internet reflects the composition of early users.

5% of the world speaks English at home, yet 63.7% of internet communication is in English.

English Is the Internet's Universal Language

Share of websites using selected languages vs. estimated share of internet users speaking those languages*

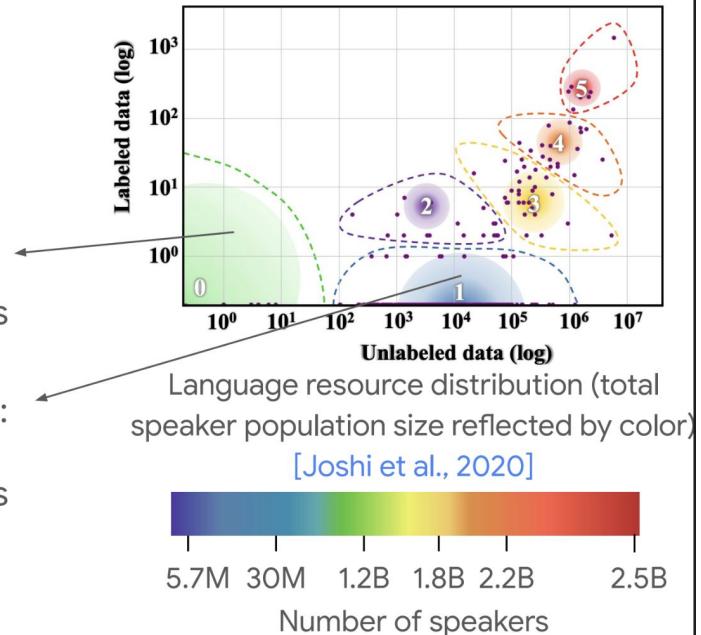


Under-resourced languages may have limited data available....

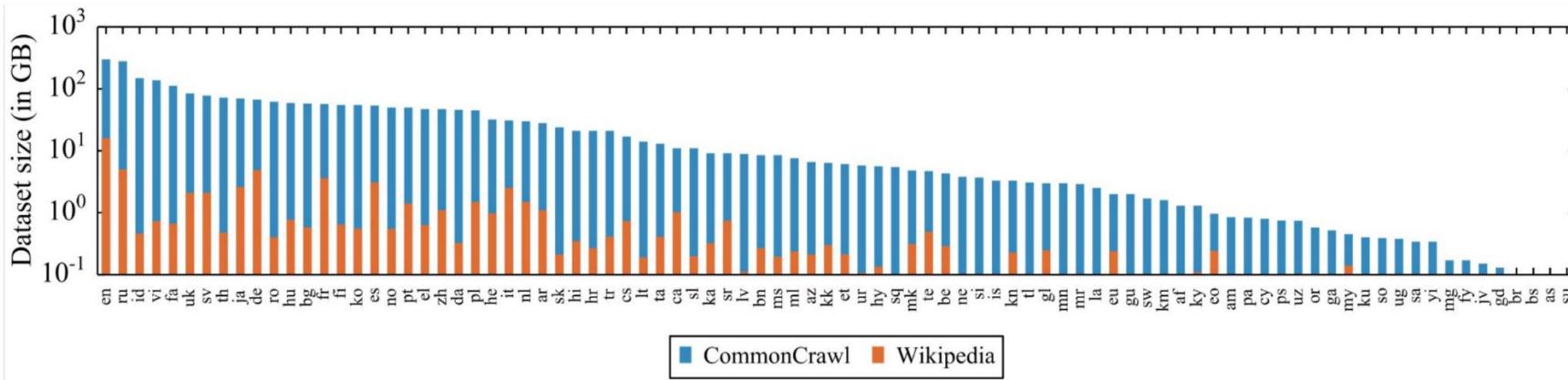
An astounding 80%
of languages have
no-text available =
only ~1400
languages have text
corpus for NLP
modeling.

No-text:
80% of
languages

Few-text:
5% of
languages



The long-tail of multilinguality, few high resource languages and many sparsely populated languages.



Often the data that is available is also low quality...

"44 of the 65 languages that we audited for CCAigned containing under 50% correct sentences, and 19 of the 20 in WikiMatrix."

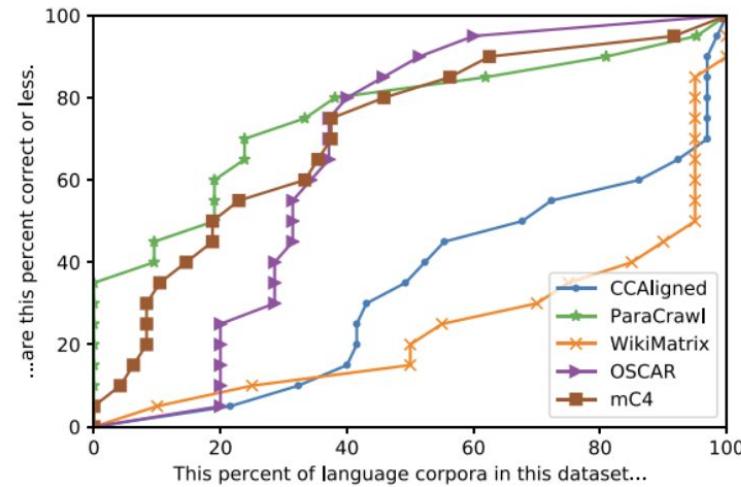
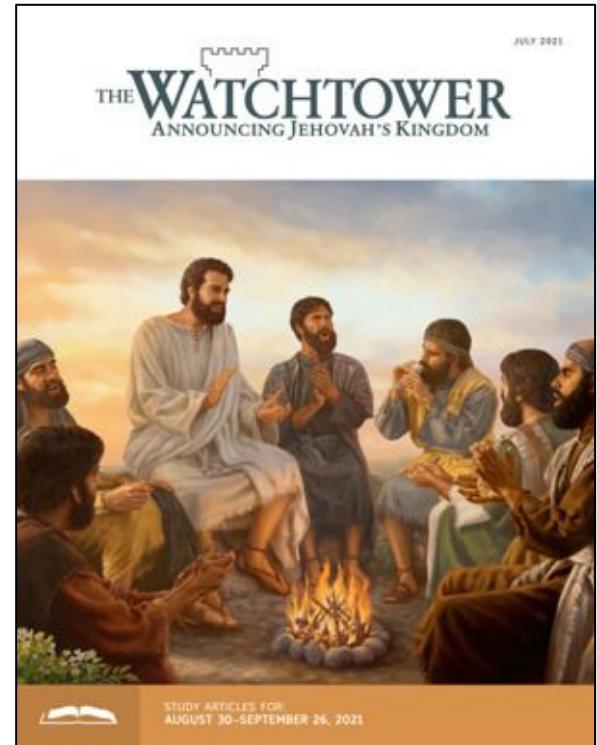


