Re-examining Metrics for Success in Machine Learning, from Fairness and Interpretability to Protein Design

Frances Ding

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-156
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-156.html

August 4, 2024

Re-examining Metrics for Success in Machine Learning, from Fairness and Interpretability to Protein Design

By

Frances Ding

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Assistant Professor Jacob Steinhardt, Co-chair
Associate Professor Moritz Hardt, Co-chair
Professor Yun Song
Professor Jeremy Reiter

Summer 2024

Re-examining Metrics for Success in Machine Learning, from Fairness and Interpretability to Protein Design

Abstract

Re-examining Metrics for Success in Machine Learning, from Fairness and Interpretability to Protein Design

By

Frances Ding

Doctor of Philosophy in Computer Science

University of California, Berkeley

Assistant Professor Jacob Steinhardt, Co-chair

Associate Professor Moritz Hardt, Co-chair

Quantitative metrics, along with datasets to assess them with, are key ingredients that have fueled rapid progress in machine learning (ML) in recent years. These metrics, datasets, and benchmarks define priorities and facilitate efficient discovery of model designs that make progress on those priorities. Ideally, metrics track real world goals, such that improvement on them translates to improvement in related, real tasks. Creating metrics that achieve this external validity is an ever-present challenge in ML. Thus, the science of metrics is an iterative one, as identifying and resolving one issue allows other, more subtle ones, to become apparent.

In this thesis, we describe a series of works that highlight limitations in metrics across different subfields of ML and design new metrics to fill these gaps. We first examine *representation similarity* metrics used in the interpretability subfield to compare neural network representations. We show that current popular metrics often disagree on fundamental observations, making it unclear which one we should believe. We develop practical, statistically grounded tests to evaluate these metrics and find different weaknesses in each. We next examine metrics and benchmarks for *fair classification*. We highlight idiosyncrasies in the popular UCI Adult dataset that limit its external validity, and we contribute a suite of new datasets derived from US Census surveys that extend the existing data ecosystem for research on fair machine learning. Finally, we examine the subfield of *protein modeling* with ML. We develop metrics to quantify a novel type of bias present in popular protein language models–bias towards sequences from certain evolutionary taxa. We additionally introduce a method to mitigate this bias. Across these works in diverse subfields, we demonstrate the challenges and opportunities present in developing metrics that advance technical capabilities in alignment with real world needs.

To my parents, Christine and George.

Contents

List of Figures

List of Tables

Acknowledgments

My academic journey so far, including this PhD, has been a winding and non-linear adventure. Through all those twists and turns, I am grateful to so many mentors, colleagues, friends, and family for their support and guidance.

First I'd like to thank my advisors, Jacob Steinhardt and Moritz Hardt. Jacob started at Berkeley the same year I did, and I feel privileged that I got to watch and learn from him as he built a research group and culture centered on both doing good science and being a good human. Jacob encouraged me to take the time to find problems that I was excited about, and I'm grateful for his support as I jumped from fairness and interpretability to computational biology. As a testament to his versatility as an advisor, he always provided insightful advice, from low-level technical details to high-level vision, across the subfields we tackled. Jacob also led by example in asking for feedback to constantly grow and improve, and I will take those lessons with me for the rest of my career.

Moritz similarly encouraged me to take time to explore different ideas, and when I was worried about early projects fizzling out, he reassured me that good ideas had a way of developing while on the back burner and solidifying once their time arrived. That has proven to be sage advice, as my final PhD projects have incorporated elements from all the subfields I've worked in. Moritz was also always wonderful to brainstorm with–his creativity, ability to connect different fields together, and willingness to challenge the status quo continue to be an inspiration to me.

I'm grateful to the other members of my dissertation committee, Yun Song and Jeremy Reiter. Along with providing helpful feedback on my work, Yun generously allowed me to attend his group meetings, as I pivoted to computational biology in the third and fourth years of my PhD. Jeremy also always made me feel welcome in his lab, and I've learned so much about cilia and GPCR biology from him. Jeremy has an infectious enthusiasm for solving scientific puzzles, and I'm grateful we got a chance to work together on a project using machine learning to predict GPCR localization. That project didn't make it into this dissertation, but it heavily informed the dissertation and will continue to inform my next projects.

I'd also like to thank Ben Recht and Jennifer Listgarten for welcoming me to their group meetings and broadening my research horizons. Ben did an amazing job making Soda 5 a dynamic, friendly environment, and I always appreciated hearing his honest opinions on research directions. Jenn similarly cultivated a wonderful ScienceML group, and I learned a lot from her high expectations for rigor and practical usefulness when applying machine learning to biology.

Next, I'd like to thank my incredible research collaborators. First, thank you to Jean-Stanislas Denain, who helped Jacob and I round out our project on representation similarity metrics, and who I had many thought-provoking conversations with about interpretability and the societal impacts of machine learning. Thank you to John Miller and Ludwig Schmidt, who along with Moritz, built the folktables project; I learned so much about experiment design, maintainable software, and data sleuthing from them. Thank you to Celestine

Mendler-Dünner and Yixin Wang, who taught me about causal inference and performativity through our project on predicting from predictions. Finally, thank you to Mia Konjikusic and Thi Nguyen for doing the experimental work (which was substantially harder than the computational side) for our project with Jeremy on ciliary GPCRs.

I'm also grateful to my team at Google X–Jon Deaton, Ivan Grubisic, Ryan Poplin, Michelle Wynn, Anand Pai, Lance Co Ting Keh, and Hayley Weir for a great internship as I explored using ML to design biological sequences.

Thank you to the many administrators and staff members who kept operations at Berkeley running: Louise Verkin, Ryan Lovett, Shirley Salanio, Jean Nguyen, Angie Abbatecola, and Naomi Yamasaki kept us fed and funded, kept the GPUs on, and made all the work in this dissertation possible.

I'd like to thank the mentors that helped me reach Berkeley and this point. Thank you to Jeffrey Macklis, Vibhu Sahni, and Eva Gillis-Buck for introducing me to research and the amazing complexity of the brain. Thank you to Cynthia Dwork for believing in me when I pivoted to ML, teaching me how to do computer science research, and leading by example in engaging deeply with the societal impacts of computational research. Thank you to Sebastian Tschiatschek for introducing me to interpretability research and how to work with deep neural networks.

I'm very lucky to have wonderful friends and labmates, without whom my PhD journey wouldn't be the same. I'm grateful I got to spend time at Berkeley alongside Ajay Jain, Akosua Busia, Alex Pan, Alex Wei, Alok Tripathy, Anastasios Angelopoulos, Cassidy Laidlaw, Cathy Chen, Chloe Hsu, Clara Wong-Fannjiang, Daniel Rothchild, David Bruns-Smith, Deb Raji, Erik Jones, Evonne Ng, Ezinne Nwankwo, Hanlun Jiang, Hunter Nisonoff, James Bowden, Jiahai Feng, Juanky Perdomo, Junhao Xiong, Karl Krauth, Kayo Yin, Kevin Lin, Lucy Li, Lydia Liu, Mariel Werner, Meena Jagadeesan, Mihaela Curmei, Milind Jagota, Neil Thomas, Nilesh Tripuraneni, Norman Mu, Orr Paradise, Paula Gradu, Reese Pathak, Robbie Netzorg, Rudy Corona, Ruiqi Zhong, Sam Robertson, Sara Fridovich-Keil, Serena Wang, Scott Emmons, Sebastian Prillo, Smitha Milli, Stephan Allenspach, Suzie Petryk, Tijana Zrnic, Tim Brooks, Vickie Ye, Yifan Wu, Yu Sun, and Yuqing Du (plus of course the friends I've already mentioned above).

I'm also grateful to friends who reminded me of life outside of grad school and made my journey much more enjoyable: Alex Wang, Angie Jo, Charlene Hong, Connie Wu, Eesha Khare, Evan Yao, Grace Xiao, Grace Young, Grant Uy, Jacob Brown, Jacob Van Buren, Jade Moon, Jimmy Lin, Kelly Zhang, Kevin Ma, Kimberley Yu, Lily Zhang, Nicholas Laurus-Stone, Nick Gale, Peggy Xu, Rose Burnam, Sarah Yeoh-Wang, Sven Wang, Tez Clark, Wayne Zhao, and many others.

Thank you to Thi, for all the support in the tough times and the celebrations in the good times. You make me a better person. Thank you to my brother Tony and sister-in-law Vivian for always making me laugh.

Finally, thank you to my parents, for everything. I picked up a love of learning from you, and I would not be here without your endless support.

Chapter 1

Introduction

Quantitative metrics, along with datasets to assess them with, are key ingredients that have fueled rapid progress in machine learning (ML) in recent years. These metrics, datasets, and benchmarks define priorities and facilitate efficient discovery of model designs that make progress on those priorities. For example, the ImageNet Large Scale Visual Recognition Challenge [108] provided one of the first demonstrations of deep learning's effectiveness. As ML has become increasingly deployed in application, the ML community has recognized multifaceted priorities and developed new metrics with expanded scope. Commonly assessed metrics now go beyond accuracy on a test set and evaluate many other desiderata, such as robustness to distribution shift, fairness across demographic groups, and interpretability of outputs.

Ultimately, metrics and benchmarks are meant to track real world goals, such that improvement on them translates to improvement in related, real tasks. Creating benchmarks that achieve this *external validity* is an ever-present challenge in the field, with many potential pitfalls [77]. Thus, the science of benchmarking is an iterative one, as identifying and resolving one issue allows other, more subtle ones, to become apparent.

In this thesis, we describe a series of works aimed at interrogating how well-known metrics and benchmarks across different subfields of ML fall short on external validity, and how to design new metrics and benchmarks. By examining fairness and interpretability, we highlight the need for metrics that reflect social and practical considerations. We also explore the emerging field of protein design, demonstrating how novel metrics are essential in ML applications in the natural sciences. Through these works, we hope to pave the way for future research that not only advances technical capabilities but also aligns closely with societal values and needs.

We now describe each section in more detail.

Evaluating Representation Similarity Metrics

In Chapter 2 we examine metrics used to understand and interpret neural network behavior, specifically *dissimilarity measures* that compare different networks' learned representations, such as canonical correlation analysis (CCA), centered kernel alignment (CKA), and other measures. Unfortunately, these widely used measures often disagree on fundamental observations, such as whether deep networks differing only in random initialization learn similar representations. These disagreements raise the question: which, if any, of these dissimilarity measures should we believe?

We provide a framework to ground this question through a concrete test: measures should have *sensitivity* to changes that affect functional behavior, and *specificity* against changes that do not. We quantify this through a variety of functional behaviors including probing accuracy and robustness to distribution shift, and examine changes such as varying random initialization and deleting principal components. We find that current metrics exhibit different weaknesses, note that a classical baseline performs surprisingly well, and highlight settings where all metrics appear to fail, thus providing a challenge set for further improvement.

This work appears as Ding, Denain, and Steinhardt [34].

Assessing Fair Machine Learning with New Datasets

In Chapter 3 we turn to benchmarks that assess whether ML algorithms satisfy fairness considerations across demographic and other characteristics. Although the fairness community has recognized the importance of data, researchers in the area primarily rely on UCI Adult when it comes to tabular data. Derived from a 1994 US Census survey, this dataset has appeared in hundreds of research papers where it served as the basis for the development and comparison of many algorithmic fairness interventions. We reconstruct a superset of the UCI Adult data from available US Census sources and reveal idiosyncrasies of the UCI Adult dataset that limit its external validity. Our primary contribution is a suite of new datasets derived from US Census surveys that extend the existing data ecosystem for research on fair machine learning.

We also create new prediction tasks relating to income, employment, health, transportation, and housing. The data span multiple years and all states of the United States, allowing researchers to study temporal shift and geographic variation. We highlight a broad initial sweep of new empirical insights relating to trade-offs between fairness criteria, performance of algorithmic interventions, and the role of distribution shift based on our new datasets. Our findings inform ongoing debates, challenge some existing narratives, and point to future research directions in fair machine learning.

This work appears as Ding, Hardt, Miller, and Schmidt [35].

Identifying Biases in Protein Language Models

In Chapter 4 we bring insights from benchmarking for interpretability and fairness to the field of protein modeling. Recently, protein language models (PLMs) trained on large protein sequence databases have been used to understand disease and design novel proteins. In design tasks, the likelihood of a protein sequence under a PLM is often used as a proxy for protein fitness, so it is critical to understand what signals likelihoods capture. In this chapter we show that PLM likelihoods unintentionally encode a species bias: likelihoods of protein sequences from certain species are systematically higher, independent of the protein in question. We quantify this bias and show that it arises in large part because of unequal representation of different branches of the evolutionary tree in popular protein sequence databases. We further show that the bias can be detrimental for some protein design applications, such as enhancing thermostability. Finally, we develop post-hoc bias mitigation strategies that reduce these detrimental effects on design. These results highlight the importance of understanding and curating PLM pre-training data to mitigate biases and improve protein design capabilities in under-explored parts of sequence space.

This work appears as Ding and Steinhardt [33].

Chapter 2

Evaluating Representation Similarity Metrics

2.1 Introduction

Understanding neural networks is not only scientifically interesting, but critical for applying deep networks in high-stakes situations. Recent work has highlighted the value of analyzing not just the final outputs of a network, but also its intermediate representations [75, 104]. This has motivated the development of representation similarity measures, which can provide insight into how different training schemes, architectures, and datasets affect networks' learned representations.

A number of similarity measures have been proposed, including centered kernel alignment (CKA) [63], ones based on canonical correlation analysis (CCA) [88, 105], single neuron alignment [75], vector space alignment [8, 28, 114], and others [4, 42, 68, 72, 76, 126]. Unfortunately, these different measures tell different stories. For instance, CKA and projection weighted CCA disagree on which layers of different networks are most similar [63]. This lack of consensus is worrying, as measures are often designed according to different and incompatible intuitive desiderata, such as whether finding a one-to-one assignment, or finding few-to-one mappings, between neurons is more appropriate [75]. As a community, we need well-chosen formal criteria for evaluating metrics to avoid over-reliance on intuition and the pitfalls of too many researcher degrees of freedom [70].

In this paper we view representation dissimilarity measures as implicitly answering a classification question–whether two representations are essentially similar or importantly different. Thus, in analogy to statistical testing, we can evaluate them based on their *sensitivity* to important change and *specificity* (non-responsiveness) against unimportant changes or noise.

As a warm-up, we first initially consider two intuitive criteria: first, that metrics should have specificity against random initialization; and second, that they should be sensitive to deleting important principal components (those that affect probing accuracy). Unfortunately,

popular metrics fail at least one of these two tests. CCA is not specific – random initialization noise overwhelms differences between even far-apart layers in a network (Section 2.3). CKA on the other hand is not sensitive, failing to detect changes in all but the top 10 principal components of a representation (Section 2.3).

We next construct quantitative benchmarks to evaluate a dissimilarity measure's quality. To move beyond our intuitive criteria, we need a ground truth. For this we turn to the functional behavior of the representations we are comparing, measured through probing accuracy (an indicator of syntactic information) [14, 100, 119] and out-of-distribution performance of the model they belong to [30, 83, 89]. We then score dissimilarity measures based on their rank correlation with these measured functional differences. Overall our benchmarks contain 30,480 examples and vary representations across several axes including random seed, layer depth, and low-rank approximation (Section 2.4)[1].

Our benchmarks confirm our two intuitive observations: on subtasks that consider layer depth and principal component deletion, we measure the rank correlation with probing accuracy and find CCA and CKA lacking as the previous warm-up experiments suggested. Meanwhile, the Orthogonal Procrustes distance, a classical but often overlooked[2] dissimilarity measure, balances gracefully between CKA and CCA and consistently performs well. This underscores the need for systematic evaluation, otherwise we may fall to recency bias that undervalues classical baselines.

Other subtasks measure correlation with OOD accuracy, motivated by the observation that random initialization sometimes has large effects on OOD performance [83]. We find that dissimilarity measures can sometimes predict OOD performance using only the in-distribution representations, but we also identify a challenge set on which none of the measures do statistically better than chance. We hope this challenge set will help measure and spur progress in the future.

2.2 Problem Setup: Metrics and Models

Our goal is to quantify the similarity between two different groups of neurons (usually layers). We do this by comparing how their activations behave on the same dataset. Thus for a layer with p_1 neurons, we define $A \in \mathbb{R}^{p_1 \times n}$, the matrix of activations of the p_1 neurons on n data points, to be that layer's raw representation of the data. Similarly, let $B \in \mathbb{R}^{p_2 \times n}$ be a matrix of the activations of p_2 neurons on the same n data points. We center and normalize these representations before computing dissimilarity, per standard practice. Specifically, for a raw representation A we first subtract the mean value from each column, then divide by the Frobenius norm, to produce the normalized representation A^*, used in all our dissimilarity computations. In this work we study dissimilarity measures $d(A^*, B^*)$

[1] Code to replicate our results can be found at `https://github.com/js-d/sim_metric`.

[2] For instance, Raghu et al. [105] and Morcos et al. [88] do not mention it, and Kornblith et al. [63] relegates it to the appendix; although Smith et al. [114] does use it to analyze word embeddings and prefers it to CCA.

that allow for quantitative comparisons of representations both within and across different networks. We colloquially refer to values of $d(A^*, B^*)$ as distances, although they do not necessarily satisfy the triangle inequality required of a proper metric.

We study five dissimilarity measures: centered kernel alignment (CKA), three measures derived from canonical correlation analysis (CCA), and a measure derived from the orthogonal Procrustes problem.

Centered kernel alignment (CKA) uses an inner product to quantify similarity between two representations. It is based on the idea that one can first choose a kernel, compute the $n \times n$ kernel matrix for each representation, and then measure similarity as the alignment between these two kernel matrices. The measure of similarity thus depends on one's choice of kernel; in this work we consider **Linear CKA**:

$$d_{\text{Linear CKA}}(A, B) = 1 - \frac{\|AB^\top\|_F^2}{\|AA^\top\|_F \|BB^\top\|_F} \tag{2.1}$$

as proposed in Kornblith et al. [63]. Other choices of kernel are also valid; we focus on Linear CKA here since Kornblith et al. [63] report similar results from using either a linear or RBF kernel.

Canonical correlation analysis (CCA) finds orthogonal bases (w_A^i, w_B^i) for two matrices such that after projection onto w_A^i, w_B^i, the projected matrices have maximally correlated rows. For $1 \leq i \leq p_1$, the i^{th} canonical correlation coefficient ρ_i is computed as follows:

$$\rho_i = \max_{w_A^i, w_B^i} \frac{\langle {w_A^i}^\top A, {w_B^i}^\top B \rangle}{\|{w_A^i}^\top A\| \cdot \|{w_B^i}^\top B\|} \tag{2.2}$$

$$s.t. \quad \langle {w_A^i}^\top A, {w_A^j}^\top A \rangle = 0, \ \forall j < i, \quad \langle {w_B^i}^\top B, {w_B^j}^\top B \rangle = 0, \ \forall j < i \tag{2.3}$$

To transform the vector of correlation coefficients into a scalar measure, two options considered previously [63] are the **mean correlation coefficient, $\bar{\rho}_{\text{CCA}}$**, and the **mean squared correlation coefficient, R^2_{CCA}**, defined as follows:

$$d_{\bar{\rho}_{\text{CCA}}}(A, B) = 1 - \frac{1}{p_1} \sum_i \rho_i, \qquad d_{R^2_{\text{CCA}}}(A, B) = 1 - \frac{1}{p_1} \sum_i \rho_i^2 \tag{2.4}$$

To improve the robustness of CCA, Morcos et al. [88] propose **projection-weighted CCA (PWCCA)** as another scalar summary of CCA:

$$d_{\text{PWCCA}}(A, B) = 1 - \frac{\sum_i \alpha_i \rho_i}{\sum_i \alpha_i}, \quad \alpha_i = \sum_j |\langle h_i, a_j \rangle| \tag{2.5}$$

where a_j is the j^{th} row of A, and $h_i = {w_A^i}^\top A$ is the projection of A onto the i^{th} canonical direction. We find that PWCCA performs far better than $\bar{\rho}_{\text{CCA}}$ and R^2_{CCA}, so we focus on PWCCA in the main text, but include results on the other two measures in the appendix.