Figure 1: Fraction of languages in each dataset below a given quality threshold (percent correct).

... or may be difficult to generalize from.

One of the most commonly used low resource corpus is JW300 – is very specialized religious corpus. From translated Jehovah witness writings. This leads to very specialized language that may not generalize to other settings we care about.



The under-indexing of certain languages is also driven by access to compute resources.

The double-low resource bind refers to the co-occurrence of limited data availability and high compute costs.

- Mobile data, compute, and other computational resources may often be expensive or unavailable

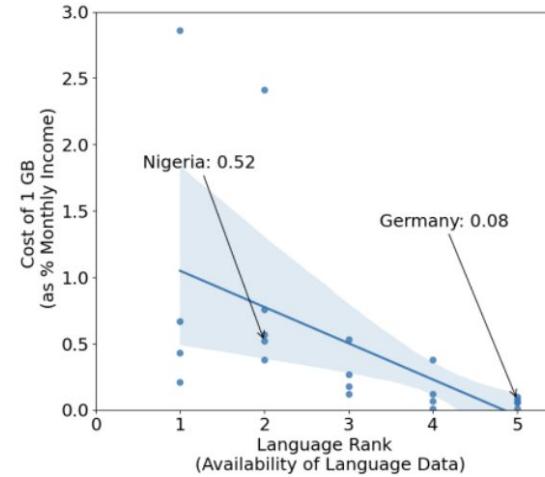
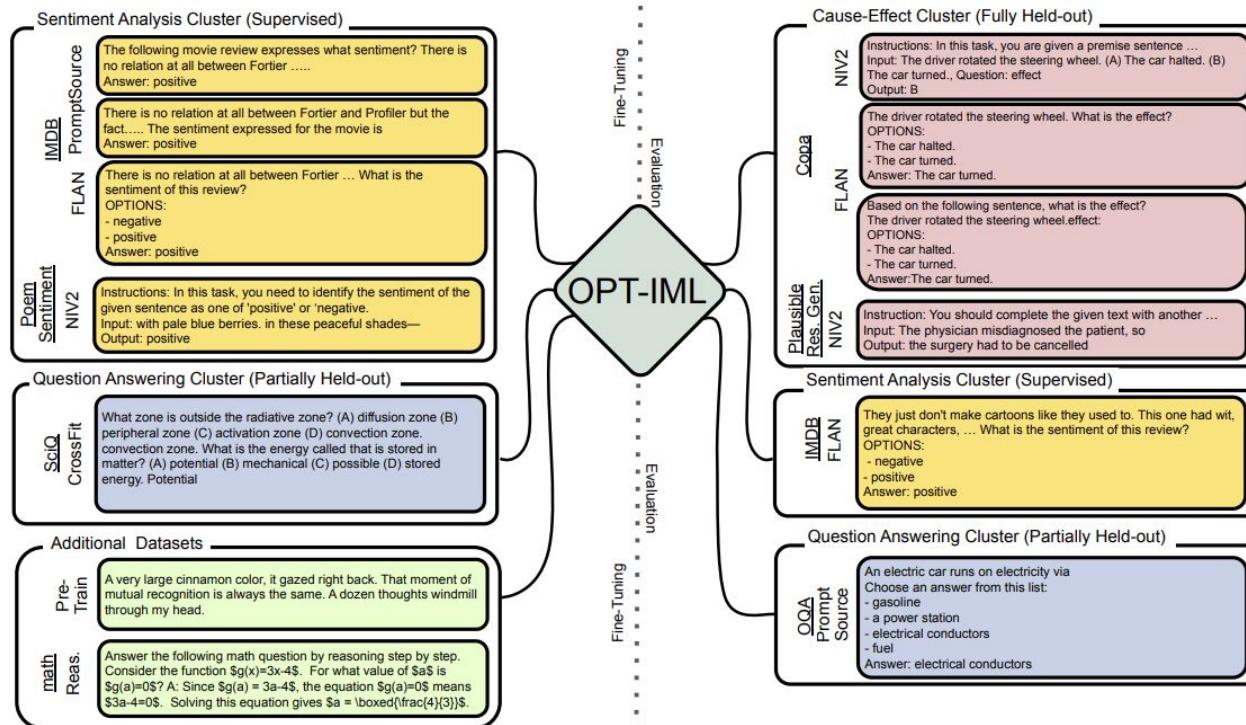
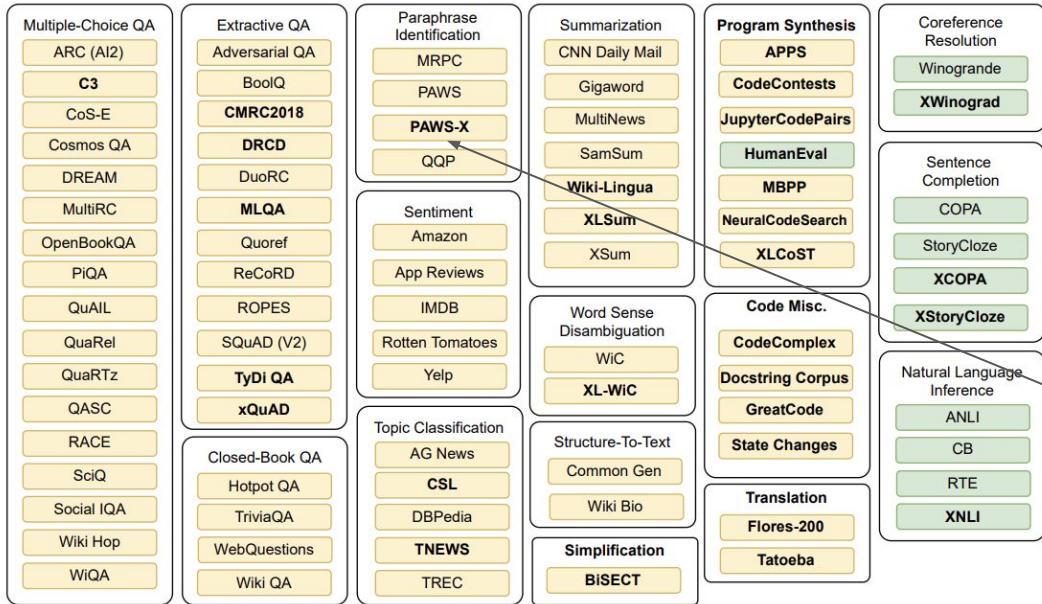


Figure 1: Cost of mobile data by country per language rank according to the taxonomy by Joshi et al. (2020).

While considerable work has focused on English language instruct style datasets, far less has explored the benefits for a multilingual setting.



Most relevant is work released in November 2022 by [Muennighoff et al.](#)



Added multilingual
and program
synthesis datasets
to P3.

Figure 1: An overview of datasets in xP3. Datasets added to P3 in this work are marked **bold**. Yellow datasets are trained on. Green datasets are held out for evaluation.

[Muennighoff et al.](#)

Observed boosts in performance over base multilingual models.

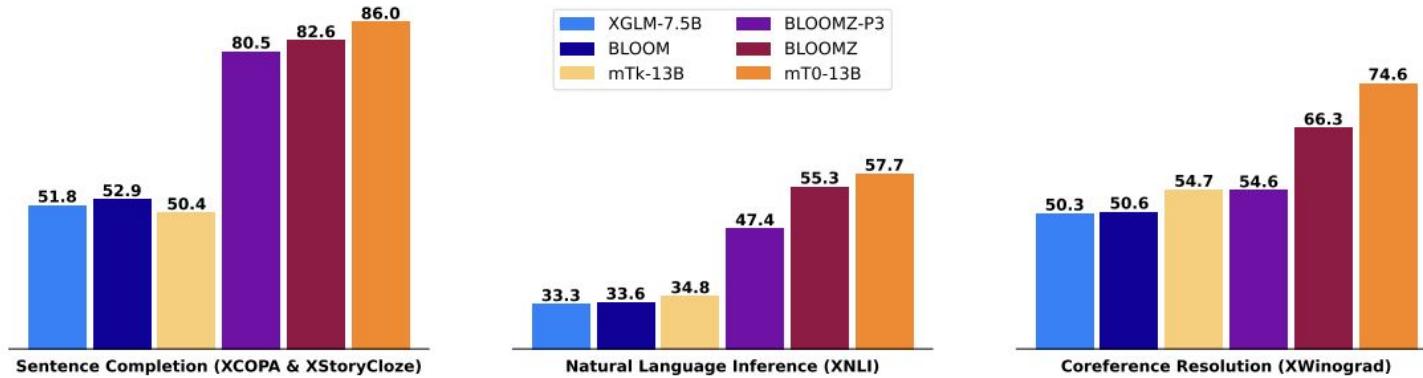


Figure 4: Zero-shot multilingual task generalization with English prompts. BLOOM models have 176 billion parameters. Scores are the language average for each task. Appendix §B breaks down performance by language.

This is despite the datasets added only covering 46 languages, and having no human feedback optimization.