The **orthogonal Procrustes** problem consists of finding the left-rotation of A that is closest to B in Frobenius norm, i.e. solving the optimization problem:

$$\min_R \|B - RA\|_{\mathrm{F}}^2, \text{ subject to } R^\top R = I. \tag{2.6}$$

The minimum is the squared **orthogonal Procrustes distance** between A and B, and is equal to

$$d_{\mathrm{Proc}}(A, B) = \|A\|_F^2 + \|B\|_F^2 - 2\|A^\top B\|_*, \tag{2.7}$$

where $\|\cdot\|_*$ is the nuclear norm [110]. Unlike the other metrics, the orthogonal Procrustes distance is not normalized between 0 and 1, although for normalized A^*, B^* it lies in $[0, 2]$.

Models we study

In this work we study representations of both text and image inputs. For text, we investigate representations computed by Transformer architectures in the BERT model family [32] on sentences from the Multigenre Natural Language Inference (MNLI) dataset [127]. We study BERT models of two sizes: BERT base, with 12 hidden layers of 768 neurons, and BERT medium, with 8 hidden layers of 512 neurons. We use the same architectures as in the open source BERT release[3], but to generate diversity we study 3 variations of these models:

1. 10 BERT base models pretrained with different random seeds but not finetuned for particular tasks, released by Zhong et al. [132][4].
2. 10 BERT medium models initialized from pretrained models released by Zhong et al. [132], that we further finetuned on MNLI with 10 different finetuning seeds (100 models total).
3. 100 BERT base models that were initialized from the pretrained BERT model in [32] and finetuned on MNLI with different seeds, released by McCoy et al. [83][5].

For images, we investigate representations computed by ResNets [52] on CIFAR-10 test set images [65]. We train 100 ResNet-14 models[6] from random initialization with different seeds on the CIFAR-10 training set and collect representations after each convolutional layer.

Further training details, as well as checks that our training protocols result in models with comparable performance to the original model releases, can be found in Appendix 2.6.

2.3 Warm-up: Intuitive Tests for Sensitivity and Specificity

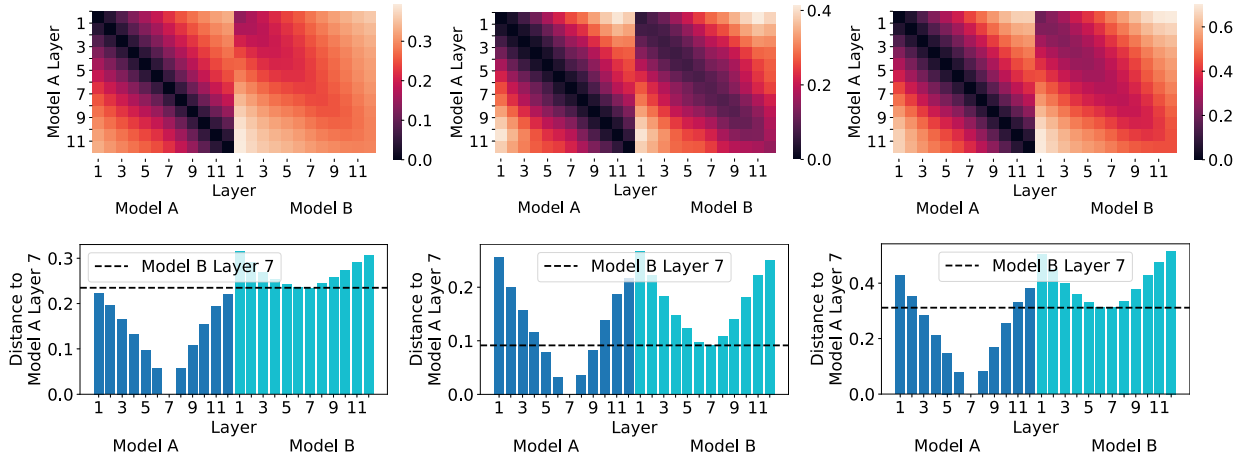
When designing dissimilarity measures, researchers usually consider invariants that these measures should not be sensitive to [63]; for example, symmetries in neural networks imply

[3]available at `https://github.com/google-research/bert`

[4]available at `https://github.com/ruiqi-zhong/acl2021-instance-level`

[5]available at `https://github.com/tommccoy1/hans/tree/master/berts_of_a_feather`

[6]from `https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py`



(a) PWCCA distances (b) CKA distances (c) Orthogonal Procrustes distances

Figure 2.1: **PWCCA fails the intuitive specificity test.** Top: PWCCA, CKA, and Orthogonal Procrustes pairwise distances between each layer of two differently initialized networks (Model A and B). Bottom: We zoom in to analyze the 7^{th} layer of Model A, plotting this layer's distance to every other layer in both networks; the dashed line indicates the distance to the corresponding 7^{th} layer in Model B. For PWCCA, none of the distances in model A exceed this line, indicating that random initialization affects this distance more than large changes in layer depth.

that permuting the neurons in a fully connected layer does not change the representations learned. We take this one step further and frame dissimilarity measures as answering whether representations are essentially the same, or importantly different. We can then evaluate measures based on whether they respond to important changes (sensitivity) while ignoring changes that don't matter (specificity).

Assessing sensitivity and specificity requires a ground truth–which representations are truly different? To answer this, we begin with the following two intuitions[7]: 1) neural network representations trained on the same data but from different random initializations are similar, and 2) representations lose crucial information as principal components are deleted. These motivate the following intuitive tests of specificity and sensitivity: we expect a dissimilarity measure to: 1) assign a small distance between architecturally identical neural networks that only differ in initialization seed, and 2) assign a large distance between a representation A and the representation $\hat{A}$ after deleting important principal components (enough to affect accuracy). We will see that PWCCA fails the first test (specificity), while CKA fails the second (sensitivity).

[7]Note we will see later that these intuitions need refinement.

Specificity against changes to random seed

Neural networks with the same architecture trained from different random initializations show many similarities, such as highly correlated predictions on in-distribution data points [83]. Thus it seems natural to expect a good similarity measure to assign small distances between architecturally corresponding layers of networks that are identical except for initialization seed.

To check this property, we take two BERT base models pre-trained with different random seeds and, for every layer in the first model, compute its dissimilarity to every layer in both the first and second model. We do this for 5 separate pairs of models and average the results. To pass the intuitive specificity test, a dissimilarity measure should assign relatively small distances between a layer in the first network and its corresponding layer in the second network.

Figure 2.1 displays the average pair-wise PWCCA, CKA, and Orthogonal Procrustes distances between layers of two networks differing only in random seed. According to PWCCA, these networks' representations are quite dissimilar; for instance, the two layer 7 representations are further apart than they are from any other layer in the same network. PWCCA is thus not specific against random initialization, as it can outweigh even large changes in layer depth.

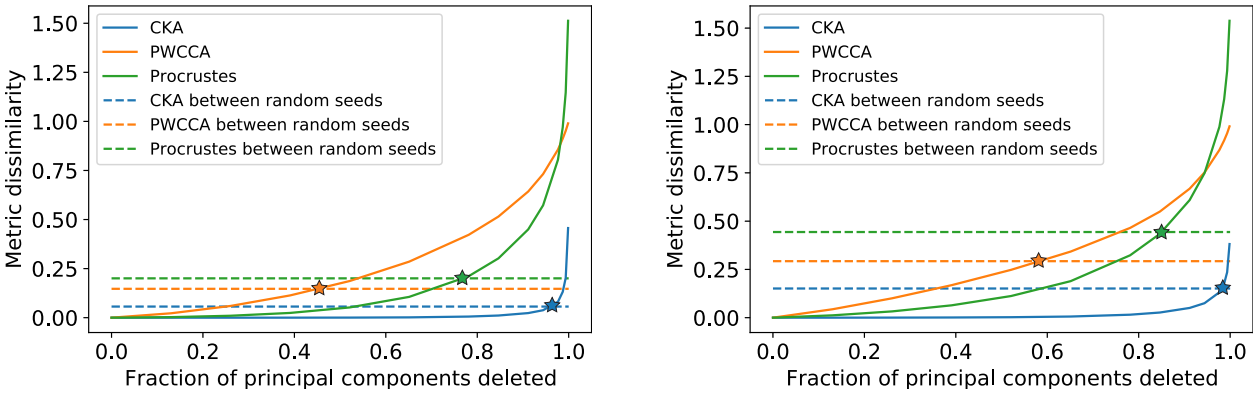
In contrast, CKA can separate layer 7 in a different network from layers 4 or 10 in the same network, showing better specificity to random initialization. Orthogonal Procrustes exhibits smaller but non-trivial specificity, distinguishing layers once they are 4-5 layers apart.

Sensitivity to removing principal components

Dissimilarity measures should also be sensitive to deleting important principal components of a representation.[8] To quantify which components are important, we fix a layer of a pre-trained BERT base model and measure how probing accuracy degrades as principal components are deleted (starting from the smallest component), since probing accuracy is a common measure of the information captured in a representation [14]. We probe linear classification performance on the Stanford Sentiment Tree Bank task (SST-2) [115], following the experimental protocol in Tamkin et al. [117]. Figure 2.3b shows how probing accuracy degrades with component deletion. Ideally, dissimilarity measures should be large by the time probing accuracy has decreased substantially.

To assess whether a dissimilarity measure is large, we need a baseline to compare to. For each measure, we define a dissimilarity score to be above the *detectable* threshold if it is larger than the dissimilarity score between networks with different random initialization. Figure 2.2 plots the dissimilarity induced by deleting principal components, as well as this baseline.

[8] For a representation A, we define $\hat{A}_{-k}$, the result of deleting the k smallest principal components from A, as follows: we compute the singular value decomposition $U\Sigma V^T = A$, construct $U_{-k} \in \mathbb{R}^{p \times p-k}$ by dropping the lowest k singular vectors of U, and finally take $\hat{A}_{-k} = U_{-k}^T A$.



(a) First layer of BERT

(b) Last layer of BERT

Figure 2.2: **CKA fails to be sensitive to all but the largest principal components.** We compute dissimilarities between a layer's representation and low-rank approximations to that representation obtained by deleting principal components, starting from the smallest (solid lines). We also compute the average distance between networks trained with different random seeds as a baseline (dotted line), and mark the intersection point with a star. The starred points indicate that CKA requires almost all the components to be deleted before CKA distance exceeds the baseline.

For the last layer of BERT, CKA requires 97% of a representation's principal components to be deleted for the dissimilarity to be detectable; after deleting these components, probing accuracy shown in Figure 2.3b drops significantly from 80% to 63% (chance is 50%). CKA thus fails to detect large accuracy drops and so fails our intuitive sensitivity test.

Other metrics perform better: Orthogonal Procrustes's detection threshold is ∼85% of the principal components, corresponding to an accuracy drop 80% to 70%. PWCCA's threshold is ∼55% of principal components, corresponding to an accuracy drop from 80% to 75%.

PWCCA's failure of specificity and CKA's failure of sensitivity on these intuitive tests are worrying. However, before declaring definitive failure, in the next section, we turn to making our assessments more rigorous.

2.4 Rigorously Evaluating Dissimilarity Metrics

In the previous section, we saw that CKA and PWCCA each failed intuitive tests, based on sensitivity to principal components and specificity to random initialization. However, these were based primarily on intuitive, qualitative desiderata. Is there some way for us to make these tests more rigorous and quantitative?

First consider the intuitive layer specificity test (Section 2.3), which revealed that random initialization affects PWCCA more than large changes in layer depth. To justify why this is undesirable, we can turn to probing accuracy, which is strongly affected by layer depth, and

only weakly affected by random seed (Figure 2.3a). This suggests a path forward: we can ground the layer test in the concrete differences in functionality captured by the probe.

More generally, we want metrics to be sensitive to changes that affect functionality, while ignoring those that don't. This motivates the following general procedure, given a distance metric d and a functionality f (which assigns a real number to a given representation):

1. Collect a set S of representations that differ along one or more axes of interest (e.g. layer depth, random seed).
2. Choose a reference representation $A \in S$. When f is an accuracy metric, it is reasonable to choose $A = \arg\max_{A \in S} f(A)$.[9]
3. For every representation $B \in S$:
   - Compute $|f(A) - f(B)|$
   - Compute $d(A, B)$
4. Report the rank correlation between $|f(A) - f(B)|$ and $d(A, B)$ (measured by Kendall's τ or Spearman ρ).

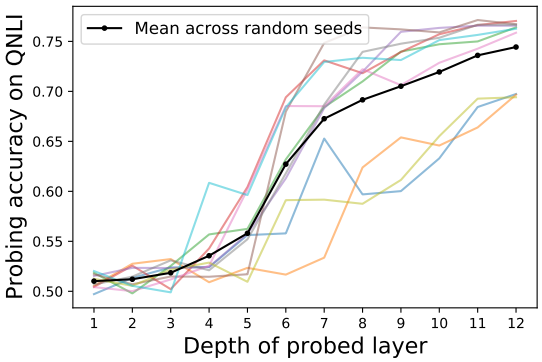
The above procedure provides a *quantitative* measure of how well the distance metric d responds to the functionality f. For instance, in the layer specificity test, since depth affects probing accuracy strongly while random seed affects it only weakly, a dissimilarity measure with high rank correlation will be strongly responsive to layer depth and weakly responsive to seed; thus rank correlation quantitatively formalizes the test from Section 2.3.

Correlation metrics also capture properties that our intuition might miss. For instance, Figure 2.3a shows that some variation in random seed actually does affect accuracy, and our procedure rewards metrics that pick up on this, while the intuitive sensitivity test would penalize them.

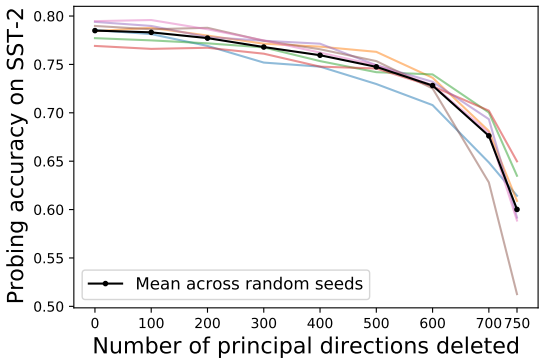
Our procedure requires choosing a collection of models S; the crucial feature of S is that it contains models with diverse behavior according to f. Different sets S, combined with a functional difference f, can be thought of as miniature "benchmarks" that surface complementary perspectives on dissimilarity measures' responsiveness to that functional difference. In the rest of this section, we instantiate this quantitative benchmark for several choices of f and S, starting with the layer and principal component tests from Section 2.3 and continuing on to several tests of OOD performance.

The overall results are summarized in Table 2.1. Note that for any single benchmark, we expect the correlation coefficients to be significantly lower than 1, since the metric D must capture all important axes of variation while f measures only one type of functionality. A

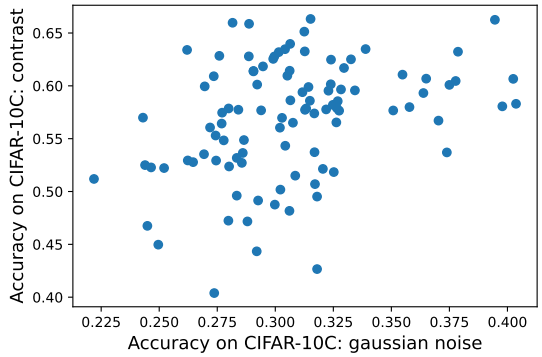
[9]Choosing the highest accuracy model as the reference makes it more likely that as accuracy changes, models are on average becoming more dissimilar. A low accuracy model may be on the "periphery" of model space, where it is dissimilar to models with high accuracy, but potentially even more dissimilar to other low accuracy models that make different mistakes.



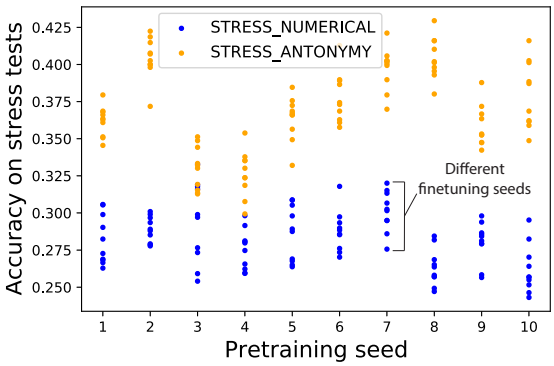
(a) Probing variation across BERT base layers



(b) Variation across BERT base PC deletions



(c) Variation across ResNet-14 training seeds



(d) Variation across BERT med training seeds

Figure 2.3: **Our perturbations induce substantial variation on probing tasks and stress tests:** (2.3a): Changing the depth of the examined BERT base layer strongly affects probing accuracy on QNLI. The trend for each randomly initialized model is displayed semi-transparently, and the solid black line is the mean trend. (2.3b): Truncating principal components from pretrained BERT base significantly degrades probing accuracy on SST-2 (BERT layer 12 shown here). (2.3c): Training ResNet-14 on CIFAR-10 with different seeds leads to variation in accuracies on CIFAR-10C corruptions (here Gaussian noise and contrast). (2.3d): Pretraining and finetuning BERT medium with 10 different pretraining seeds and 10 different finetuning seeds per pretrained model leads to variation in accuracies on the Antonymy (yellow scatter points) and Numerical (blue scatter points) stress tests [89].

good metric is one that has consistently high correlation across many different functional measures.

Benchmark 1: Layer depth. We turn the layer test into a benchmark for both text and images. For the text setting, we construct a set S of 120 representations by pretraining 10 BERT base models with different initialization seeds and including each of the 12 BERT layers as a representation. We separately consider two functionalities f: probing accuracy on QNLI [124] and SST-2 [115]. To compute the rank correlation, we take the reference representation A to be the representation with highest probing accuracy. We compute the Kendall's τ and Spearman's ρ rank correlations between the dissimilarities and the probing accuracy differences and report the results in Table 2.1.

For the image setting, we similarly construct a set S of 70 representations by training 5 ResNet-14 models with different initialization seeds and including each of the 14 layers' representations. We also consider two functionalities f for these vision models: probing accuracy on CIFAR-100 [65] and on SVHN [90], and compute rank correlations in the same way.

We find that PWCCA has lower rank correlations compared to CKA and Procrustes for both language probing tasks. This corroborates the intuitive specificity test (Section 2.3), suggesting that PWCCA registers too large of a dissimilarity across random initializations. For the vision tasks, CKA and Procrustes achieve similar rank correlations, while PWCCA cannot be computed because $n < d$.

Benchmark 2: Principal component (PC) deletion. We next quantify the PC deletion test from Section 2.3, by constructing a set S of representations that vary in both random initialization and fraction of principal components deleted. We pretrain 10 BERT base models with different initializations, and for each pretrained model we obtain 14 different representations by deleting that representation's k smallest principal components, with $k \in \{0, 100, 200, 300, 400, 500, 600, 650, 700, 725, 750, 758, 763, 767\}$. Thus S has $10 \times 14 = 140$ elements. The representations themselves are the layer-ℓ activations, for $\ell \in \{8, 9, \ldots, 12\}$,[10] so there are 5 different choices of S. We use SST-2 probing accuracy as the functionality of interest f, and select the reference representation A as the element in S with highest accuracy. Rank correlation results are consistent across the 5 choices of S (Appendix 2.6), so we report the average as a summary statistic in Table 2.1.

We find that PWCCA has the highest rank correlation between dissimilarity and probing accuracy, followed by Procrustes, and distantly followed by CKA. This corroborates the intuitive observations from Section 2.3 that CKA is not sensitive to principal component deletion.

[10] Earlier layers have near-chance accuracy on probing tasks, so we ignore them.

Table 2.1: **Summary of rank correlation results.** For Benchmarks #1-3 in both language and vision, all dissimilarity measures successfully achieve significant positive rank correlation with the functionality of interest–both CKA and PWCCA dominate certain benchmarks and fall behind on others, while Procrustes is more consistent and often close to the leader. Benchmark #4 is more challenging, and no dissimilarity measure achieves a high correlation. The vision experiments do not have results for PWCCA because $n < d$.

#	Perturbation	Subtask Size	Functionality	Procrustes		CKA		PWCCA	
				ρ	τ	ρ	τ	ρ	τ
			Modality: Language						
1	Pretraining seed, layer depth	120	Probe: QNLI	0.862	0.670	**0.876**	**0.685**	0.763	0.564
		120	Probe: SST-2	0.890	0.707	**0.905**	**0.732**	0.829	0.637
2	Pretraining seed, PC deletion	140×5	Probe: SST-2	0.860	0.677	0.751	0.564	**0.870**	**0.690**
3	Finetuning seed	100×12	OOD: HANS Lexical non-entailed	0.551	0.398	0.462	0.329	**0.568**	**0.412**
4	Pretraining and finetuning seeds	100×8	OOD: Antonymy stress test	**0.243**	**0.178**	0.227	0.160	0.204	0.152
		100×8	OOD: Numerical stress test	0.071	0.049	**0.122**	**0.084**	0.031	0.023
	Total (language)	3740	Average	**0.580**	**0.447**	0.557	0.426	0.544	0.413
			Modality: Vision						
1	Training seed, layer depth	70	Probe: CIFAR-100	0.485	**0.376**	**0.507**	0.359	-	-
		70	Probe: SVHN	0.363	**0.272**	**0.372**	0.255	-	-
4	Training seed	1900×14	OOD: CIFAR-10C	**0.060**	**0.057**	0.041	0.038	-	-
	Total (vision)	26740	Average	0.303	**0.235**	**0.307**	0.217	-	-

Investigating variation in OOD performance across random seeds

So far our benchmarks have been based on probing accuracy, which only measures in-distribution behavior (the train and test set of the probe are typically i.i.d.). In addition, the BERT models were always pretrained on language modeling but not finetuned for classification. To add diversity to our benchmarks, we next consider the out-of-distribution performance of language and vision models trained for classification tasks.

Benchmark 3: Changing fine-tuning seeds. McCoy et al. [83] show that a single pretrained BERT base model finetuned on MNLI with different random initializations will produce models with similar in-distribution performance, but widely variable performance on out-of-distribution data. We thus create a benchmark S out of McCoy et al.'s 100 released fine-tuned models, using OOD accuracy on the "Lexical Heuristic (Non-entailment)" subset of the HANS dataset [82] as our functionality f. This functionality is associated with the entire model, rather than an individual layer (in contrast to the probing functionality), but we consider one layer at a time to measure whether dissimilarities between representations at that layer correlate with f. This allows us to also localize whether certain layers are more predictive of f.

We construct 12 different S (one for each of the 12 layers of BERT base), taking the reference representation A to be that of the highest accuracy model according to f. As before, we report each dissimilarity measure's rank correlation with f in Table 2.1, averaged over the 12 runs.

All three dissimilarity measures correlate with OOD accuracy, with Orthogonal Procrustes and PWCCA being more correlated than CKA. Since the representations in our benchmarks were computed on in-distribution MNLI data, this has the interesting implication that dissimilarity measures can detect OOD differences without access to OOD data. It also implies that random initialization leads to meaningful functional differences that are picked up by these measures, especially Procrustes and PWCCA. Contrast this with our intuitive specificity test in Section 2.3, where all sensitivity to random initialization was seen as a shortcoming. Our more quantitative benchmark here suggests that some of that sensitivity tracks true functionality.

To check that the differences in rank correlation for Procrustes, PWCCA, and CKA are statistically significant, we compute bootstrap estimates of their 95% confidence intervals. With 2000 bootstrapped samples, we find statistically significant differences between all pairs of measures for most choices of layer depth S, so we conclude PWCCA $>$ Orthogonal Procrustes $>$ CKA (the full results are in Appendix 2.6). We do not apply this procedure for the previous two benchmarks, because the different models have correlated randomness and so any p-value based on independence assumptions would be invalid.

Benchmark 4: Challenge sets: Changing pretraining and fine-tuning seeds. We also construct benchmarks using models trained from scratch with different random seeds (for language, this is pretraining and fine-tuning, and for vision, this is standard training). For language, we construct benchmarks from a collection of 100 BERT medium models, trained with all combinations of 10 pretraining and 10 fine-tuning seeds. The models are fine-tuned on MNLI, and we consider two different functionalities of interest f: accuracy on the OOD Antonymy stress test and on the OOD Numerical stress test [89], which both show significant variation in accuracy across models (see Figure 2.3d). We obtain 8 different sets S (one for each of the 8 layer depths in BERT medium), again taking A to be the representation of the highest-accuracy model according to f. Rank correlations for each dissimilarity measure are

averaged over the 8 runs and reported in Table 2.1.

For vision, we construct benchmarks from a collection of 100 ResNet-14 models, trained with different random seeds on CIFAR-10. We consider 19 different functionalities of interest—the 19 types of corruptions in the CIFAR-10C dataset [53], which show significant variation in accuracy across models (see Figure 2.3c). We obtain 14 different sets S (one for each of the 14 layers), taking A to be the representation of the highest-accuracy model according to f. Rank correlations for each dissimilarity measure are averaged over the 14 runs and over the 19 corruption types and reported in Table 2.1. Results for each of the 19 corruptions individually can be found in Appendix 2.6..

None of the dissimilarity measures show a large rank correlation for either the language or vision tasks, and for the Numerical stress test, at most layers, the associated p-values (assuming independence) are non-significant at the 0.05 level (see Appendix 2.6). [11] Thus we conclude that all measures fail to be sensitive to OOD accuracy in these settings. One reason for this could be that there is less variation in the OOD accuracies compared to the previous experiment with the HANS dataset (there accuracies varied from 0 to nearly 60%). Another reason could be that it is harder to correctly account for both pretraining and fine-tuning variation at the same time. Either way, we hope that future dissimilarity measures can improve upon these results, and we present this benchmark as a challenge task to motivate progress.

2.5 Discussion

In this work we proposed a quantitative measure for evaluating similarity metrics, based on the rank correlation with functional behavior. Using this, we generated tasks motivated by sensitivity to deleting important directions, specificity to random initialization, and sensitivity to out-of-distribution performance. Popular existing metrics such as CKA and CCA often performed poorly on these tasks, sometimes in striking ways. Meanwhile, the classical Orthogonal Procrustes transform attained consistently good performance.

Given the success of Orthogonal Procrustes, it is worth reflecting on how it differs from the other metrics and why it might perform well. To do so, we consider a simplified case where A and B have the same singular vectors but different singular values. Thus without loss of generality $A = \Lambda_1$ and $B = \Lambda_2$, where the Λ_i are both diagonal. In this case, the Orthogonal Procrustes distance reduces to $\|\Lambda_1 - \Lambda_2\|_F^2$, or the sum of the squared distances between the singular values. We will see that both CCA and CKA reduce to less reasonable formulae in this case.

Orthogonal Procrustes vs. CCA. All three metrics derived from CCA assign *zero* distance even when the (non-zero) singular values are arbitrarily different. This is because CCA

[11] See Appendix 2.6 for p-values as produced by sci-kit learn. Strictly speaking, the p-values are invalid because they assume independence, but the pretraining seed induces correlations. However, correctly accounting for these would tend to make the p-values larger, thus preserving our conclusion of non-significance .

correlation coefficients are invariant to all invertible linear transformations. This invariance property may help explain why CCA metrics generally find layers within the same network to be much more similar than networks trained with different randomness. Random initialization introduces noise, particularly in unimportant principal components, while representations within the same network more easily preserve these components, and CCA may place too much weight on their associated correlation coefficients.

Orthogonal Procrustes vs. CKA. In contrast to the squared distance of Orthogonal Procrustes, CKA actually reduces to a quartic function based on the dot products between the *squared* entries of Λ_1 and Λ_2. As a consequence, CKA is dominated by representations' largest singular values, leaving it insensitive to meaningful differences in smaller singular values as illustrated in Figure 2.2. This lack of sensitivity to moderate-sized differences may help explain why CKA fails to track out-of-distribution error effectively.

In addition to helping understand similarity measures, our benchmarks pinpoint directions for improvement. No method was sensitive to accuracy on the Numerical stress test in our challenge set, possibly due to a lower signal-to-noise ratio. Since Orthogonal Procrustes performed well on most of our tasks, it could be a promising foundation for a new measure, and recent work shows how to regularize Orthogonal Procrustes to handle high noise [102]. Perhaps similar techniques could be adapted here.

An alternative to our benchmarking approach is to directly define two representations' dissimilarity as their difference in a functional behavior of interest. Feng et al. [42] take this approach, defining dissimilarity as difference in accuracy on a handful of probing tasks. One drawback of this approach is that a small set of probes may not capture all the differences in representations, so it is useful to base dissimilarity measures on representations' intrinsic properties. Intrinsically defined dissimilarities also have the potential to highlight new functional behaviors, as we found that representations with similar in-distribution probing accuracy often have highly variable OOD accuracy.

A limitation of our work is that we only consider a handful of model variations and functional behaviors, and restricting our attention to these settings could overlook other important considerations. To address this, we envision a paradigm in which a rich tapestry of benchmarks are used to ground and validate neural network interpretations. Other axes of variation in models could include training on more or fewer examples, training on shuffled labels vs. real labels, training from specifically chosen initializations [45], and using different architectures. Other functional behaviors to examine could include modularity and meta-learning capabilities. Benchmarks could also be applied to other interpretability tools beyond dissimilarity. For example, sensitivity to deleting principal components could provide an additional sanity check for saliency maps and other visualization tools [2].

More broadly, many interpretability tools are designed as *audits* of models, although it is often unclear what characteristics of the models are consistently audited. We position this work as a *counter-audit*, where by collecting models that differ in functional behavior, we can assess whether the interpretability tools CKA, PWCCA, etc., accurately reflect the behavioral differences. Many other types of counter-audits may be designed to assess other interpretability tools. For example, models that have backdoors built into them to

misclassify certain inputs provide counter-audits for interpretability tools that explain model predictions–these explanations should reflect any backdoors present [26, 67, 74, 125]. We are hopeful that more comprehensive checks on interpretability tools will provide deeper understanding of neural networks, and more reliable models.

2.6 Supplementary Materials

Training details

BERT finetuning details

We fine-tuned models from Zhong et al. [132] and the original BERT models from Devlin et al. [32] on three tasks – Quora Question Pairs (QQP)[12], Multi-Genre Natural Language Inference (MNLI; Williams et al. [127]), and the Stanford Sentiment Treebank (SST-2; Socher et al. [115]), and show each model's accuracy on these tasks in Table 2.2. Our models generally have comparable accuracy.

As in Turc et al. [122], we finetune for 4 epochs for each dataset. For each task and model size, we tune hyperparameters in the following way: we first randomly split our new training set into 80% and 20%; then we finetune on the 80% split with all 9 combination of batch size [16, 32, 64] and learning rate [1e-4, 5e-5, 3e-5], and choose the combination that leads to the best average accuracy on the remaining 20%. Finetuning these models for all three tasks requires around 500 hours.

Table 2.2: Comparing accuracy of our pretrained model (superscript ^{ours}) to the original release by Devlin et al. [32] and Turc et al. [122] (superscript ^{orig}) on a variety of fine-tuned tasks.

	QQP	MNLI	SST-2
BERT medium $^{\text{orig}}$	89.8%	79.6%	94.2%
BERT medium $^{\text{ours}}$	89.5%	78.9%	94.2%
BERT base $^{\text{orig}}$	90.8%	83.8%	95.0%
BERT base $^{\text{ours}}$	90.6%	81.2%	94.6%

ResNet training details

We trained ResNet-14 models on CIFAR-10 training data with the following hyperparameters:

- learning rate: 0.1

[12]https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

- epochs: 100
- learning rate decay: 0.1 at epoch 50 and epoch 75
- batch size: 128

The 100 models we trained have an average accuracy on the CIFAR-10 test set of 90.2%, with standard deviation 0.2%. Training these models requires around 20 hours.

Licenses

The source code for BERT models available at `https://github.com/google-research/bert` is licensed under the Apache License 2.0.

The model weights for the 100 BERT base models provided by McCoy et al. [83] are licensed under the Creative Commons Attribution 4.0 International license, and their source code is licensed under the MIT license (`https://github.com/tommccoy1/hans/blob/master/LICENSE.md`).

Layer-wise results

Some of the results presented in Table 2.1 were averaged over multiple layers, since rankings between dissimilarity measures were consistent across different layers. Rank correlation scores are higher across all measures for certain layers, however, so we include layer-by-layer results here for completeness. We also include scores for $\bar{\rho}_{\text{CCA}}$ and R^2_{CCA} here, and note that they are often similar to PWCCA, and generally dominated by other measures. We expand each row of Table 2.1 into a subsection of its own. We also include p-values as reported by sci-kit learn, although we note that because random seeds are shared among some representations, these p-values are all inflated, with the exception of those for the experiment perturbing only fine-tuning seed, and assessing functionality through HANS (2.6). The invalid p-values may all be thought of as upper-bounds for the significance of the rank correlation results.

Perturbation: pretraining seed and layer depth

Tables 2.3 and 2.4 show the full results (including p-values and all 5 dissimilarity measures) using the QNLI probe as the functionality of interest, for Spearman ρ and Kendall's τ, respectively. Table 2.5 and 2.6 present results for the probing task SST-2 as the functionality of interest.

Table 2.3: Spearman ρ results for perturbing pretraining seed and layer depth, and assessing functionality through the QNLI probe

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
12	0.862 (6.5E-37)	0.876 (1.6E-39)	0.763 (2.2E-24)	0.849 (1.0E+00)	0.846 (1.0E+00)

Table 2.4: Kendall's τ results for perturbing pretraining seed and layer depth, and assessing functionality through the QNLI probe

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
12	0.670 (1.1E-27)	0.685 (7.4E-29)	0.564 (3.2E-20)	0.652 (1.0E+00)	0.647 (1.0E+00)

Table 2.5: Spearman ρ results for perturbing pretraining seed and layer depth, and assessing functionality through the SST-2 probe

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
12	0.890 (2.7E-42)	0.905 (5.3E-46)	0.829 (7.7E-32)	0.857 (1.0E+00)	0.854 (1.0E+00)

Table 2.6: Kendall's τ results for perturbing pretraining seed and layer depth, and assessing functionality through the SST-2 probe

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
12	0.707 (1.2E-30)	0.732 (1.0E-32)	0.637 (3.1E-25)	0.662 (1.0E+00)	0.658 (1.0E+00)

Perturbation: pretraining seed and principal component deletion

We find that for these experiments, results are consistent across the layers we analyze (the last 6 layers of BERT base). Tables 2.7 and 2.8 show results for Spearman ρ and Kendall's τ, respectively.

Table 2.7: Layer-wise Spearman ρ results for perturbing pretraining seed and principal component deletion, and assessing functionality through the SST-2 probe

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
8	0.764 (2.4E-36)	0.668 (3.2E-25)	0.776 (3.4E-38)	0.700 (1.9E-28)	0.700 (1.8E-28)
9	0.813 (2.1E-44)	0.706 (4.0E-29)	0.825 (9.2E-47)	0.728 (1.3E-31)	0.728 (1.2E-31)
10	0.873 (2.1E-58)	0.818 (2.7E-45)	0.874 (1.1E-58)	0.748 (3.2E-34)	0.749 (2.7E-34)
11	0.918 (1.2E-74)	0.797 (1.4E-41)	0.922 (1.7E-76)	0.781 (6.6E-39)	0.781 (7.0E-39)
12	0.932 (1.1E-81)	0.766 (1.1E-36)	0.955 (4.2E-97)	0.810 (6.1E-44)	0.810 (6.1E-44)

Table 2.8: Layer-wise Kendall's τ results for perturbing pretraining seed and principal component deletion, and assessing functionality through the SST-2 probe

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
8	0.560 (1.8E-29)	0.479 (4.4E-22)	0.573 (1.1E-30)	0.512 (6.8E-25)	0.512 (6.6E-25)
9	0.602 (1.2E-33)	0.509 (1.2E-24)	0.618 (2.5E-35)	0.542 (1.1E-27)	0.543 (9.7E-28)
10	0.684 (5.6E-43)	0.627 (2.1E-36)	0.685 (5.3E-43)	0.588 (2.9E-32)	0.589 (2.5E-32)
11	0.751 (2.8E-51)	0.616 (3.3E-35)	0.756 (6.4E-52)	0.648 (9.2E-39)	0.648 (9.2E-39)
12	0.787 (3.4E-56)	0.588 (2.9E-32)	0.819 (1.2E-60)	0.701 (4.7E-45)	0.701 (4.9E-45)

Perturbation: fine-tuning seed, Functionality: HANS

Results for this experiment are similar across layers for Procrustes and all three CCA-based measures, with middle layers of BERT base having a slightly higher rank correlation score in general. For CKA, this effect is even more pronounced. Tables 2.9 and 2.10 show the results for Spearman ρ and Kendall's τ, respectively.