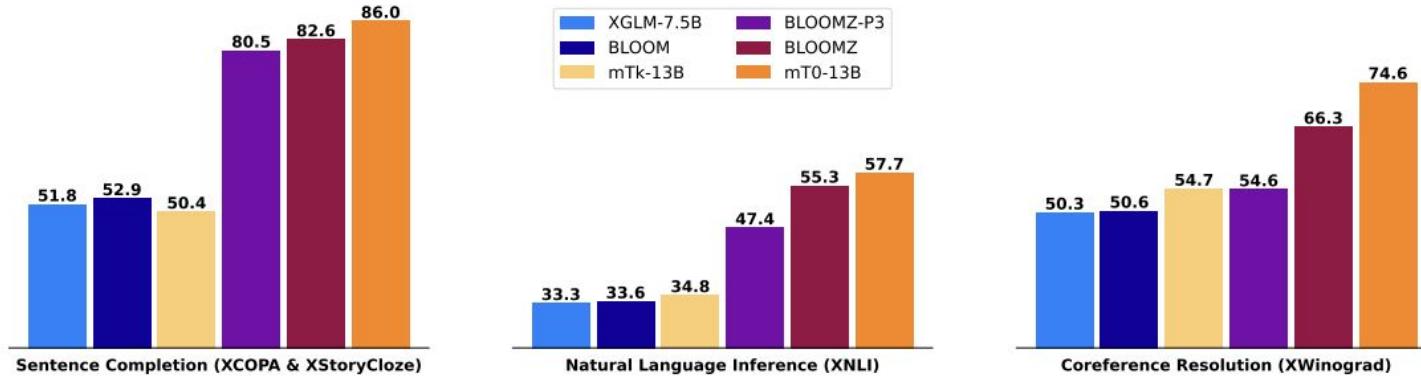
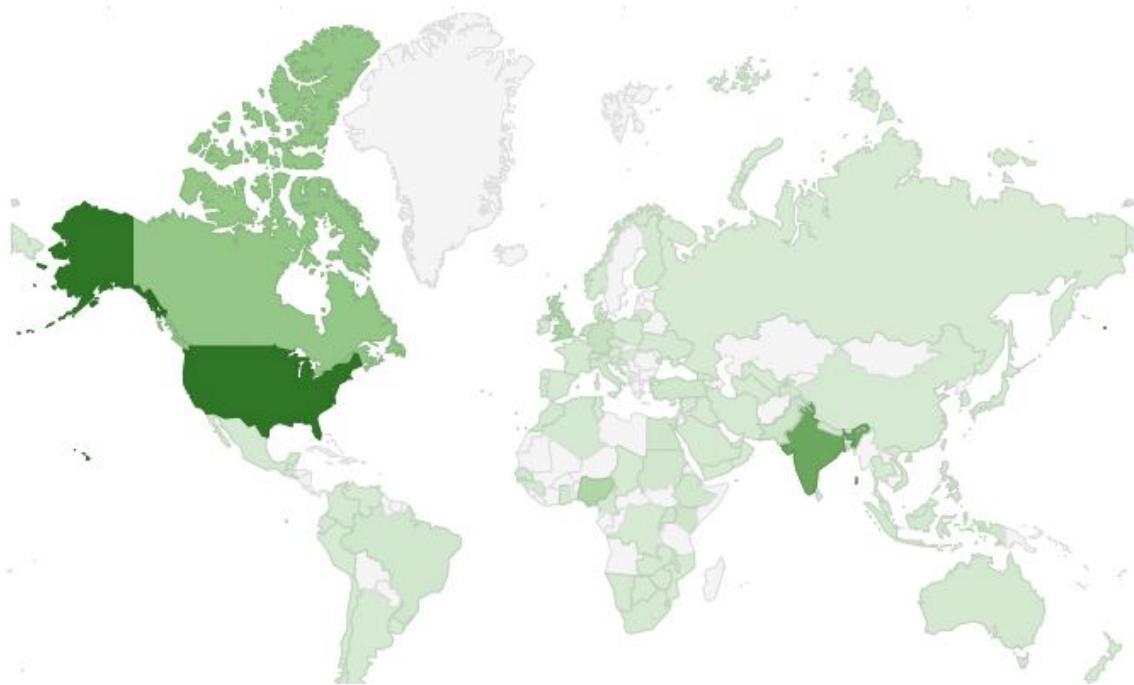


Figure 4: Zero-shot multilingual task generalization with English prompts. BLOOM models have 176 billion parameters. Scores are the language average for each task. Appendix §B breaks down performance by language.

There are plenty of low hanging fruit for improving the performance of multilingual on zero-shot and held-out test set settings.

- Human feedback loop optimization for multilingual
- Native speaker generation of templates for available multilingual datasets
- Auditing the quality of existing multilingual datasets
- Coverage of languages
- Exploring data augmentation strategies.
- Understanding multi-task weighting strategies.

Multi Institutional effort with researchers from industry and academia.
Compute provided by C4AI. Over 100+ countries represented.



Open science commitment – We will release Aya dataset and model.

Community Project – Instruct Multilingual

Cohere For AI

Aya: An Open Science Initiative to Accelerate Multilingual AI Progress

Our goal is to accelerate NLP breakthroughs for the rest of the world's languages through open science collaboration.

Community Project – Instruct Multilingual

Just a step more...

Choose Your Country
Select Country

Choose Your Language
Select Language

Submit

What is Aya?

Recent breakthroughs in NLP technology have focused on English, leaving other languages behind. One of the biggest hurdles to improving multilingual model performance is access to high-quality examples of multilingual text. In January 2023 the Cohere For AI community set out on an ambitious open science research project.

With members from over 100 countries around the world, we sought to leverage the great strength of our diversity to make meaningful contributions to fundamental machine-learning questions. Our ultimate goal is to release a high-quality multilingual dataset. In sharing this artifact broadly, we will support future projects that aim to build technology for everyone, including those who use any of the 7000+ languages spoken around the world. As technological progress races forward, join us to ensure no language is left behind.

You can learn more about the multilingual research project in these slides, and join our next open to all monthly meeting in the discord.

How it Works?

Task 1
You will be asked to rate and edit model data to improve it.
Get Started →

Task 2
You can share your own examples of data that you think is great.
Get Started →

Community Project – Instruct Multilingual

Welcome <USER>

LeaderBoard

Task 1

Task Type

Task 1

Prompt

If you have any additional feedback, please type it in here...