Table 2.9: Layer-wise Spearman ρ results for perturbing finetuning seed, and assessing functionality through the HANS: Lexical (non-entailment) OOD dataset

Layer	Procrustes (p)	CKA (p)	PWCCA (p)	$\bar{\rho}_{\text{CCA}}$ (p)	R^2_{CCA} (p)
1	0.425 (5.1E-06)	0.361 (1.1E-04)	0.405 (1.4E-05)	0.388 (3.4E-05)	0.389 (3.2E-05)
2	0.510 (3.1E-08)	0.410 (1.2E-05)	0.486 (1.5E-07)	0.488 (1.3E-07)	0.483 (1.8E-07)
3	0.531 (6.6E-09)	0.427 (4.6E-06)	0.538 (3.8E-09)	0.533 (5.6E-09)	0.532 (6.2E-09)
4	0.543 (2.6E-09)	0.506 (3.9E-08)	0.552 (1.4E-09)	0.555 (1.0E-09)	0.550 (1.5E-09)
5	0.563 (5.3E-10)	0.512 (2.6E-08)	0.570 (2.9E-10)	0.582 (1.1E-10)	0.580 (1.3E-10)
6	0.629 (1.2E-12)	0.641 (3.6E-13)	0.621 (2.8E-12)	0.621 (2.7E-12)	0.622 (2.5E-12)
7	0.647 (1.7E-13)	0.658 (5.0E-14)	0.647 (1.7E-13)	0.653 (9.0E-14)	0.650 (1.2E-13)
8	0.643 (2.7E-13)	0.552 (1.3E-09)	0.653 (9.5E-14)	0.651 (1.1E-13)	0.651 (1.2E-13)
9	0.589 (5.9E-11)	0.419 (7.1E-06)	0.641 (3.5E-13)	0.662 (3.3E-14)	0.660 (4.2E-14)
10	0.536 (4.6E-09)	0.437 (2.7E-06)	0.559 (7.3E-10)	0.612 (6.6E-12)	0.614 (5.4E-12)
11	0.532 (6.2E-09)	0.426 (4.9E-06)	0.565 (4.7E-10)	0.619 (3.4E-12)	0.614 (5.5E-12)
12	0.465 (5.3E-07)	0.192 (2.8E-02)	0.574 (2.1E-10)	0.609 (9.2E-12)	0.610 (7.9E-12)

Table 2.10: Layer-wise Kendall's τ results for perturbing finetuning seed, and assessing functionality through the HANS: Lexical (non-entailment) OOD dataset

Layer	Procrustes (p)	CKA (p)	PWCCA (p)	$\bar{\rho}_{\text{CCA}}$ (p)	R^2_{CCA} (p)
1	0.295 (6.7E-06)	0.269 (3.6E-05)	0.277 (2.2E-05)	0.265 (4.7E-05)	0.268 (4.0E-05)
2	0.363 (4.6E-08)	0.288 (1.1E-05)	0.343 (2.1E-07)	0.342 (2.3E-07)	0.342 (2.4E-07)
3	0.372 (2.1E-08)	0.290 (9.5E-06)	0.378 (1.3E-08)	0.375 (1.6E-08)	0.375 (1.6E-08)
4	0.393 (3.4E-09)	0.358 (6.6E-08)	0.401 (1.7E-09)	0.405 (1.2E-09)	0.403 (1.4E-09)
5	0.410 (7.7E-10)	0.367 (3.3E-08)	0.417 (4.1E-10)	0.428 (1.4E-10)	0.424 (2.0E-10)
6	0.464 (4.2E-12)	0.474 (1.5E-12)	0.460 (6.3E-12)	0.460 (5.8E-12)	0.461 (5.6E-12)
7	0.483 (5.5E-13)	0.488 (3.3E-13)	0.481 (7.1E-13)	0.486 (3.9E-13)	0.483 (5.5E-13)
8	0.478 (9.2E-13)	0.392 (3.7E-09)	0.483 (5.7E-13)	0.481 (6.5E-13)	0.480 (7.7E-13)
9	0.432 (1.0E-10)	0.293 (7.7E-06)	0.475 (1.2E-12)	0.496 (1.3E-13)	0.494 (1.6E-13)
10	0.380 (1.0E-08)	0.306 (3.4E-06)	0.401 (1.7E-09)	0.447 (2.3E-11)	0.448 (2.1E-11)
11	0.376 (1.5E-08)	0.292 (8.3E-06)	0.411 (6.9E-10)	0.448 (2.1E-11)	0.445 (2.7E-11)
12	0.330 (5.7E-07)	0.127 (3.1E-02)	0.416 (4.4E-10)	0.446 (2.5E-11)	0.447 (2.2E-11)

Table 2.11: Layer-wise Spearman ρ results for perturbing pretraining seed and finetuning seed, and assessing functionality through the Antonymy stress test

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
1	0.252 (5.7E-03)	0.241 (7.8E-03)	0.168 (4.7E-02)	0.305 (1.0E+00)	0.327 (1.0E+00)
2	0.213 (1.7E-02)	0.145 (7.5E-02)	0.131 (9.7E-02)	0.047 (6.8E-01)	0.031 (6.2E-01)
3	0.260 (4.5E-03)	0.262 (4.2E-03)	0.208 (1.9E-02)	0.137 (9.1E-01)	0.111 (8.6E-01)
4	0.260 (4.5E-03)	0.265 (3.8E-03)	0.265 (3.8E-03)	0.276 (1.0E+00)	0.254 (9.9E-01)
5	0.273 (3.0E-03)	0.302 (1.1E-03)	0.278 (2.5E-03)	0.339 (1.0E+00)	0.310 (1.0E+00)
6	0.330 (3.9E-04)	0.280 (2.4E-03)	0.346 (2.1E-04)	0.313 (1.0E+00)	0.304 (1.0E+00)
7	0.271 (3.2E-03)	0.315 (7.1E-04)	0.111 (1.4E-01)	0.091 (8.2E-01)	0.090 (8.1E-01)
8	0.084 (2.0E-01)	0.004 (4.8E-01)	0.123 (1.1E-01)	0.204 (9.8E-01)	0.198 (9.8E-01)

Table 2.12: Layer-wise Kendall's τ results for perturbing pretraining seed and finetuning seed, and assessing functionality through the Antonymy stress test

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
1	0.199 (1.7E-03)	0.171 (5.9E-03)	0.126 (3.3E-02)	0.244 (1.0E+00)	0.243 (1.0E+00)
2	0.179 (4.3E-03)	0.123 (3.5E-02)	0.118 (4.2E-02)	0.061 (8.1E-01)	0.042 (7.3E-01)
3	0.185 (3.3E-03)	0.186 (3.2E-03)	0.139 (2.0E-02)	0.110 (9.5E-01)	0.096 (9.2E-01)
4	0.187 (3.0E-03)	0.191 (2.6E-03)	0.188 (2.9E-03)	0.206 (1.0E+00)	0.193 (1.0E+00)
5	0.192 (2.4E-03)	0.194 (2.2E-03)	0.202 (1.5E-03)	0.267 (1.0E+00)	0.242 (1.0E+00)
6	0.236 (2.7E-04)	0.197 (1.9E-03)	0.252 (1.1E-04)	0.229 (1.0E+00)	0.221 (1.0E+00)
7	0.189 (2.8E-03)	0.217 (7.3E-04)	0.091 (9.1E-02)	0.081 (8.8E-01)	0.082 (8.9E-01)
8	0.061 (1.9E-01)	-0.000 (5.0E-01)	0.101 (6.9E-02)	0.155 (9.9E-01)	0.150 (9.9E-01)

Perturbation: pretraining seeds and finetuning seeds of BERT medium

Rank correlation scores are low across the board for this task, suggesting that it is difficult for all existing dissimilarity measures, regardless of the layer within a network. Results on the Antonymy stress test for Spearman ρ and Kendall's τ are in Tables 2.11 and 2.12, respectively. Results on the Numerical stress test for Spearman ρ and Kendall's τ are in Tables 2.13 and 2.14, respectively.

Table 2.13: Layer-wise Spearman ρ results for perturbing pretraining seed and finetuning seed, and assessing functionality through the Numerical stress test

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
1	0.137 (8.7E-02)	0.108 (1.4E-01)	0.107 (1.4E-01)	0.072 (7.6E-01)	0.072 (7.6E-01)
2	-0.012 (5.5E-01)	0.060 (2.8E-01)	0.062 (2.7E-01)	0.004 (5.1E-01)	0.001 (5.0E-01)
3	-0.059 (7.2E-01)	0.011 (4.6E-01)	-0.031 (6.2E-01)	-0.060 (2.8E-01)	-0.056 (2.9E-01)
4	0.041 (3.4E-01)	0.052 (3.0E-01)	-0.026 (6.0E-01)	-0.101 (1.6E-01)	-0.084 (2.0E-01)
5	0.003 (4.9E-01)	0.131 (9.7E-02)	-0.047 (6.8E-01)	-0.061 (2.7E-01)	-0.061 (2.7E-01)
6	0.092 (1.8E-01)	0.260 (4.5E-03)	-0.029 (6.1E-01)	-0.064 (2.6E-01)	-0.056 (2.9E-01)
7	0.164 (5.2E-02)	0.250 (6.1E-03)	0.037 (3.6E-01)	0.040 (6.5E-01)	0.040 (6.5E-01)
8	0.202 (2.2E-02)	0.105 (1.5E-01)	0.175 (4.1E-02)	0.134 (9.1E-01)	0.143 (9.2E-01)

Table 2.14: Layer-wise Kendall's τ results for perturbing pretraining seed and finetuning seed, and assessing functionality through the Numerical stress test

Layer	Procrustes	CKA	PWCCA	$\bar{\rho}_{\text{CCA}}$	R^2_{CCA}
1	0.103 (6.5E-02)	0.083 (1.1E-01)	0.074 (1.4E-01)	0.050 (7.7E-01)	0.048 (7.6E-01)
2	-0.010 (5.6E-01)	0.046 (2.5E-01)	0.046 (2.5E-01)	0.006 (5.3E-01)	0.001 (5.0E-01)
3	-0.041 (7.3E-01)	0.014 (4.2E-01)	-0.018 (6.0E-01)	-0.047 (2.5E-01)	-0.047 (2.4E-01)
4	0.031 (3.2E-01)	0.038 (2.9E-01)	-0.020 (6.2E-01)	-0.076 (1.3E-01)	-0.065 (1.7E-01)
5	0.005 (4.7E-01)	0.086 (1.0E-01)	-0.031 (6.8E-01)	-0.042 (2.7E-01)	-0.042 (2.7E-01)
6	0.060 (1.9E-01)	0.175 (5.1E-03)	-0.020 (6.2E-01)	-0.050 (2.3E-01)	-0.046 (2.5E-01)
7	0.112 (4.9E-02)	0.168 (6.8E-03)	0.030 (3.3E-01)	0.019 (6.1E-01)	0.024 (6.4E-01)
8	0.131 (2.7E-02)	0.063 (1.8E-01)	0.125 (3.3E-02)	0.099 (9.3E-01)	0.103 (9.4E-01)

CIFAR-10C subtask-wise results

Table 2.15: Results for perturbing training seed and assessing functionality through CIFAR-10C

Table 2.16: Spearman ρ results

Corruption	Procrustes	CKA
gaussian_noise	0.083	0.076
shot_noise	0.171	0.161
impulse_noise	0.104	0.083
defocus_blur	-0.025	0.021
glass_blur	0.082	0.073
motion_blur	0.033	0.035
zoom_blur	-0.023	0.020
snow	0.087	0.060
frost	-0.062	-0.081
fog	-0.029	-0.039
brightness	0.122	0.110
contrast	-0.225	-0.145
elastic_transform	0.137	0.122
pixelate	0.118	0.098
jpeg_compression	0.149	0.102
speckle_noise	0.028	0.033
gaussian_blur	0.149	0.141
spatter	0.089	0.079
saturate	0.143	0.135
Average	0.060	0.057

Table 2.17: Kendall τ results

Corruption	Procrustes	CKA
gaussian_noise	0.057	0.050
shot_noise	0.118	0.110
impulse_noise	0.070	0.055
defocus_blur	-0.016	0.013
glass_blur	0.057	0.047
motion_blur	0.021	0.022
zoom_blur	-0.014	0.013
snow	0.059	0.042
frost	-0.046	-0.059
fog	-0.020	-0.025
brightness	0.084	0.077
contrast	-0.158	-0.102
elastic_transform	0.094	0.085
pixelate	0.081	0.066
jpeg_compression	0.103	0.070
speckle_noise	0.019	0.022
gaussian_blur	0.102	0.095
spatter	0.059	0.053
saturate	0.100	0.096
Average	0.041	0.038

Bootstrap significance testing for changing fine-tuning seeds

To assess whether the differences between rank correlations are statistically significant in the experiments varying finetuning seed and comparing functional behavior on the OOD HANS dataset, we conduct bootstrap resampling. Concretely, for every pair of metrics and every layer depth, we do the following:

- Sample 100 models with replacement, and collect their representations at the specified layer depth

- Let the reference A be the representation corresponding to the sampled model with maximum accuracy at that depth
- Compute the dissimilarities between A and the 100 sampled representations
- Compute the Kendall's τ and Spearman's ρ rank correlations for Orthogonal Procrustes, CKA, and PWCCA
- Record ρ(Procrustes) - ρ(CKA), ρ(PWCCA) - ρ(CKA), and ρ(PWCCA) - ρ(Procrustes), and the same pairwise differences for Kendall's τ.
- Repeat the above 2000 times

This gives us bootstrap distributions for the differences in rank correlations, and we may compute the 95% confidence intervals for these distributions. When the confidence interval does not overlap with 0, we conclude that the difference in rank correlation is statistically significant. The figures below show the results for each layer. We see that in the deeper layers of the network (layers 8-12), PWCCA has statistically significantly higher rank correlation than Orthogonal Procrustes, which in turn has statistically significantly higher rank correlation than CKA. In earlier layers, results are sometimes statistically significant, but not always.