Completion

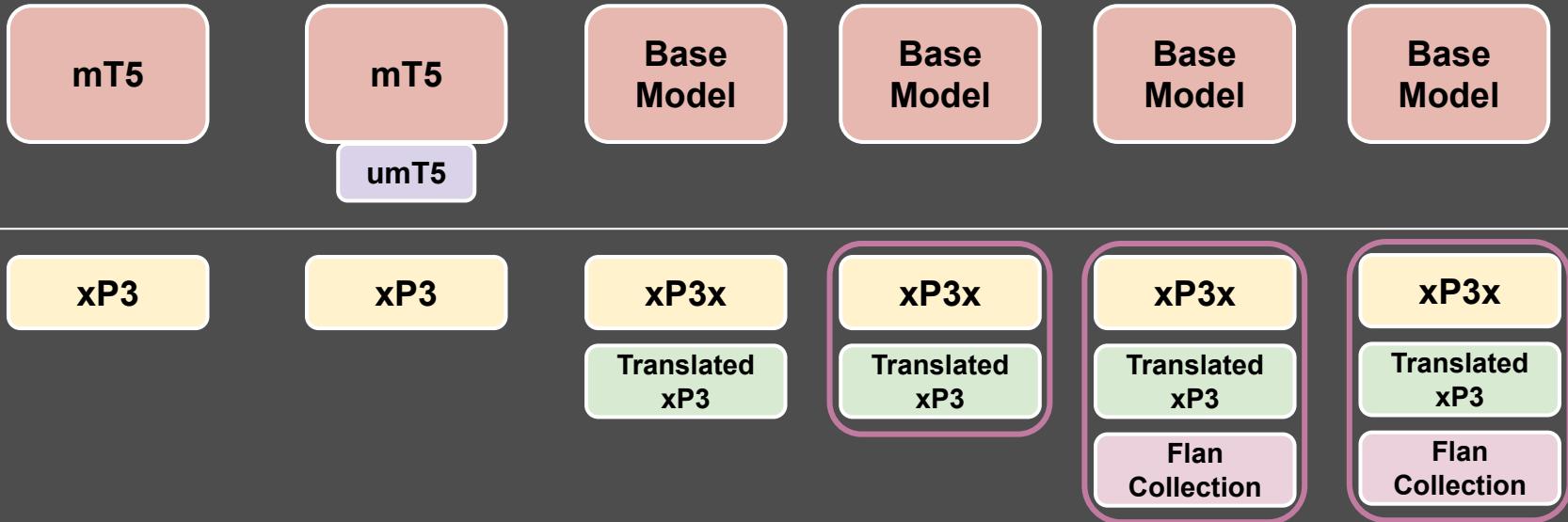
If you have any additional feedback, please type it in here...

Submit feedback

- ★ User login by country and language
- ★ Task selection
- ★ Feedback for prompt/completion
- ★ Leaderboard

Next Experiments

We're here!



Base Model
Comparison

Add Translated
xP3

Few-shot
augmentations

Add Flan
Collection

Language
Balancing

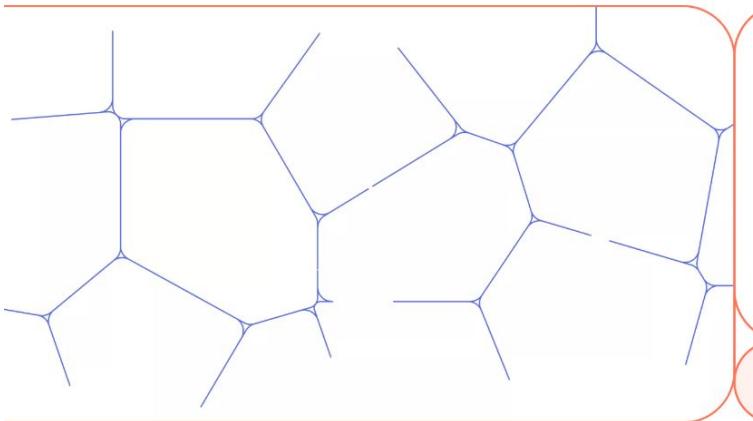
Why are we calling this project AYA?

a fern and is a West African symbol of “endurance, independence, defiance against difficulties, hardiness, perseverance, and resourcefulness.



This is an active open science project – part of our commitment of collaborating widely with other institutions to contribute research.

You can join the open science community if you want to get involved in the project.



Our Open Science Community

We're not just another research group. We are *the open science community* to conduct top-tier ML research while creating more points of entry into the field.

Our research community is a space where researchers, engineers, linguists, social scientists, and lifelong learners connect and collaborate with each other. We come together from over 100 countries around the world and support large and small scale research collaborations.

Join Us ➔

Where does that leave us?

Exciting time to do research:

- Recent breakthroughs have unlocked more interesting problems
- Role of data has again begun to take on renewed importance
- Underexplored directions in multilingual, efficiency (both algorithmic, hardware<>algorithm co-design, in the data space).
- Important reliability issues - hallucinations, ability to update models efficiently.

Let's open up to discussion
and questions.

Questions?

The Low-Resource Double Bind: An Empirical Study of Pruning for Low-Resource Machine Translation, Oreva Ahia, Julia Kreutzer, Sara Hooker [\[\[link\]\]](#)

Intriguing Properties of Compression on Multilingual Models Kelechi Ogueji, Orevaoghene Ahia, Gbemileke Onilude, Sebastian Gehrmann, Sara Hooker, Julia Kreutzer [\[\[link\]\]](#)

AYA project – underway, you can join now in the discord. We need language ambassadors and researchers who want to be involved.