Figure 2.4: Bootstrap comparison of ρ between metrics, layers 1-4

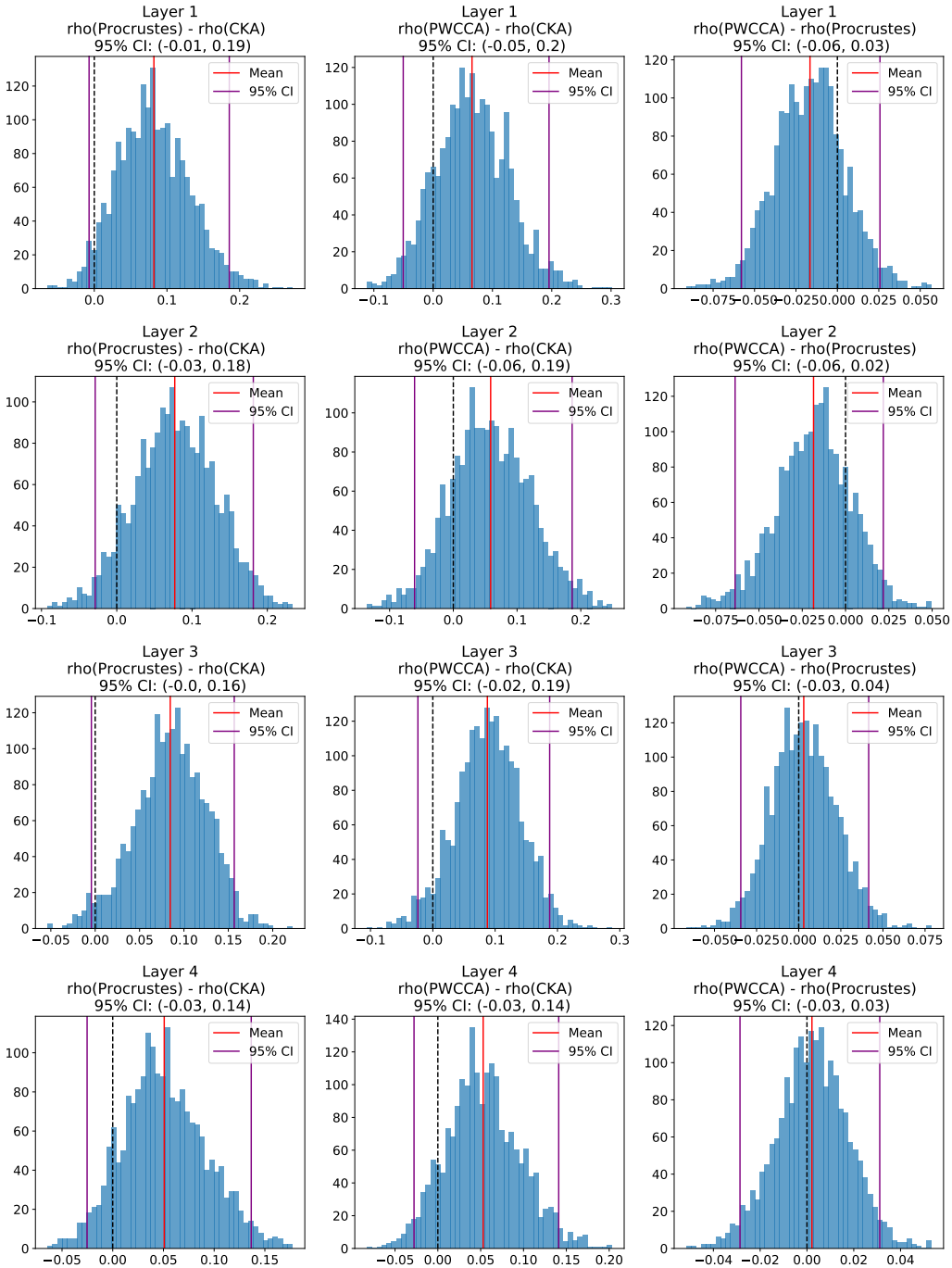


Figure 2.5: Bootstrap comparison of ρ between metrics, layers 5-8

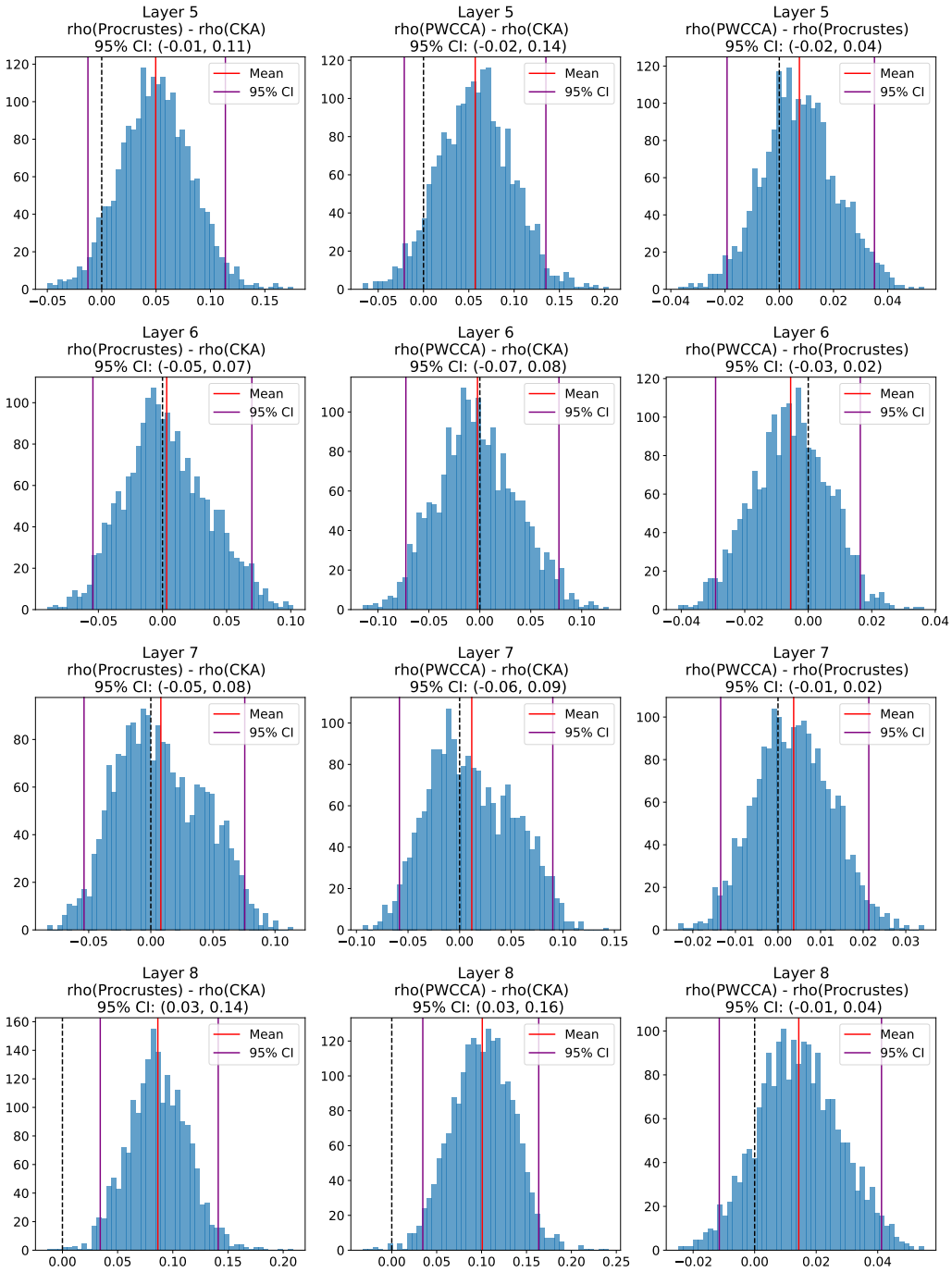


Figure 2.6: Bootstrap comparison of ρ between metrics, layers 9-12

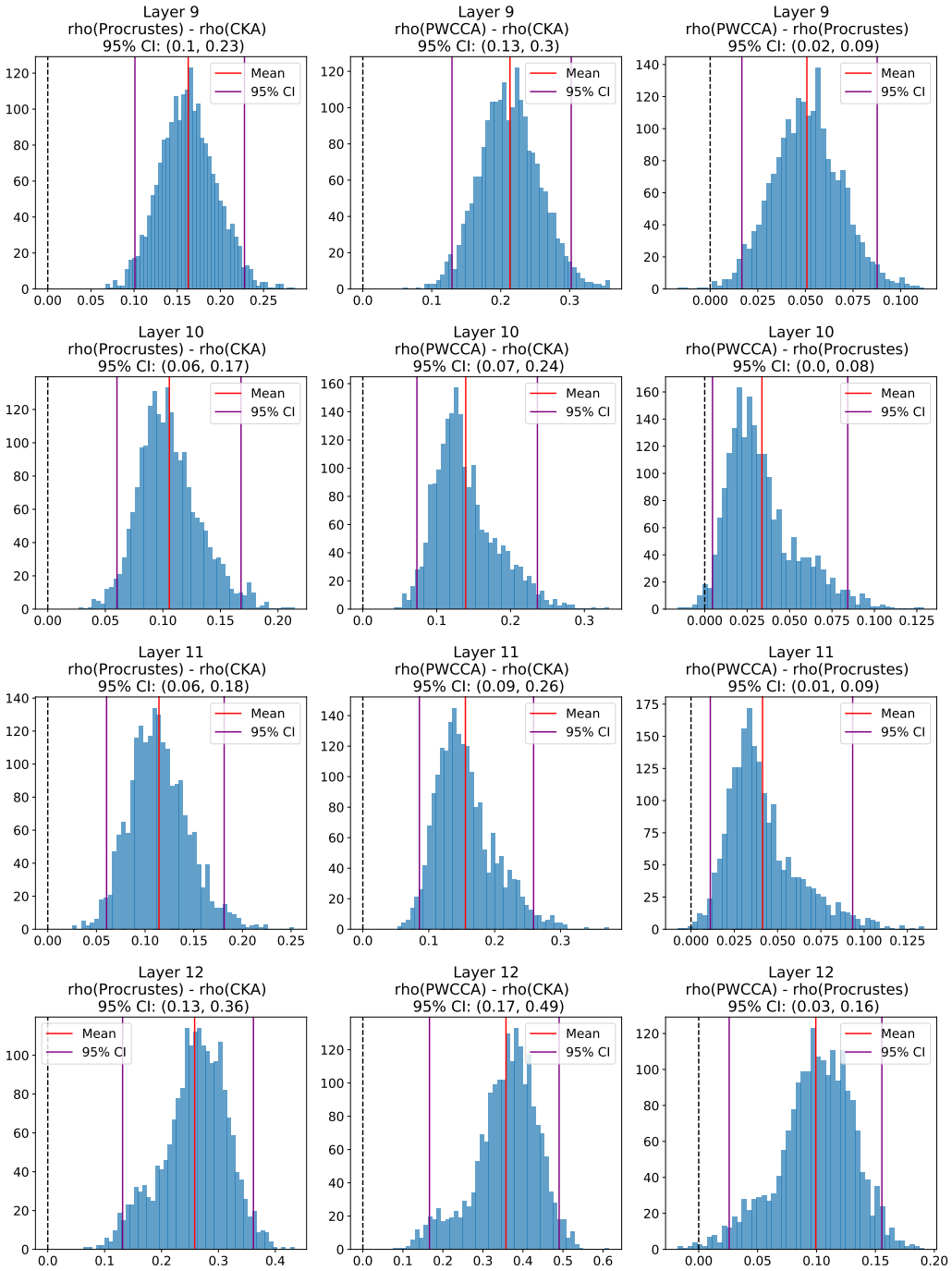


Figure 2.7: Bootstrap comparison of τ between metrics, layers 1-4

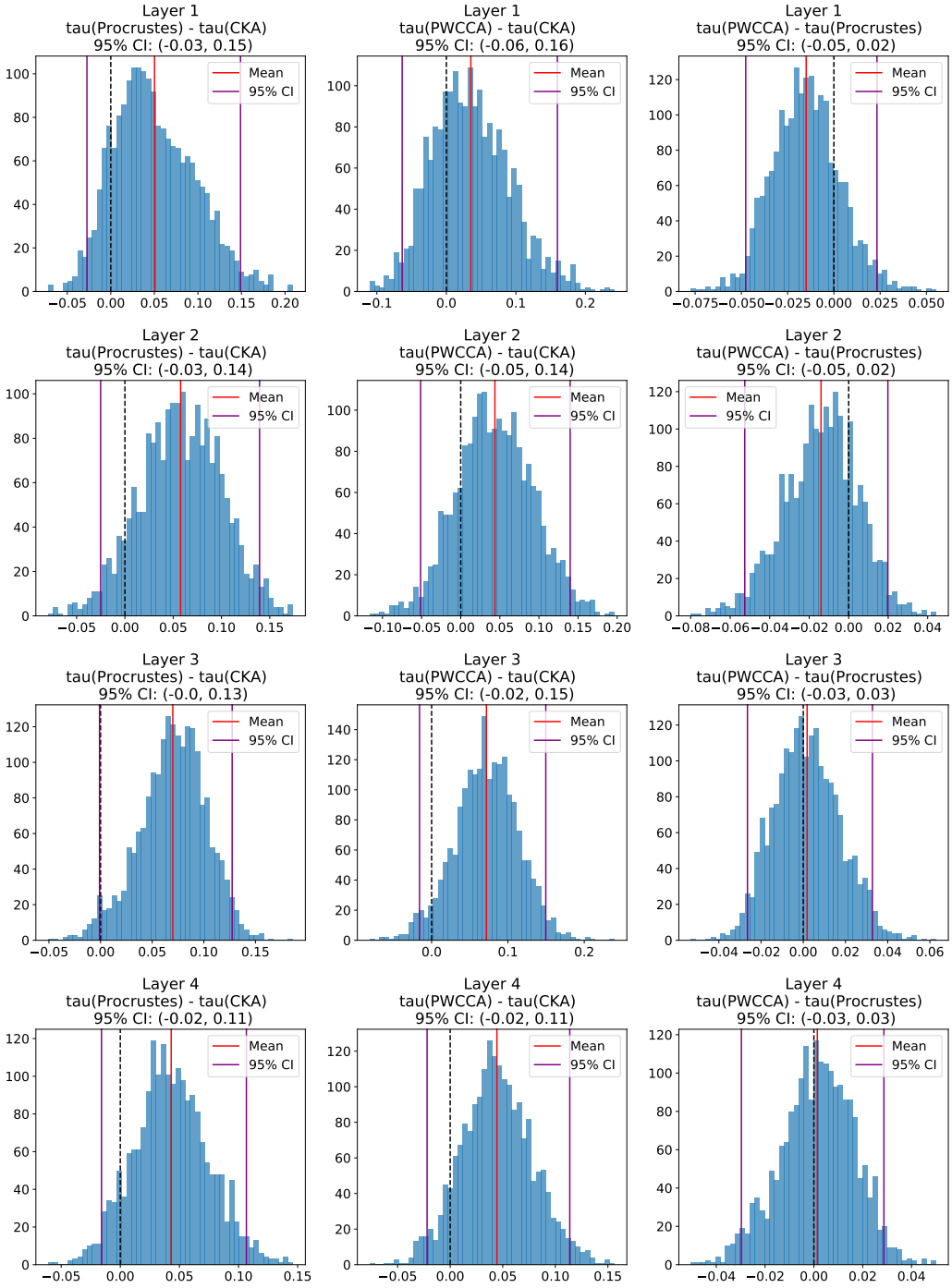


Figure 2.8: Bootstrap comparison of τ between metrics, layers 5-8

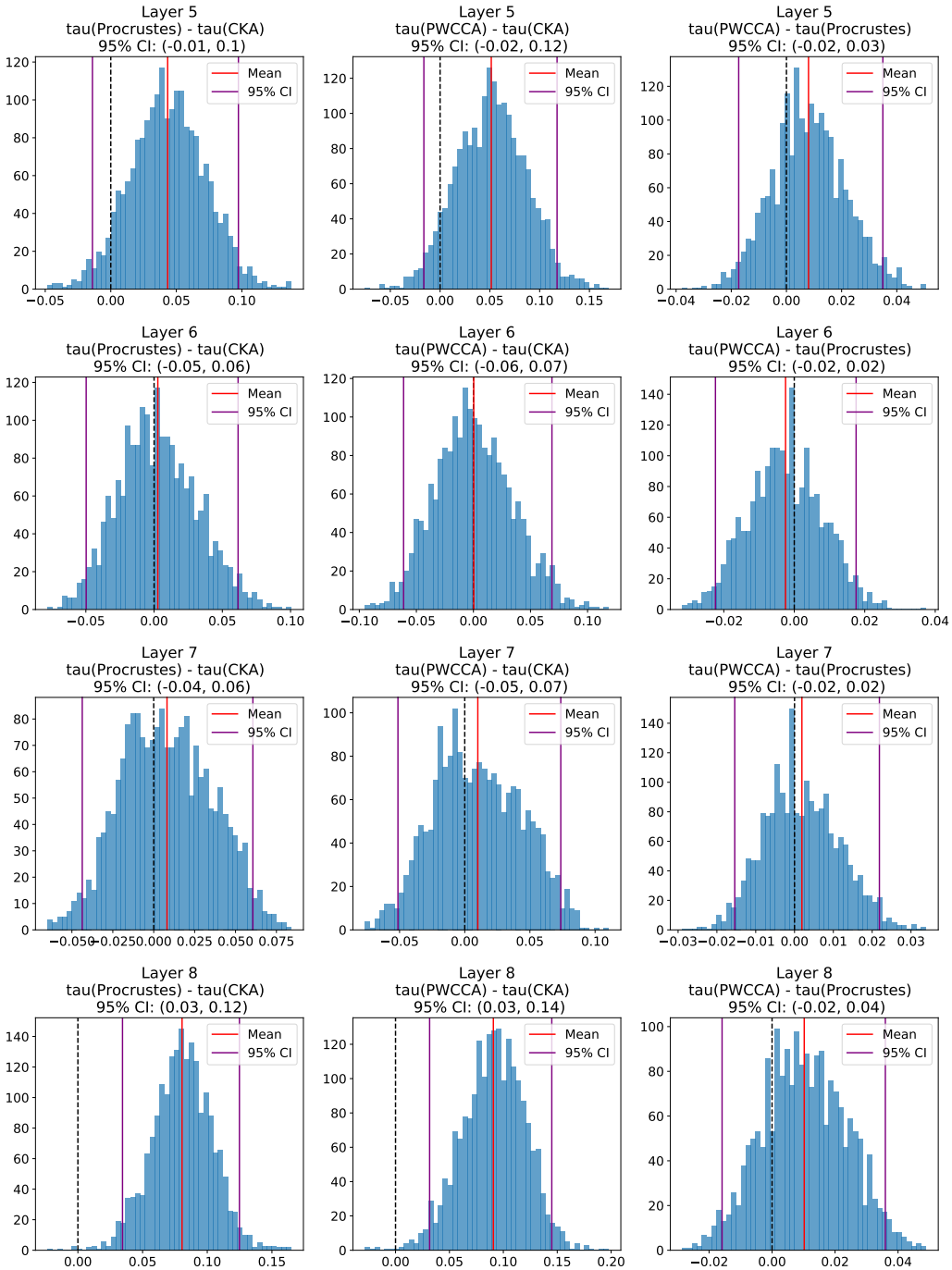
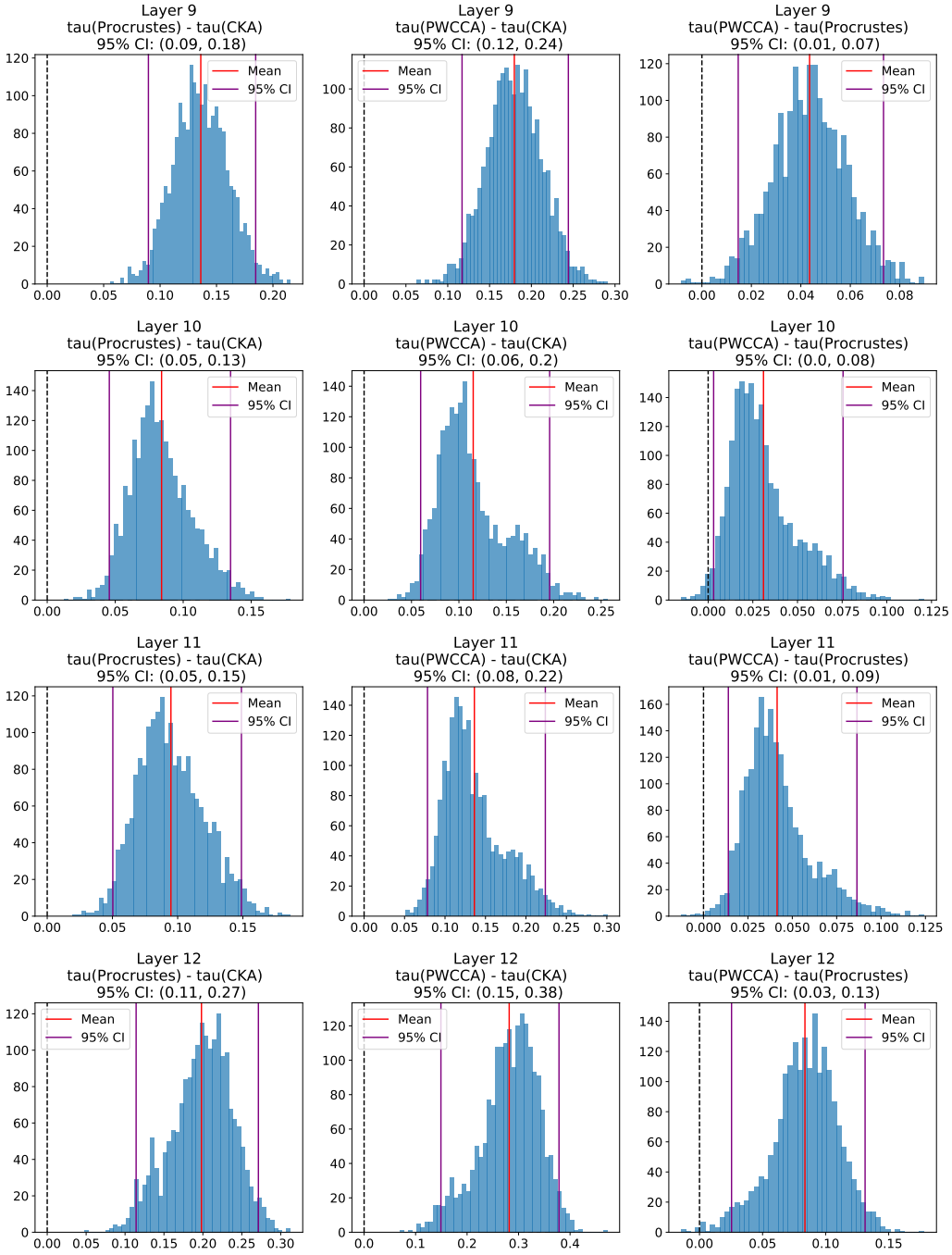


Figure 2.9: Bootstrap comparison of τ between metrics, layers 9-12



Chapter 3

Assessing Fair Machine Learning with New Datasets

3.1 Introduction

Datasets are central to the machine learning ecosystem. Besides providing training and testing data for model builders, datasets formulate problems, organize communities, and interface between academia and industry. Influential works relating to the ethics and fairness of machine learning recognize the centrality of datasets, pointing to significant harms associated with data, as well as better data practices [21, 47, 58, 94, 97]. While the discourse about data has prioritized cognitive domains such as vision, speech, or language, numerous consequential applications of predictive modeling and risk assessment involve bureaucratic, organizational, and administrative records best represented as tabular data [16, 40, 96].

When it comes to tabular data, surprisingly, most research papers on algorithmic fairness continue to involve a fairly limited collection of datasets, chief among them the *UCI Adult* dataset [62]. Derived from the 1994 Current Population Survey conducted by the US Census Bureau, this dataset has made an appearance in more than three hundred research papers related to fairness where it served as the basis for the development and comparison of many algorithmic fairness interventions.

Our work begins with a critical examination of the UCI Adult dataset—its origin, impact, and limitations. To guide this investigation we identify the previously undocumented exact source of the UCI Adult dataset, allowing us to reconstruct a superset of the data from available US Census records. This reconstruction reveals a significant idiosyncrasy of the UCI Adult prediction task that limits its external validity.

While some issues with UCI Adult are readily apparent, such as its age, limited documentation, and outdated feature encodings, a significant problem may be less obvious at first glance. Specifically, UCI Adult has a binary target label indicating whether the income of a person is greater or less than fifty thousand US dollars. This income threshold of $50k US dollars corresponds to the 76th quantile of individual income in the United States in

1994, the 88th quantile in the Black population, and the 89th quantile among women. We show how empirical findings relating to algorithmic fairness are sensitive to the choice of the income threshold, and how UCI Adult exposes a rather extreme threshold. Specifically, the magnitude of violations in different fairness criteria, trade-offs between them, and the effectiveness of algorithmic interventions all vary significantly with the income threshold. In many cases, the $50k threshold understates and misrepresents the broader picture.

Turning to our primary contribution, we provide a suite of new datasets derived from US Census data that extend the existing data ecosystem for research on fair machine learning. These datasets are derived from two different data products provided by the US Census Bureau. One is the Public Use Microdata Sample of the American Community Survey, involving millions of US households each year. The other is the Annual Social and Economic Supplement of the Current Population Survey. Both released annually, they represent major surveying efforts of the Census Bureau that are the basis of important policy decisions, as well as vital resources for social scientists.

We create prediction tasks in different domains, including income, employment, health, transportation, and housing. The datasets span multiple years and all states of the United States, in particular, allowing researchers to study temporal shift and geographic variation. Alongside these prediction tasks, we release a Python package called `folktables` which interfaces with Census data sources and allows users to both access our new predictions tasks and create new tasks from Census data through a simple API[1].

We contribute a broad initial sweep of new empirical insights into algorithmic fairness based on our new datasets. Our findings inform ongoing debates and in some cases challenge existing narratives about statistical fairness criteria and algorithmic fairness interventions. We highlight three robust observations:

1. Variation within the population plays a major role in empirical observations and how they should be interpreted:

   (a) Fairness criteria and the effect size of different interventions varies greatly by state. This shows that statistical claims about algorithmic fairness must be qualified carefully by context, even though they often are not.

   (b) Training on one state and testing on another generally leads to unpredictable results. Accuracy and fairness criteria could change in either direction. This shows that algorithmic tools developed in one context may not transfer gracefully to another.

   (c) Somewhat surprisingly, fairness criteria appear to be more stable over time than predictive accuracy. This is true both before and after intervention.

2. Algorithmic fairness interventions must specify a locus of intervention. For example, a model could be trained on the entire US population, or on a state-by-state basis. The

[1]The datasets and Python package are available for download at `https://github.com/zykls/folktables`.

results differ significantly. Recognition of the need for such a choice is still lacking, as is scholarship guiding the practitioner on how to navigate this choice and its associated trade-offs.

3. Increased dataset size does not necessarily help in reducing observed disparities. Neither does social progress as measured in years passed. This is in contrast to intuition from cognitive machine learning tasks where more representative data can improve metrics such as error rate disparities between different groups.

Our observations apply to years of active research into algorithmic fairness, and our work provides new datasets necessary to re-evaluate and extend the empirical foundations of the field.

3.2 Archaeology of UCI Adult: Origin, Impact, Limitations

> *Archaeology organises the past to understand the present. It lifts the dust-cover off a world that we take for granted. It makes us reconsider what we experience as inevitable.*
>
> — Ian Hacking

Although taken for granted today, the use of benchmark datasets in machine learning emerged only in late 1980s [50]. Created in 1987, the UCI Machine Learning Repository contributed to this development by providing researchers with numerous datasets each with a fixed training and testing split [69]. As of writing, the UCI Adult dataset is the second most popular dataset among more than five hundred datasets in the UCI repository. An identical dataset is called "Census Income Data Set" and a closely related larger dataset goes by "Census-Income (KDD) Data Set".

At the outset, UCI Adult contains 48,842 rows each apparently describing one individual with 14 attributes. The dataset information reveals that it was extracted from the "1994 Census database" according to certain filtering criteria. Since the US Census Bureau provides several data products, as we will review shortly, this piece of information does not identify the source of the dataset.

The fourteen features of UCI Adult include what the fairness community calls *sensitive* or *protected* attributes, such as, age, sex, and race. The earliest paper on algorithmic fairness that used UCI Adult to our knowledge is a work by Calders et al. [22] from 2009. The availability of sensitive attributes contributed to the choice of the dataset for the purposes of this work. An earlier paper in this context by Pedreschi et al. [99] used the UCI German credit dataset, which is smaller and ended up being less widely used in the community. Another highly cited paper on algorithmic fairness that popularized UCI Adult is the work of Zemel et al. [130] on *learning fair representations* (LFR). Published in 2013, the work introduced the idea of changing the data representation to achieve a particular fairness criterion, in this

case, demographic parity, while representing the original data as well as possible. This idea remains popular in the community and the LFR method has become a standard baseline.

Representation learning is not the only topic for which UCI Adult became the standard test case. The dataset has become broadly used throughout the area for purposes including the development of new fairness criteria, algorithmic interventions and fairness promoting methods, as well as causal modeling. Major software packages, such as AI Fairness 360 [15] and Fairlearn [17], expose UCI Adult as one of a few standard examples. Indeed, based on bibliographic information available on Google Scholar there appear to be more than 300 papers related to algorithmic fairness that used the UCI Adult dataset at the time of writing.

Reconstruction of UCI Adult

Creating a dataset involves a multitude of design choices that substantially affect the validity of experiments conducted with the dataset. To fully understand the context of UCI Adult and explore variations of its design choices, we reconstructed a closely matching superset from the original Census sources. We now describe our reconstruction in detail and then investigate one specific design choice, the income binarization threshold, in Section 3.2.

The first step in our reconstruction of UCI Adult was identifying the original data source. As mentioned above, the "1994 census database" description in the UCI Adult documentation does not uniquely identify the data product provided by the US Census Bureau. Based on the documentation of the closely related "Census-Income (KDD) Data Set,"[2] we decided to start with the Current Population Survey (CPS) data, specifically the Annual Social and Economic Supplement (ASEC) from 1994. We utilized the IPUMS interface to the CPS data [44] and hence refer to our reconstruction as IPUMS Adult.

The next step in the reconstruction was matching the 15 features in UCI Adult to the CPS data. This was a non-trivial task: the UCI Adult documentation does not mention any specific CPS variable names and IPUMS CPS contains more than 400 candidate variables for the 1994 ASEC. To address this challenge, we designed the following matching procedure that we repeated for each feature in UCI Adult: First, identify a set of candidate variables in CPS via the IPUMS keyword search. For each candidate variable, use the CPS documentation to manually derive a mapping from the CPS encoding to the UCI Adult encoding. Finally, match each row in UCI Adult to its nearest neighbor in the partial reconstruction assembled from previous exact variable matches.

We only included a candidate variable if the nearest neighbor match was *exact*, i.e., we could find an exact match in the IPUMS CPS data for each row in UCI Adult that matched *both* the candidate variable and all earlier variables also identified via exact matches. There were only two exceptions to this rule. We discuss them in Appendix 3.6. After completing the variable matching, our reconstruction has 49,531 rows when we use the same inclusion criteria as UCI Adult to the extent possible, which is slightly more than the 48,842 rows in UCI Adult. The discrepancy likely stems from the fact that UCI Adult used the variable

[2]Ron Kohavi is a co-creator of both datasets.

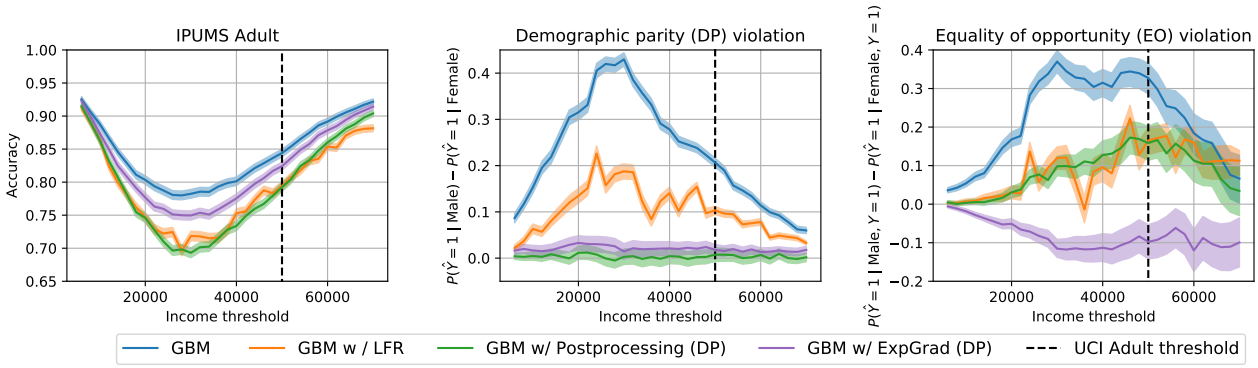


Figure 3.1: Fairness interventions with varying income threshold on IPUMS Adult. We compare three methods for achieving demographic parity: a pre-processing method (LFR), an in-training method based on Agarwal et al. [3] (ExpGrad), and a post-processing adjustment method [51]. We apply each method using a gradient boosted decision tree (GBM) as the base classifier. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for DP.

"fnlwgt" in its inclusion criteria and we did not due to the lack of an exact match for this variable. This made our inclusion criteria slightly more permissive than those of UCI Adult. The fact that we found exact matches for 13 of the 15 UCI Adult variables and a very close match for "native-country" is evidence that our reconstruction of UCI Adult is accurate.

Varying income threshold

The goal in the UCI Adult dataset is to predict whether an individual earns greater than 50,000 US dollars a year. The choice of the 50,000 dollar threshold is idiosyncratic and potentially limits the external validity of UCI Adult as a benchmark for algorithmic fairness. In 1994, the median US income was 26,000 dollars, and 50,000 dollars corresponds to the 76th quantile of the income distribution, and the 88th and 89th quantiles of the income distribution for the Black and female populations, respectively. Consequently, *almost all of the Black and female instances in the dataset fall below the threshold* and models trained on UCI adult tend to have substantially higher accuracies on these subpopulations. For instance, a standard logistic regression model trained on UCI Adult dataset achieves 85% accuracy overall, 91.4% accuracy on the Black instances, and 92.7% on Female instances. This is a rather untypical situation since often machine learning models perform more poorly on historically disadvantaged groups.

To understand the sensitivity of the empirical findings on UCI Adult to the choice of threshold, we leverage our IPUMS Adult reconstruction, which includes the continuous, unthresholded income variable, and construct a new collection of datasets where the income threshold varies from 6,000 to 70,000. For each threshold, we first train a standard gradient

boosted decision tree and evaluate both its accuracy and its violation of two common fairness criteria: *demographic parity* (equality of positive rates) and *equal opportunity* (equality of true positive rates). See the text by Barocas et al. [13] for background. The results are presented in Figure 3.1, where we see both accuracy and the magnitude of violations of these criteria vary substantially with the threshold choice.

We then evaluate how the choice of threshold affects three common classes of fairness interventions: the preprocessing method LFR [130] mentioned earlier, an *in-processing* or *in-training* method based on the reductions approach in Agarwal et al. [3], and the post-processing method from Hardt et al. [51]. In Figure 3.1, we plot model accuracy after applying each intervention to achieve demographic parity as well as violations of both demographic parity and equality of opportunity as the income threshold varies. In Appendix 3.6, we conduct the same experiment for methods to achieve equality of opportunity. There are three salient findings. First, the effectiveness of each intervention depends on the threshold. For values of the threshold near 25,000, the accuracy drop needed to achieve demographic parity or equal opportunity is significantly larger than closer to 50,000. Second, the trade-offs between different criteria vary substantially with the threshold. Indeed, for the in-processing method enforcing demographic parity, as the threshold varies, the equality of opportunity violation is monotonically increasing. Third, for high values of the threshold, the small number of positive instances substantially enlarges the confidence intervals for equality of opportunity, which makes it difficult to meaningfully compare the performance of methods for satisfying this constraint.

3.3 New datasets for algorithmic fairness

At least one aspect of UCI Adult is remarkably positive. The US Census Bureau invests heavily in high quality data collection, surveying methodology, and documentation based on decades of experience. Moreover, responses to some US Census Bureau surveys are legally mandated and hence enjoy high response rates resulting in a representative sample. In contrast, some notable datasets in machine learning are collected in an ad-hoc manner, plagued by skews in representation [18, 23, 120, 128], often lacking copyright [73] or consent from subjects [101], and involving unskilled or poorly compensated labor in the form of crowd workers [48].

In this work, we tap into the vast data ecosystem of the US Census Bureau to create new machine learning tasks that we hope help to establish stronger empirical evaluation practices within the algorithmic fairness community.

As previously discussed, UCI Adult was derived from the Annual Social and Economic Supplement (ASEC) of the Current Population Survey (CPS). The CPS is a monthly survey of approximately 60,000 US households. It's used to produce the official monthly estimates of employment and unemployment for the United States. The ASEC contains additional information collected annually.

Table 3.1: New prediction task details instantiated on 2018 US-wide ACS PUMS data

Task	Features	Datapoints	Constant predictor acc	LogReg acc	GBM acc
ACSIncome	10	1,664,500	63.1%	77.1%	79.7%
ACSPublicCoverage	19	1,138,289	70.2%	75.6%	78.5 %
ACSMobility	21	620,937	73.6%	73.7%	75.7%
ACSEmployment	17	3,236,107	56.7%	74.3%	78.5%
ACSTravelTime	16	1,466,648	56.3%	57.4%	65.0%

Another US Census data product most relevant to us are the American Community Survey (ACS) Public Use Microdata Sample (PUMS). ACS PUMS differs in some significant ways from CPS ASEC. The ACS is sent to approximately 3.5 million US households each year gathering information relating to ancestry, citizenship, education, employment, language proficiency, income, disability, and housing characteristics. Participation in the ACS is mandatory under federal law. Responses are confidential and governed by strict privacy rules. The Public Use Microdata Sample contains responses to every question from a subset of respondents. The geographic information associated with any given record is limited to a level that aims to prevent re-identification of survey participants. A number of other disclosure control heuristics are implemented. Extensive documentation is available on the websites of the US Census Bureau.

Available prediction tasks

We use ACS PUMS as the basis for the following new prediction tasks:

ACSIncome: predict whether an individual's income is above $50,000, after filtering the ACS PUMS data sample to only include individuals above the age of 16, who reported usual working hours of at least 1 hour per week in the past year, and an income of at least $100. The threshold of $50,000 was chosen so that this dataset can serve as a replacement to UCI Adult, but we also offer datasets with other income cutoffs described in Appendix 3.6.

ACSPublicCoverage: predict whether an individual is covered by public health insurance, after filtering the ACS PUMS data sample to only include individuals under the age of 65, and those with an income of less than $30,000. This filtering focuses the prediction problem on low-income individuals who are not eligible for Medicare.

ACSMobility: predict whether an individual had the same residential address one year ago, after filtering the ACS PUMS data sample to only include individuals between the ages of 18 and 35. This filtering increases the difficulty of the prediction task, as the base rate of staying at the same address is above 90% for the general population.

ACSEmployment: predict whether an individual is employed, after filtering the ACS PUMS data sample to only include individuals between the ages of 16 and 90.

ACSTravelTime: predict whether an individual has a commute to work that is longer than 20 minutes, after filtering the ACS PUMS data sample to only include individuals who are employed and above the age of 16. The threshold of 20 minutes was chosen as it is the US-wide median travel time to work in the 2018 ACS PUMS data release.

All our tasks contain features for age, race, and sex, which correspond to *protected categories* in different domains under US anti-discrimination laws [12]. Further, each prediction task can be instantiated on different ACS PUMS data samples, allowing for comparison across geographic and temporal variation. We provide datasets for each task corresponding to 1) all fifty US states and Puerto Rico, and 2) five different years of data collection: 2014–2018 inclusive, resulting in a total of 255 distinct datasets per task to assess distribution shift. We also provide US-wide datasets for each task, constructed from concatenating each state's data. Table 3.1 displays more details about each prediction task as instantiated on the 2018 US-wide ACS PUMS data sample. Our new tasks constitute a diverse collection of prediction problems ranging from those where machine learning achieves significantly higher accuracy than a baseline constant predictor to other potentially low-signal problems (ACSMobility) where accuracy improvement appears to be more challenging. We also provide the exact features included in each prediction task, and other details, in Appendix 3.6. A datasheet [47] for our datasets is provided in Appendix 3.7.

These prediction tasks are by no means exhausitive of the potential tasks one can construct using the ACS PUMS data. The `folktables` package we introduce provides a simple API that allows users to construct new tasks using the ACS PUMS data, and we encourage the community to explore additional prediction tasks beyond those introduced in this paper.

Scope and limitations

One distinction is important. Census data is often used by social scientists to study the extent of inequality in income, employment, education, housing or other aspects of life. Such important substantive investigations should necessarily inform debates about discrimination in classification scenarios within these domains. However, our contribution is not in this direction. We instead use census data for the empirical study of algorithmic fairness. This generally may include performance claims about specific methods, the comparison of different methods for achieving a given fairness metric, the relationships of different fairness criteria in concrete settings, causal modeling of different scenarios, and the ability of different methods to transfer successfully from one context to another. We hope that our work leads to more comprehensive empirical evaluations in research papers on the topic, at the very least reducing the overreliance on UCI Adult and providing a complement to the flourishing theoretical work on the topic. The distinction we draw between benchmark data and substantive domain-specific investigations resonates with recent work that points out issues with using data about risk assessments tools from the criminal justice domain as machine learning benchmarks [11].

A notable if obvious limitation of our work is that it is entirely US-centric. A richer dataset ecosystem covering international contexts within the algorithmic fairness community is still lacking. Although empirical work in the Global South is central in other disciplines,

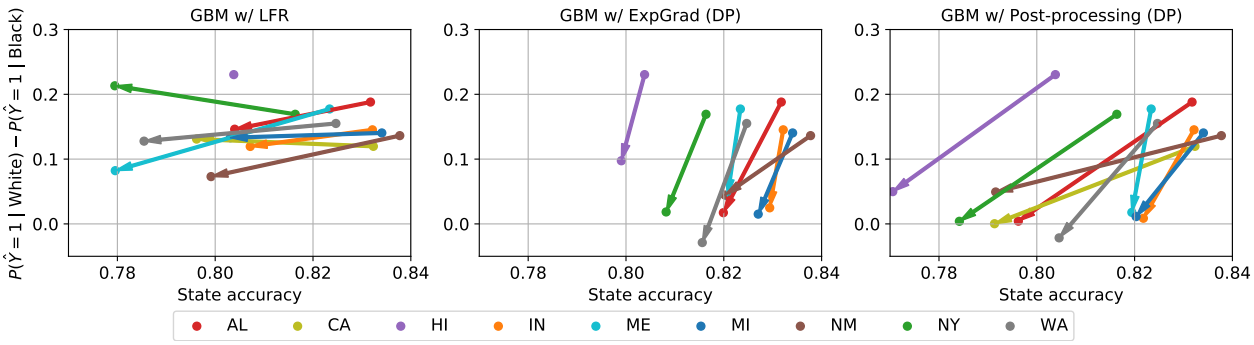


Figure 3.2: The effect size of fairness interventions varies by state. Each panel shows the change in accuracy and demographic parity on the ACSIncome task after applying a fairness intervention to an unconstrained gradient boosted decision tree (GBM). Each arrow corresponds to a different state distribution. The arrow base represents the (accuracy, DP) point corresponding to the unconstrained GBM, and the head represents the (accuracy, DP) point obtained after applying the intervention. The arrow for HI in the LFR plot is entirely covered by the start and end points.

there continues to be much need for the North American fairness community to engage with it more strongly [1].

3.4 A tour of empirical observations

In this section, we highlight an initial sweep of empirical observations enabled by our new ACS PUMS derived prediction tasks. Our experiments focus on three fundamental issues in fair machine learning: (i) variation within the population of interest, e.g., how does the effectiveness of interventions vary between different states or over time?, (ii) the locus of intervention, e.g. should interventions be performed at the state or national level?, and (iii) whether increased dataset size or the passage of time mitigates observed disparities?

Our experiments are not exhaustive and are intended to highlight the perspective a broader empirical evaluation with our new datasets can contribute to addressing questions within algorithmic fairness. The goal of the experiments is not to provide a complete overview of all the questions that one can answer using our datasets. Rather, we hope to inspire other researchers to creatively use our datasets to further probe these question as well as propose new ones leveraging the ACS PUMS data.

Variation within the population

The ACS PUMS prediction tasks present two natural axes of variation: geographic variation between states and temporal variation between years the ACS is conducted. This variation allows us to both measure the performance of different fairness interventions on a broad

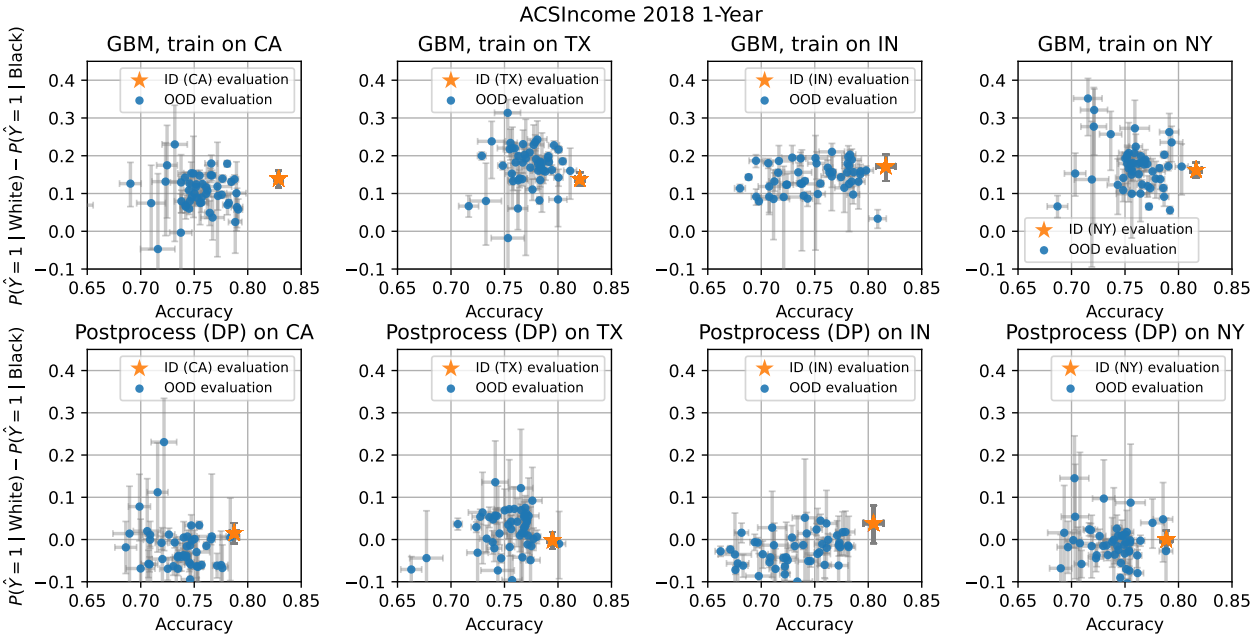


Figure 3.3: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSIncome task and evaluated both in-distribution (ID) on the same state and out-of-distribution (OOD) on the 49 other states in terms of accuracy and demographic parity violation. **Bottom:** Each panel shows an GBM with post-processing to enforce demographic parity on the state on which it was trained and evaluated both ID and OOD on all 50 states. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for demographic parity.

collection of different distributions, as well as study the performance of these interventions under geographical and temporal *distribution shift* when the test dataset differs from the one on which the model was trained.