Feel free to reach out if any of these ideas is relevant to work you are doing on your team.

 Cohere For AI

Cohere For AI

Aya: An Open Science Initiative to Accelerate Multilingual AI Progress

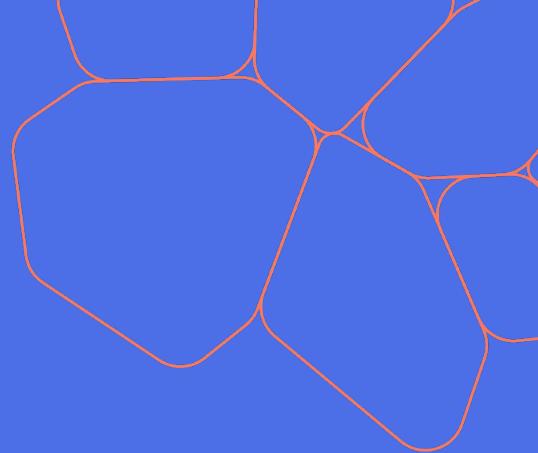
Our goal is to accelerate NLP breakthroughs for the rest of the world's languages through open science collaboration.

 Join Aya Discord

 Go to workspace

Email: sarahooker@cohere.com

Questions?



Feel free to reach
out:

sarahooker@cohere.com

Why is **sparse-to-sparse** training more challenging? If we can remove weights during training, why can't we start sparse?

- Optimization dynamics of how to start sparse is an important area of research [[Frankle et al. 2021](#), [Utku et al. 2022](#)]
- Not just about initialization, activation functions and other architecture design choices have overfit to benefit dense networks over sparse [[\[Tessera et al. 2021\]](#)].

KEEP THE GRADIENTS FLOWING: USING GRADIENT FLOW TO STUDY SPARSE NETWORK OPTIMIZATION

Kale-ab Tessera^{1,3}, Sara Hooker², and Benjamin Rosman¹

¹School of Computer Science and Applied Mathematics, University of the Witwatersrand

²Google Brain

³InstaDeep

kaleabtessera@gmail.com, shooker@google.com, Benjamin.Rosman@wits.ac.za

ABSTRACT

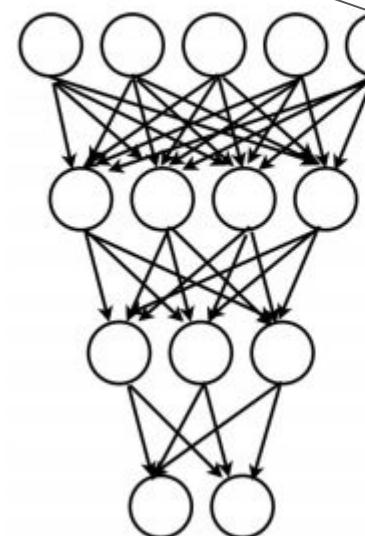
Training sparse networks to converge to the same performance as dense neural architectures has proved to be elusive. Recent work suggests that initialization is the key. However, while this direction of research has had some success, focusing on initialization alone appears to be inadequate. In this paper, we take a broader view of training sparse networks and consider the role of regularization, optimization, and architecture choices on sparse models. We propose a simple experimental framework, *Same Capacity Sparse vs Dense Comparison* (SC-SDC), that allows for a fair comparison of sparse and dense networks. Furthermore, we propose a new measure of gradient flow, *Effective Gradient Flow* (EGF), that better correlates to performance in sparse networks. Using top-line metrics, SC-SDC and EGF, we show that default choices of optimizers, activation functions and regularizers used for dense networks can disadvantage sparse networks. Based upon these findings, we show that gradient flow in sparse networks can be improved by reconsidering aspects of the architecture design and the training regime. Our work suggests that initialization is only one piece of the puzzle and taking a wider view of tailoring optimization to sparse networks yields promising results.

Under the term sparsity, there are a variety of techniques that differ by the unit of removal.

Unstructured pruning removes weights.

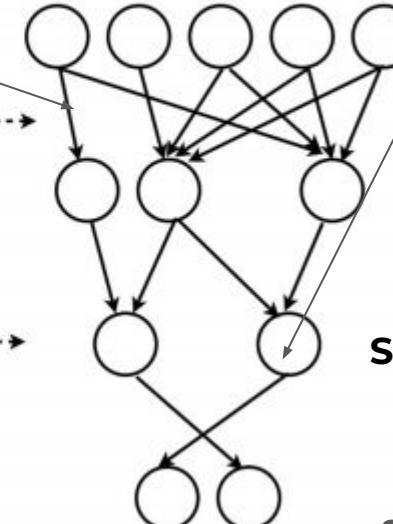
Data pruning removes data points.