Due to space constraints, we focus our experiments in this section on the ACSIncome prediction task with demographic parity as the fairness criterion of interest. We present similar results for our other prediction tasks and fairness criteria, as well as full experimental details in Appendix 3.6.

Intervention effect sizes vary across states. The fifty US states which comprise the ACS PUMS data present a broad set of different experimental conditions on which to evaluate the performance of fairness interventions. At the most basic level, we can train and evaluate different fairness interventions on each of the states and compare the interventions' efficacy on these different distributions. Concretely, we first train an unconstrained gradient boosted decision tree (GBM) on each state, and we compare the accuracy and fairness

criterion violation of this unconstrained model with the same model after applying one of three common fairness intervention: pre-processing (LFR), the in-processing fair reductions methods from Agarwal et al. [3] (ExpGrad), and the simple post-processing method that adjusts group-based acceptance thresholds to satisfy a constraint [51]. Figure 3.2 shows the result of this experiment for the ACSIncome prediction task for interventions to achieve demographic parity. For a given method, performance can differ markedly between states. For instance, LFR decreases the demographic parity violation by 10% in some states and in other states the decrease is close to zero. Similarly, the post-processing adjustment to enforce demographic parity incurs accuracy drops of less than 1% in some states, whereas in others the drop is closer to 5%.

Training and testing on different states leads to unpredictable results. Beyond training and evaluating interventions on different states, we also use the ACS PUMS data to study the performance of interventions under *geographic* distribution shift, where we train a model on one state and test it on another. In Figure 3.3, we plot accuracy and demographic parity violation with respect to race for both an unconstrained GBM and the same model after applying a post-processing adjustment to achieve demographic parity on a natural suite of test sets: the in-distribution (same state test set) and the out-of-distribution test sets for the 49 other states. For both the unconstrained and post-processed model, model accuracy and demographic parity violation varies substantially across different state test sets. In particular, even when a method achieves demographic parity in one state, it may no longer satisfy the fairness constraint when naively deployed on another.

Fairness criteria are more stable over time than predictive accuracy. In contrast to the unpredictable results that occur under geographic distribution shift, the fairness criteria and interventions we study are much more stable under *temporal* distribution shift. Specifically, in Figure 3.4, we plot model accuracy and demographic parity violation for GBM trained on the ACSIncome task using US-wide data from 2014 and evaluated on the test sets for the same task drawn from years 2014-2018. Perhaps unsurprisingly, model accuracy degrades slightly over time. However, the associated fairness metric is stable and essentially constant over time. Moreover, this same trend holds for the fairness interventions previously discussed. The same base GBM with pre-processing (LFR), in-processing (ExpGrad), or post-processing to satisfy demographic parity in 2014, all have a similar degradation in accuracy, but the fairness metrics remain stable. Thus, a classifier that satisfies demographic parity on the 2014 data continues to satisfy the constraint on 2015-2018 data.

Specifying a locus of intervention

On the ACSPUMs prediction task, fairness interventions can be applied either on a state-by-state basis or on the entire US population. In Table 3.2, we compare the performance of LFR and the post-processing adjustment method applied at the US-level with the aggregate performance of both methods applied on a state-by-state basis, using a GBM as the base

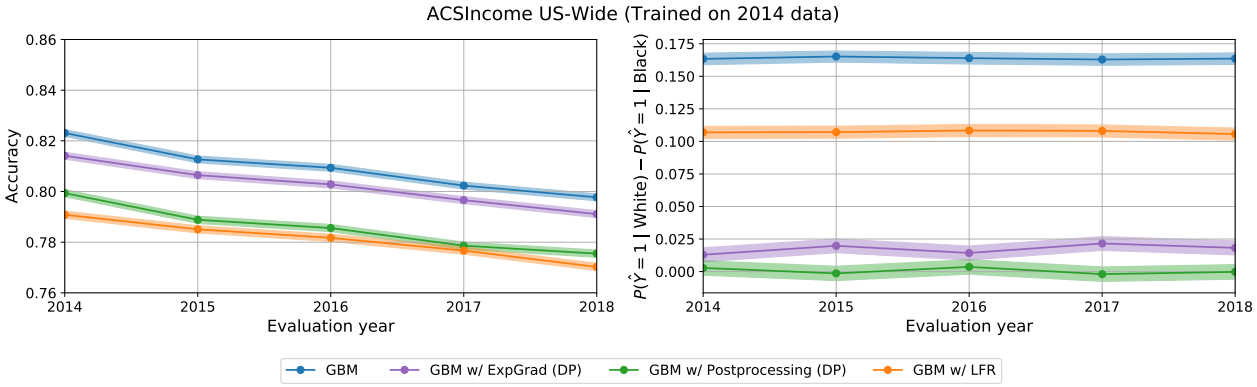


Figure 3.4: Fairness criteria are more stable over time than accuracy. **Left:** Models trained in 2014 on US-wide ACSIncome with and without fairness interventions to achieve demographic parity and evaluated on data in subsequent years suffer a drop in accuracy over time. **Right:** However, the violation of demographic parity remains essentially constant over time. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for demographic parity.

Table 3.2: Comparison of two different strategies for applying an intervention to achieve demographic parity (DP) on the US-wide ACSIncome task. *US-level* corresponds to training one classifier and applying the intervention on the entire US population. *State-level* corresponds to training a classifier and applying the intervention separately for each state and then aggregating the results over all states. Here, DP refers to $P(\hat{Y} = 1 \mid \text{White}) - P(\hat{Y} = 1 \mid \text{Black})$. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for DP.

	US-level acc	US-level DP violation	State-level acc	State-level DP violation
Unconstrained GBM	81.7 ± 0.1 %	17.7 ± 0.2%	82.8 ± 0.1 %	16.9 ± 0.2%
GBM w/ LFR	78.7 ± 0.1 %	16.6 ± 0.2%	79.4 ± 0.1%	14.0 ± 0.2%
GBM w/ post-processing (DP)	79.2 ± 0.1 %	0.3 ± 0.3 %	80.2 ± 0.1%	-0.6 ± 0.3%

classifier. In both cases, applying the intervention on a state-by-state improves US-wide accuracy while still preserving demographic parity (post-processing) or further mitigating violations of demographic parity (LFR).

Table 3.3: Disparities persist despite increasing dataset size and social progress.

Dataset	Year	Datapoints	GBM acc	TPR White	TPR Black	TPR disparity
IPUMS Adult	1994	49,531	86.4%	58.0%	46.5 %	11.5%
ACSIncome	2018	1,664,500	80.8%	66.5%	51.7%	14.8%

Increased dataset size doesn't necessarily mitigate observed disparities

To mitigate disparities in error rates, commonly suggested remedies include collecting a) larger datasets and b) more representative data reflective of social progress. For example, in response to research revealing the stark accuracy disparities of commercial facial recognition algorithms, particularly for dark-skinned females [21], IBM collected a more diverse training set of images, retrained its facial recognition model, and reported a 10-fold decrease in error for this subgroup [103]. However, on our tabular datasets, larger datasets collected in more socially progressive times do not automatically mitigate disparities. Table 3.3 shows that unconstrained gradient boosted decision tree trained on a newer, larger dataset (ACSIncome vs. IPUMS Adult), does not improve disparities such as in true positive rate (TPR). A fundamental reason for this is the persistent social inequality that is reflected in the data. It is well known that given a disparity in base rates between groups, a predictive model cannot be both calibrated and equal in error rates across groups [27], except if the model has 100% accuracy. This observation highlights a key difference between cognitive machine learning and tabular data prediction – the Bayes error rate is zero for cognitive machine learning. Thus larger and more representative datasets eventually address disparities by pushing error rates to zero for all subgroups. In the tabular datasets we collect, the Bayes error rate of an optimal classifier is almost certainly far from zero, so some individuals will inevitably be incorrectly classified. Rather than hope for future datasets to implicitly address disparities, we must directly contend with how dataset and model design choices distribute the burden of these errors.

3.5 Discussion and future directions

Rather than settled conclusions, our empirical observations are intended to spark additional work on our new datasets. Of particular interest is a broad and comprehensive evaluation of existing methods on all datasets. We only evaluated some methods so far. One interesting question is if there is a method for achieving either demographic parity or error rate parity that outperforms threshold adjustment (based on the best known unconstrained classifier) on any of our datasets? We conjecture that the answer is *no*. The reason is that we believe on our datasets a well-tuned tree-ensemble achieves classification error close to the Bayes error

bound. Existing theory (Theorem 5.3 in [51]) would then show that threshold adjustment based on this model is, in fact, optimal. Our conjecture motivates drawing a distinction between classification scenarios where a nearly Bayes optimal classifier is known and those where there isn't. How close we are to Bayes optimal on any of our new prediction tasks is a good question. The role of distribution shift also deserves more attention. Are there methods that achieve consistent performance across geographic contexts? Why does there appear to be more temporal than geographic stability? What does the sensitivity to distribution shift say about algorithmic tools developed in one context and deployed in another? Answers to these questions seem highly relevant to policy-making around the deployment of algorithmic risk assessment tools. Finally, our datasets are also interesting test cases for causal inference methods, which we haven't yet explored. How would, for example, methods like *invariant risk minimization* [7] perform on different geographic contexts?

3.6 Supplementary Materials

Adult reconstruction

Additional reconstruction details

We only included a candidate variable if the nearest neighbor match was *exact*, i.e., we could find an exact match in the IPUMS CPS data for each row in UCI Adult that matched *both* the candidate variable and all earlier variables also identified via exact matches. There were only two exceptions to this rule:

- The UCI Adult feature "native-country". Here we could match the vast majority of rows in UCI Adult to the IPUMS CPS variable "UH_NATVTY_A1". To get an exact match for all rows, we had to map the country codes for Russia and Guyana in "UH_NATVTY_A1" to the value for "unknown". The documentation for UCI Adult also mentions neither Russia nor Guyana as possible values for "native-country". We do not know the reason for this discrepancy.

- The UCI Adult feature "fnlwgt". This column is actually not a demographic feature of an individual but a weight value computed by the Census Bureau to make the sample representative for the US population. We compared the "fnlwgt" data to all weight variables available in IPUMS CPS but did not find an exact match. The closest match is the variable "UH_WGTS_A1", which has a similar distribution. Since we did not identify an exact match for "fnlwgt" and the variable is not a property of an individual, we do not utilize it further in our experiments.

Varying the income threshold experiments

In our experiments, we randomly split the 49,531 examples in the IPUMS Adult reconstruction into a training set of size 32,094 and a test-set of size 13,755. We vary the threshold from

6,000 to 72,000. Concretely, for a given threshold, e.g. 25,000, the task is to predict whether the individual's income is greater than 25,000. We use a one-hot encoding for the categorical features, and we use the same clustering preprocessing for the Education-Num and Age features as Bellamy et al. [15]. All features are further scaled to be zero-mean and have unit variance.

In our experiments, as the "unconstrained" base classifier, we use the gradient boosted decision tree classifier provided by Pedregosa et al. [98] with exponential loss, num_estimators 5, max_depth 5, and all other hyperparameters set to the default. We found this to slightly outperform the default gradient boosting machine at threshold 50,000. For the three fairness interventions, we used the implementation of LFR [130] provided by Bellamy et al. [15] with hyperparameters Ax 1e-4, Ay 1.0, Az 1000, maxiter 20000, and maxfun 20000, which were chosen by a grid search at threshold 50,000 to maximize the difference between accuracy and the demographic parity disparity. We used the implementation of the reductions approach of Agarwal et al. [3] provided by Bird et al. [17] with the default hyperparameters, and we used implementation of post-processing [51] provided by Bellamy et al. [15].

In Figure 3.1 in the main text, we compare the performance of these three fairness interventions when enforcing demographic parity as the threshold varies. In Figure 3.5, we additionally compare the performance of in-processing method (ExpGrad) and the post-processing method when enforcing equality of opportunity (EO). We exclude LFR from the comparison because this method does not enforce equality of opportunity without additional modification. The results from this experiment are very similar to the experiment enforcing demographic parity. As the threshold varies, the accuracy drop needed to enforce EO varies substantially, as does the trade-off between criteria when enforcing EO. Moreover, for high values of the threshold, the small number of positive instances substantially increases the confidence intervals around the report EO values and makes it difficult to compare the different interventions.

New prediction task details

In this section we detail the target variable, features, and filters that comprise each of our prediction tasks; more information about each feature can be found from the ACS PUMS documentation.[3] For each feature, we list the variable code as provided by the ACS PUMS data sample, its extended description in parentheses, and finally the range of values for the variable.

ACSIncome

Predict whether US working adults' yearly income is above $50,000.

[3]https://www.census.gov/programs-surveys/acs/microdata/documentation.html

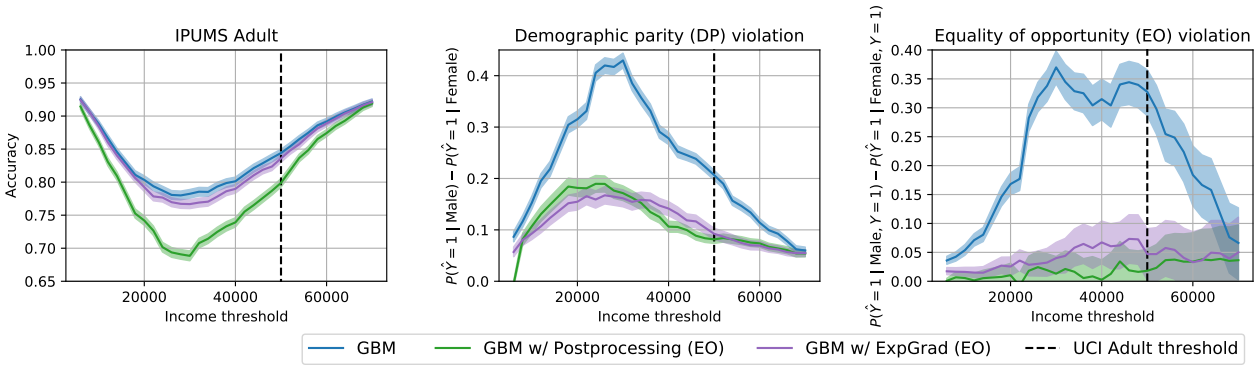


Figure 3.5: Fairness interventions with varying income threshold on IPUMS Adult. Comparison of in-processing and post-processing methods for achieving equality of opportunity (EO). LFR does not target EO, so we exclude it from the comparison. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for equality of opportunity.

Target: PINCP (Total person's income): an individual's label is 1 if PINCP $>$ 50000, otherwise 0. Note that with our software package, this chosen income threshold can be toggled easily to label the ACS PUMS data differently, and construct a new prediction task.

Features:

- AGEP (Age): Range of values:
  - 0 - 99 (integers)
  - 0 indicates less than 1 year old.
- COW (Class of worker): Range of values:
  - N/A (not in universe)
  - 1: Employee of a private for-profit company or business, or of an individual, for wages, salary, or commissions
  - 2: Employee of a private not-for-profit, tax-exempt, or charitable organization
  - 3: Local government employee (city, county, etc.)
  - 4: State government employee
  - 5: Federal government employee
  - 6: Self-employed in own not incorporated business, professional practice, or farm
  - 7: Self-employed in own incorporated business, professional practice or farm
  - 8: Working without pay in family business or farm

  - 9: Unemployed and last worked 5 years ago or earlier or never worked

- SCHL (Educational attainment): Range of values:
  - N/A (less than 3 years old)
  - 1: No schooling completed
  - 2: Nursery school/preschool
  - 3: Kindergarten
  - 4: Grade 1
  - 5: Grade 2
  - 6: Grade 3
  - 7: Grade 4
  - 8: Grade 5
  - 9: Grade 6
  - 10: Grade 7
  - 11: Grade 8
  - 12: Grade 9
  - 13: Grade 10
  - 14: Grade 11
  - 15: 12th Grade - no diploma
  - 16: Regular high school diploma
  - 17: GED or alternative credential
  - 18: Some college but less than 1 year
  - 19: 1 or more years of college credit but no degree
  - 20: Associate's degree
  - 21: Bachelor's degree
  - 22: Master's degree
  - 23: Professional degree beyond a bachelor's degree
  - 24: Doctorate degree

- MAR (Marital status): Range of values:
  - 1: Married
  - 2: Widowed
  - 3: Divorced

  - 4: Separated
  - 5: Never married or under 15 years old

- OCCP (Occupation): Please see ACS PUMS documentation for the full list of occupation codes
- POBP (Place of birth): Range of values includes most countries and individual US states; please see ACS PUMS documentation for the full list.
- RELP (Relationship): Range of values:
  - 0: Reference person
  - 1: Husband/wife
  - 2: Biological son or daughter
  - 3: Adopted son or daughter
  - 4: Stepson or stepdaughter
  - 5: Brother or sister
  - 6: Father or mother
  - 7: Grandchild
  - 8: Parent-in-law
  - 9: Son-in-law or daughter-in-law
  - 10: Other relative
  - 11: Roomer or boarder
  - 12: Housemate or roommate
  - 13: Unmarried partner
  - 14: Foster child
  - 15: Other nonrelative
  - 16: Institutionalized group quarters population
  - 17: Noninstitutionalized group quarters population
- WKHP (Usual hours worked per week past 12 months): Range of values:
  - N/A (less than 16 years old / did not work during the past 12 months)
  - 1 - 98 integer valued: usual hours worked
  - 99: 99 or more usual hours
- SEX (Sex): Range of values:

  - 1: Male
  - 2: Female

- RAC1P (Recoded detailed race code): Range of values:
  - 1: White alone
  - 2: Black or African American alone
  - 3: American Indian alone
  - 4: Alaska Native alone
  - 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
  - 6: Asian alone
  - 7: Native Hawaiian and Other Pacific Islander alone
  - 8: Some Other Race alone
  - 9: Two or More Races

Filters:

- AGEP (Age): Must be greater than 16
- PINCP (Total person's income): Must be greater than 100
- WKHP (Usual hours worked per week past 12 months): Must be greater than 0
- PWGTP (Person weight (relevant for re-weighting dataset to represent the general US population most accurately)): Must be greater than or equal to 1

ACSPublicCoverage

Predict whether a low-income individual, not eligible for Medicare, has coverage from public health insurance.

Target: PUBCOV (Public health coverage): an individual's label is 1 if PUBCOV == 1 (with public health coverage), otherwise 0.

Features:

- AGEP (Age): Range of values:
  - 0 - 99 (integers)
  - 0 indicates less than 1 year old.