Original Network



Pruning
Synapses

Pruning

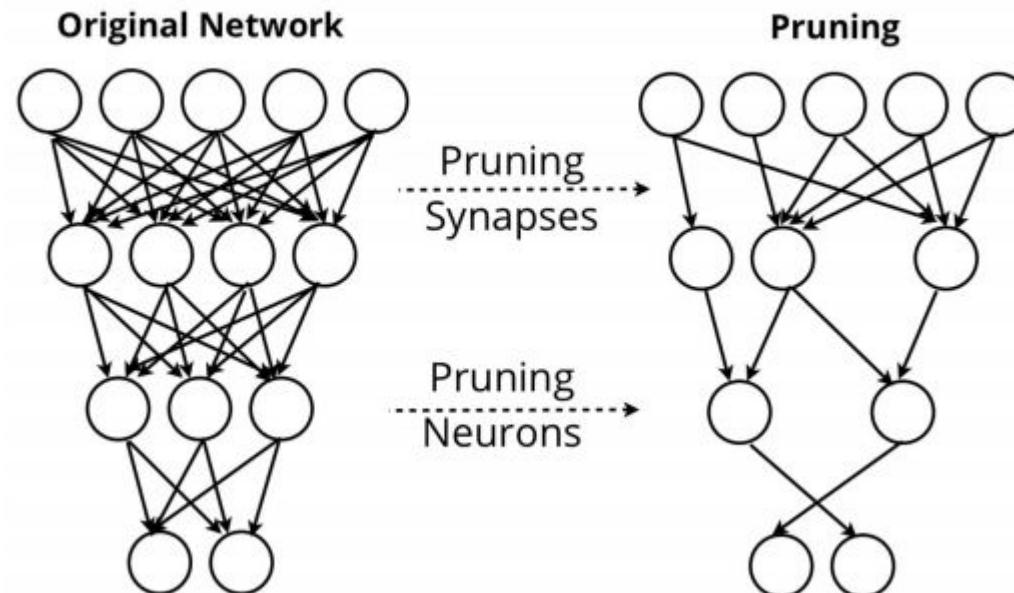


Pruning
Neurons

Structured pruning removes neurons/activations.

Structured pruning can also refer to remove entire heads or model sections [[mixture of experts]]

Unstructured Sparsity of 90% means that by the end of training the model only has 10% of all weights remaining. Apply mask of 0 to remaining weights.



Initial weight matrix

After activations have been removed.

Image [source](#)

Within compression techniques, a variety of different techniques to estimate which weights are least important.

Regularization based techniques

[[L0 regularization [Louizos 2017](#),
L1 regularization [Liu et al. 2017](#),
[Collins et al. 2014](#),
[Wen et al. 2016](#)]]

Addition of penalty term to the loss, coupled with thresholding at end of training.

Rule based techniques

[[Dynamic magnitude pruning [Zhu et al. 2017](#),
Magnitude pruning [Han et al. 2015](#)]]

Remove weights with smallest absolute value, or retain top-k

Within compression techniques, a variety of different techniques to estimate which weights are least important.

Bayesian Techniques

Variational dropout
[[Kingma et al.](#),
[Molchanov et al.](#)]

Assume a distribution over weights. Learn per-weight mean and variance. Prune weights with high variance over weights - high noise to signal ratio.

Random Baseline

[[Liu et al. 2022](#),
[Gale, Elsen and Hooker 2019](#)]]

Remove a subset of the distribution at random.

To give a sense of differences in performances, we benchmark different methods, representative of different groupings.

- Random Baseline
- Iterative Magnitude Pruning [[Zhu et al. 2017](#)]
- L0 regularization [[Louizos 2017](#).]
- Variational dropout [[Molchanov et al.](#)]

Do top performing techniques from the literature perform well at scale?

The State of Sparsity in Deep Neural Networks

Trevor Gale ^{*1†} Erich Elsen ^{*2} Sara Hooker ^{1†}

Abstract

We rigorously evaluate three state-of-the-art techniques for inducing sparsity in deep neural networks on two large-scale learning tasks: Transformer trained on WMT 2014 English-to-German, and ResNet-50 trained on ImageNet. Across thousands of experiments, we demonstrate that complex techniques (Molchanov et al., 2017; Louizos et al., 2017b) shown to yield high compression rates on smaller datasets perform inconsistently, and that simple magnitude pruning approaches achieve comparable or better results. Based on insights from our experiments, we achieve a new state-of-the-art sparsity-accuracy trade-off for ResNet-50 using only magnitude pruning. Additionally, we repeat the experiments performed by Frankle & Carbin (2018) and Liu et al. (2018) at scale and show that unstructured sparse architectures learned through pruning cannot be trained from scratch to the same test set performance as a model trained with joint sparsification and optimization. Together, these results highlight the need for large-scale benchmarks in the field of model compression. We open-source our code, top performing model checkpoints, and results of all hyperparameter configurations to establish rigorous baselines for future work on compression and sparsification.

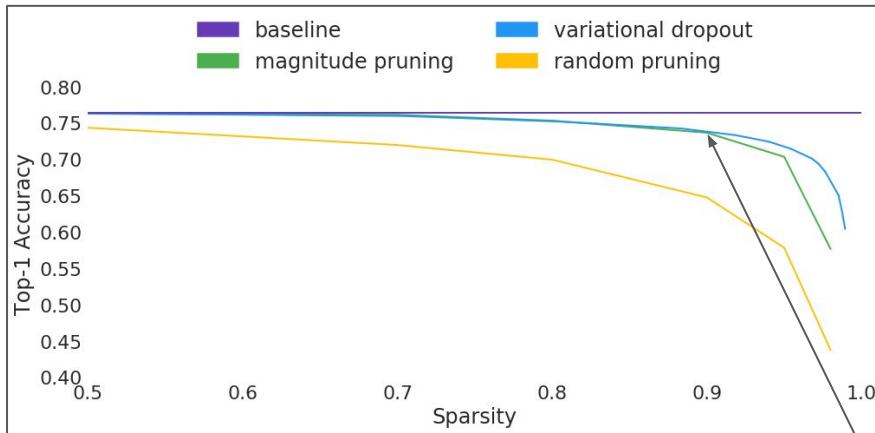
like image classification and machine translation commonly have tens of millions of parameters, and require billions of floating-point operations to make a prediction for a single input sample.

Sparsity has emerged as a leading approach to address these challenges. By sparsity, we refer to the property that a subset of the model parameters have a value of exactly zero². With zero valued weights, any multiplications (which dominate neural network computation) can be skipped, and models can be stored and transmitted compactly using sparse matrix formats. It has been shown empirically that deep neural networks can tolerate high levels of sparsity (Han et al., 2015; Narang et al., 2017; Ullrich et al., 2017), and this property has been leveraged to significantly reduce the cost associated with the deployment of deep neural networks, and to enable the deployment of state-of-the-art models in severely resource constrained environments (Theis et al., 2018; Kalchbrenner et al., 2018; Valin & Skoglund, 2018).

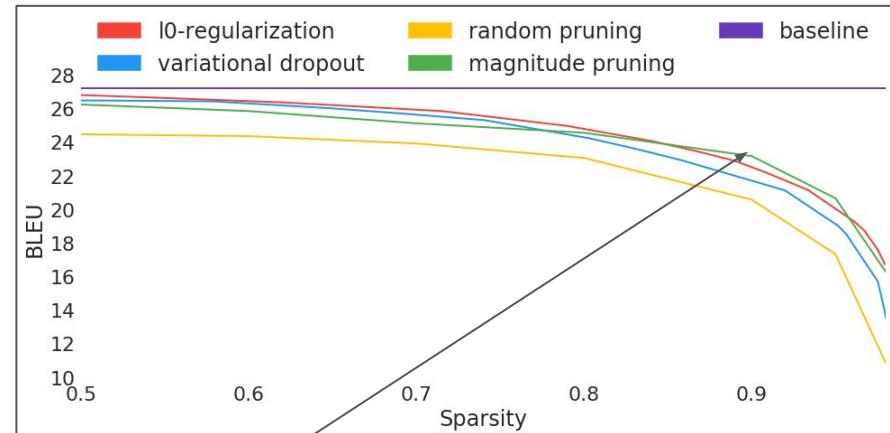
Over the past few years, numerous techniques for inducing sparsity have been proposed and the set of models and datasets used as benchmarks has grown too large to reasonably expect new approaches to explore them all. In addition to the lack of standardization in modeling tasks, the distribution of benchmarks tends to slant heavily towards convolutional architectures and computer vision tasks, and the tasks used to evaluate new techniques are frequently not representative of the scale and complexity of real-world tasks where model compression is most useful. These characteristics make it difficult to come away from the sparsity

Pruning - unsophisticated methods work remarkably well.

Resnet-50 ImageNet



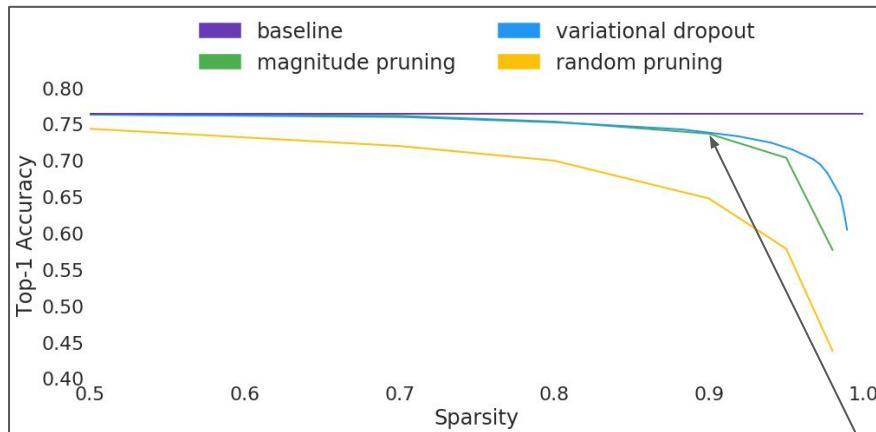
Transformer BLEU



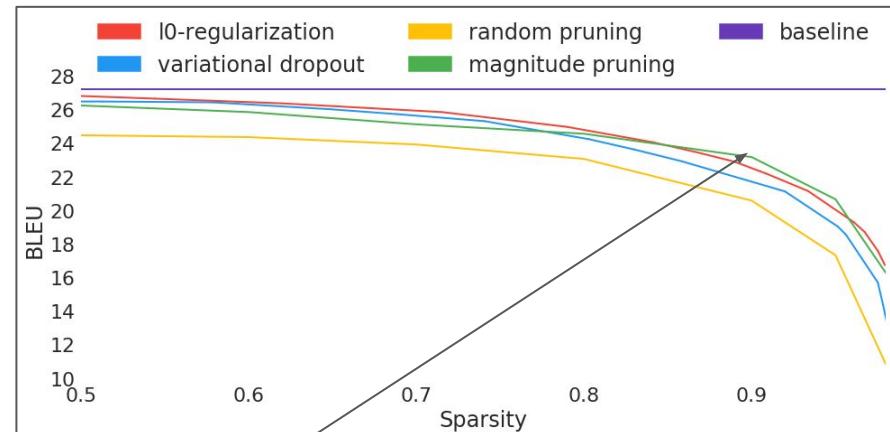
Simply removing weights below a certain threshold works as well as far more sophisticated Bayesian estimators.

Magnitude pruning comparable at most ranges of sparsity, outcompetes at high sparsity levels on transformer.

Resnet-50 ImageNet



Transformer BLEU



Simply removing weights below a certain threshold works as well as far more sophisticated Bayesian estimators.