- SCHL (Educational attainment): Range of values:
  - N/A (less than 3 years old)
  - 1: No schooling completed
  - 2: Nursery school/preschool
  - 3: Kindergarten
  - 4: Grade 1
  - 5: Grade 2
  - 6: Grade 3
  - 7: Grade 4
  - 8: Grade 5
  - 9: Grade 6
  - 10: Grade 7
  - 11: Grade 8
  - 12: Grade 9
  - 13: Grade 10
  - 14: Grade 11
  - 15: 12th Grade - no diploma
  - 16: Regular high school diploma
  - 17: GED or alternative credential
  - 18: Some college but less than 1 year
  - 19: 1 or more years of college credit but no degree
  - 20: Associate's degree
  - 21: Bachelor's degree
  - 22: Master's degree
  - 23: Professional degree beyond a bachelor's degree
  - 24: Doctorate degree
- MAR (Marital status): Range of values:
  - 1: Married
  - 2: Widowed
  - 3: Divorced
  - 4: Separated

  - 5: Never married or under 15 years old

- SEX (Sex): Range of values:
  - 1: Male
  - 2: Female
- DIS (Disability recode): Range of values:
  - 1: With a disability
  - 2: Without a disability
- ESP (Employment status of parents): Range of values:
  - N/A (not own child of householder, and not child in subfamily)
  - 1: Living with two parents: both parents in labor force
  - 2: Living with two parents: Father only in labor force
  - 3: Living with two parents: Mother only in labor force
  - 4: Living with two parents: Neither parent in labor force
  - 5: Living with father: Father in the labor force
  - 6: Living with father: Father not in labor force
  - 7: Living with mother: Mother in the labor force
  - 8: Living with mother: Mother not in labor force
- CIT (Citizenship status): Range of values:
  - 1: Born in the U.S.
  - 2: Born in Puerto Rico, Guam, the U.S. Virgin Islands, or the Northern Marianas
  - 3: Born abroad of American parent(s)
  - 4: U.S. citizen by naturalization
  - 5: Not a citizen of the U.S.
- MIG (Mobility status (lived here 1 year ago): Range of values:
  - N/A (less than 1 year old)
  - 1: Yes, same house (nonmovers)
  - 2: No, outside US and Puerto Rico
  - 3: No, different house in US or Puerto Rico
- MIL (Military service): Range of values:

  - N/A (less than 17 years old)
  - 1: Now on active duty
  - 2: On active duty in the past, but not now
  - 3: Only on active duty for training in Reserves/National Guard
  - 4: Never served in the military
- ANC (Ancestry recode): Range of values:
  - 1: Single
  - 2: Multiple
  - 3: Unclassified
  - 4: Not reported
  - 8: Suppressed for data year 2018 for select PUMAs
- NATIVITY (Nativity): Range of values:
  - 1: Native
  - 2: Foreign born
- DEAR (Hearing difficulty): Range of values:
  - 1: Yes
  - 2: No
- DEYE (Vision difficulty): Range of values:
  - 1: Yes
  - 2: No
- DREM (Cognitive difficulty): Range of values:
  - N/A (less than 5 years old)
  - 1: Yes
  - 2: No
- PINCP (Total person's income): Range of values:
  - integers between -19997 and 4209995 to indicate income in US dollars
  - loss of $19998 or more is coded as -19998.
  - income of $4209995 or more is coded as 4209995.

- ESR (Employment status recode): Range of values:
  - N/A (less than 16 years old)
  - 1: Civilian employed, at work
  - 2: Civilian employed, with a job but not at work
  - 3: Unemployed
  - 4: Armed forces, at work
  - 5: Armed forces, with a job but not at work
  - 6: Not in labor force
- ST (State code): Please see ACS PUMS documentation for the correspondence between coded values and state name.
- FER (Gave birth to child within the past 12 months): Range of values:
  - N/A (less than 15 years/greater than 50 years/male)
  - 1: Yes
  - 2: No
- RAC1P (Recoded detailed race code): Range of values:
  - 1: White alone
  - 2: Black or African American alone
  - 3: American Indian alone
  - 4: Alaska Native alone
  - 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
  - 6: Asian alone
  - 7: Native Hawaiian and Other Pacific Islander alone
  - 8: Some Other Race alone
  - 9: Two or More Races

Filters:

- AGEP (Age) must be less than 65.
- PINCP (Total person's income) must be less than $30,000.

ACSMobility

Predict whether a young adult moved addresses in the last year.

Target: MIG (Mobility status): an individual's label is 1 if MIG == 1, and 0 otherwise.

Features:

- AGEP (Age): Range of values:
  - 0 - 99 (integers)
  - 0 indicates less than 1 year old.
- SCHL (Educational attainment): Range of values:
  - N/A (less than 3 years old)
  - 1: No schooling completed
  - 2: Nursery school/preschool
  - 3: Kindergarten
  - 4: Grade 1
  - 5: Grade 2
  - 6: Grade 3
  - 7: Grade 4
  - 8: Grade 5
  - 9: Grade 6
  - 10: Grade 7
  - 11: Grade 8
  - 12: Grade 9
  - 13: Grade 10
  - 14: Grade 11
  - 15: 12th Grade - no diploma
  - 16: Regular high school diploma
  - 17: GED or alternative credential
  - 18: Some college but less than 1 year
  - 19: 1 or more years of college credit but no degree
  - 20: Associate's degree

  - 21: Bachelor's degree
  - 22: Master's degree
  - 23: Professional degree beyond a bachelor's degree
  - 24: Doctorate degree
- MAR (Marital status): Range of values:
  - 1: Married
  - 2: Widowed
  - 3: Divorced
  - 4: Separated
  - 5: Never married or under 15 years old
- SEX (Sex): Range of values:
  - 1: Male
  - 2: Female
- DIS (Disability recode): Range of values:
  - 1: With a disability
  - 2: Without a disability
- ESP (Employment status of parents): Range of values:
  - N/A (not own child of householder, and not child in subfamily)
  - 1: Living with two parents: both parents in labor force
  - 2: Living with two parents: Father only in labor force
  - 3: Living with two parents: Mother only in labor force
  - 4: Living with two parents: Neither parent in labor force
  - 5: Living with father: Father in the labor force
  - 6: Living with father: Father not in labor force
  - 7: Living with mother: Mother in the labor force
  - 8: Living with mother: Mother not in labor force
- CIT (Citizenship status): Range of values:
  - 1: Born in the U.S.
  - 2: Born in Puerto Rico, Guam, the U.S. Virgin Islands, or the Northern Marianas

  - 3: Born abroad of American parent(s)
  - 4: U.S. citizen by naturalization
  - 5: Not a citizen of the U.S.
- MIL (Military service): Range of values:
  - N/A (less than 17 years old)
  - 1: Now on active duty
  - 2: On active duty in the past, but not now
  - 3: Only on active duty for training in Reserves/National Guard
  - 4: Never served in the military
- ANC (Ancestry recode): Range of values:
  - 1: Single
  - 2: Multiple
  - 3: Unclassified
  - 4: Not reported
  - 8: Suppressed for data year 2018 for select PUMAs
- NATIVITY (Nativity): Range of values:
  - 1: Native
  - 2: Foreign born
- RELP (Relationship): Range of values:
  - 0: Reference person
  - 1: Husband/wife
  - 2: Biological son or daughter
  - 3: Adopted son or daughter
  - 4: Stepson or stepdaughter
  - 5: Brother or sister
  - 6: Father or mother
  - 7: Grandchild
  - 8: Parent-in-law
  - 9: Son-in-law or daughter-in-law
  - 10: Other relative

  - 11: Roomer or boarder
  - 12: Housemate or roommate
  - 13: Unmarried partner
  - 14: Foster child
  - 15: Other nonrelative
  - 16: Institutionalized group quarters population
  - 17: Noninstitutionalized group quarters population
- DEAR (Hearing difficulty): Range of values:
  - 1: Yes
  - 2: No
- DEYE (Vision difficulty): Range of values:
  - 1: Yes
  - 2: No
- DREM (Cognitive difficulty): Range of values:
  - N/A (less than 5 years old)
  - 1: Yes
  - 2: No
- RAC1P (Recoded detailed race code): Range of values:
  - 1: White alone
  - 2: Black or African American alone
  - 3: American Indian alone
  - 4: Alaska Native alone
  - 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
  - 6: Asian alone
  - 7: Native Hawaiian and Other Pacific Islander alone
  - 8: Some Other Race alone
  - 9: Two or More Races
- GCL (Grandparents living with grandchildren): Range of values:
  - N/A (less than 30 years/institutional GQ)

  - 1: Yes
  - 2: No

- COW (Class of worker): Range of values:
  - N/A (not in universe)
  - 1: Employee of a private for-profit company or business, or of an individual, for wages, salary, or commissions
  - 2: Employee of a private not-for-profit, tax-exempt, or charitable organization
  - 3: Local government employee (city, county, etc.)
  - 4: State government employee
  - 5: Federal government employee
  - 6: Self-employed in own not incorporated business, professional practice, or farm
  - 7: Self-employed in own incorporated business, professional practice or farm
  - 8: Working without pay in family business or farm
  - 9: Unemployed and last worked 5 years ago or earlier or never worked
- ESR (Employment status recode): Range of values:
  - N/A (less than 16 years old)
  - 1: Civilian employed, at work
  - 2: Civilian employed, with a job but not at work
  - 3: Unemployed
  - 4: Armed forces, at work
  - 5: Armed forces, with a job but not at work
  - 6: Not in labor force
- WKHP (Usual hours worked per week past 12 months): Range of values:
  - N/A (less than 16 years old / did not work during the past 12 months)
  - 1 - 98 integer valued: usual hours worked
  - 99: 99 or more usual hours
- JWMNP (Travel time to work): Range of values:
  - N/A (not a worker or a worker that worked at home)
  - integers 1 - 200 for minutes to get to work
  - top-coded at 200 so values above 200 are coded as 200

- PINCP (Total person's income): Range of values:
  - integers between -19997 and 4209995 to indicate income in US dollars
  - loss of $19998 or more is coded as -19998.
  - income of $4209995 or more is coded as 4209995.

Filters:

- AGEP (Age) must be greater than 18 and less than 35.

ACSEmployment

Predict whether an adult is employed.

Target: ESR (Employment status recode): an individual's label is 1 if ESR == 1, and 0 otherwise.

Features:

- AGEP (Age): Range of values:
  - 0 - 99 (integers)
  - 0 indicates less than 1 year old.
- SCHL (Educational attainment): Range of values:
  - N/A (less than 3 years old)
  - 1: No schooling completed
  - 2: Nursery school/preschool
  - 3: Kindergarten
  - 4: Grade 1
  - 5: Grade 2
  - 6: Grade 3
  - 7: Grade 4
  - 8: Grade 5
  - 9: Grade 6
  - 10: Grade 7
  - 11: Grade 8
  - 12: Grade 9

  - 13: Grade 10
  - 14: Grade 11
  - 15: 12th Grade - no diploma
  - 16: Regular high school diploma
  - 17: GED or alternative credential
  - 18: Some college but less than 1 year
  - 19: 1 or more years of college credit but no degree
  - 20: Associate's degree
  - 21: Bachelor's degree
  - 22: Master's degree
  - 23: Professional degree beyond a bachelor's degree
  - 24: Doctorate degree
- MAR (Marital status): Range of values:
  - 1: Married
  - 2: Widowed
  - 3: Divorced
  - 4: Separated
  - 5: Never married or under 15 years old
- SEX (Sex): Range of values:
  - 1: Male
  - 2: Female
- DIS (Disability recode): Range of values:
  - 1: With a disability
  - 2: Without a disability
- ESP (Employment status of parents): Range of values:
  - N/A (not own child of householder, and not child in subfamily)
  - 1: Living with two parents: both parents in labor force
  - 2: Living with two parents: Father only in labor force
  - 3: Living with two parents: Mother only in labor force
  - 4: Living with two parents: Neither parent in labor force

  - 5: Living with father: Father in the labor force
  - 6: Living with father: Father not in labor force
  - 7: Living with mother: Mother in the labor force
  - 8: Living with mother: Mother not in labor force
- MIG (Mobility status (lived here 1 year ago): Range of values:
  - N/A (less than 1 year old)
  - 1: Yes, same house (nonmovers)
  - 2: No, outside US and Puerto Rico
  - 3: No, different house in US or Puerto Rico
- CIT (Citizenship status): Range of values:
  - 1: Born in the U.S.
  - 2: Born in Puerto Rico, Guam, the U.S. Virgin Islands, or the Northern Marianas
  - 3: Born abroad of American parent(s)
  - 4: U.S. citizen by naturalization
  - 5: Not a citizen of the U.S.
- MIL (Military service): Range of values:
  - N/A (less than 17 years old)
  - 1: Now on active duty
  - 2: On active duty in the past, but not now
  - 3: Only on active duty for training in Reserves/National Guard
  - 4: Never served in the military
- ANC (Ancestry recode): Range of values:
  - 1: Single
  - 2: Multiple
  - 3: Unclassified
  - 4: Not reported
  - 8: Suppressed for data year 2018 for select PUMAs
- NATIVITY (Nativity): Range of values:
  - 1: Native

  - 2: Foreign born

- RELP (Relationship): Range of values:
  - 0: Reference person
  - 1: Husband/wife
  - 2: Biological son or daughter
  - 3: Adopted son or daughter
  - 4: Stepson or stepdaughter
  - 5: Brother or sister
  - 6: Father or mother
  - 7: Grandchild
  - 8: Parent-in-law
  - 9: Son-in-law or daughter-in-law
  - 10: Other relative
  - 11: Roomer or boarder
  - 12: Housemate or roommate
  - 13: Unmarried partner
  - 14: Foster child
  - 15: Other nonrelative
  - 16: Institutionalized group quarters population
  - 17: Noninstitutionalized group quarters population
- DEAR (Hearing difficulty): Range of values:
  - 1: Yes
  - 2: No
- DEYE (Vision difficulty): Range of values:
  - 1: Yes
  - 2: No
- DREM (Cognitive difficulty): Range of values:
  - N/A (less than 5 years old)
  - 1: Yes
  - 2: No

- RAC1P (Recoded detailed race code): Range of values:
  - 1: White alone
  - 2: Black or African American alone
  - 3: American Indian alone
  - 4: Alaska Native alone
  - 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
  - 6: Asian alone
  - 7: Native Hawaiian and Other Pacific Islander alone
  - 8: Some Other Race alone
  - 9: Two or More Races
- GCL (Grandparents living with grandchildren): Range of values:
  - N/A (less than 30 years/institutional GQ)
  - 1: Yes
  - 2: No

Filters:

- AGEP (Age) must be greater than 16 and less than 90.
- PWGTP (Person weight) must be greater than or equal to 1.

ACSTravelTime

Predict whether a working adult has a travel time to work of greater than 20 minutes.

Target: JWMNP (Travel time to work): an individual's label is 1 if JWMNP > 20, and 0 otherwise.

Features:

- AGEP (Age): Range of values:
  - 0 - 99 (integers)
  - 0 indicates less than 1 year old.
- SCHL (Educational attainment): Range of values:

  - N/A (less than 3 years old)
  - 1: No schooling completed
  - 2: Nursery school/preschool
  - 3: Kindergarten
  - 4: Grade 1
  - 5: Grade 2
  - 6: Grade 3
  - 7: Grade 4
  - 8: Grade 5
  - 9: Grade 6
  - 10: Grade 7
  - 11: Grade 8
  - 12: Grade 9
  - 13: Grade 10
  - 14: Grade 11
  - 15: 12th Grade - no diploma
  - 16: Regular high school diploma
  - 17: GED or alternative credential
  - 18: Some college but less than 1 year
  - 19: 1 or more years of college credit but no degree
  - 20: Associate's degree
  - 21: Bachelor's degree
  - 22: Master's degree
  - 23: Professional degree beyond a bachelor's degree
  - 24: Doctorate degree

- MAR (Marital status): Range of values:
  - 1: Married
  - 2: Widowed
  - 3: Divorced
  - 4: Separated
  - 5: Never married or under 15 years old

- SEX (Sex): Range of values:
  - 1: Male
  - 2: Female
- DIS (Disability recode): Range of values:
  - 1: With a disability
  - 2: Without a disability
- ESP (Employment status of parents): Range of values:
  - N/A (not own child of householder, and not child in subfamily)
  - 1: Living with two parents: both parents in labor force
  - 2: Living with two parents: Father only in labor force
  - 3: Living with two parents: Mother only in labor force
  - 4: Living with two parents: Neither parent in labor force
  - 5: Living with father: Father in the labor force
  - 6: Living with father: Father not in labor force
  - 7: Living with mother: Mother in the labor force
  - 8: Living with mother: Mother not in labor force
- MIG (Mobility status (lived here 1 year ago): Range of values:
  - N/A (less than 1 year old)
  - 1: Yes, same house (nonmovers)
  - 2: No, outside US and Puerto Rico
  - 3: No, different house in US or Puerto Rico
- RELP (Relationship): Range of values:
  - 0: Reference person
  - 1: Husband/wife
  - 2: Biological son or daughter
  - 3: Adopted son or daughter
  - 4: Stepson or stepdaughter
  - 5: Brother or sister
  - 6: Father or mother
  - 7: Grandchild

  - 8: Parent-in-law
  - 9: Son-in-law or daughter-in-law
  - 10: Other relative
  - 11: Roomer or boarder
  - 12: Housemate or roommate
  - 13: Unmarried partner
  - 14: Foster child
  - 15: Other nonrelative
  - 16: Institutionalized group quarters population
  - 17: Noninstitutionalized group quarters population

- RAC1P (Recoded detailed race code): Range of values:
  - 1: White alone
  - 2: Black or African American alone
  - 3: American Indian alone
  - 4: Alaska Native alone
  - 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
  - 6: Asian alone
  - 7: Native Hawaiian and Other Pacific Islander alone
  - 8: Some Other Race alone
  - 9: Two or More Races

- PUMA (Public use microdata area code (PUMA) based on 2010 Census definition (areas with population of 100,000 or more, use with ST for unique code)): Please see ACS PUMS documentation for details on the PUMA codes (which range from 100 to 70301)

- ST (State code): Please see ACS PUMS documentation for the correspondence between coded values and state name.

- CIT (Citizenship status): Range of values:
  - 1: Born in the U.S.
  - 2: Born in Puerto Rico, Guam, the U.S. Virgin Islands, or the Northern Marianas
  - 3: Born abroad of American parent(s)

  - 4: U.S. citizen by naturalization
  - 5: Not a citizen of the U.S.

- OCCP (Occupation): Please see ACS PUMS documentation for the full list of occupation codes

- JWTR (Means of transportation to work): Range of values:

  - N/A (not a worker–not in the labor force, including persons under 16 years, unemployed, employed, with a job but not at work, Armed Forces, with a job but not at work)
  - 1: Car, truck, or van
  - 2: Bus or trolley bus
  - 3: Streetcar or trolley car (carro publico in Puerto Rico)
  - 4: Subway or elevated
  - 5: Railroad
  - 6: Ferryboat
  - 7: Taxicab
  - 8: Motorcycle
  - 9: Bicycle
  - 10: Walked;
  - 11: Worked at home
  - 12: Other method

- POWPUMA (Place of work PUMA based on 2010 Census definitions): Please see ACS PUMS documentation for details on PUMA codes

- POVPIP (Income-to-poverty ratio recode): Range of values:

  - N/A
  - integers 0-500
  - 501 for 501 percent or more

Filters:

- AGEP (Age) must be greater than 16.

- PWGTP (Person weight) must be greater than or equal to 1.

- ESR (Employment status recode) must be equal to 1 (employed).

Dataset access and license

We provide a flexible software package to download ACS PUMS data and construct both the new prediction tasks discussed in Section 3.3, as well as new tasks using ACS PUMS data products. The ACS PUMS data itself is governed by the terms of service from the US Census Bureau. For more information, see `https://www.census.gov/data/developers/about/terms-of-service.html` Similarly, the IPUMS adult reconstruction is governed by the IPUMS terms of use. For more information, see `https://ipums.org/about/terms`.

Table 3.1 experiment details

For each of the tasks listed in Table 3.1 (ACSIncome, ACSPublicCoverage, ACSMobility, ACSEmployment, ACSTravelTime), we use the 1-year 2018 US-Wide ACS PUMS data. We use a maximum of 100,000 examples from each state, and randomly subsample states that have more than 100,000 examples. We randomly split 80% of the dataset into a training split and the remaining 20% into a test split. All features are standardized to be zero-mean and unit-variance. `C`onstant Predictor refers to the majority class baseline, `LogReg` refers to a logistic regression baseline, and `GBM` refers to a gradient boosted decision tree classifier. For each models, we use the implementation provided by Pedregosa et al. [98] with the default hyperparameters.

Tour of empirical observations: missing experimental details

Models and hyperparameters. All of the experiments in this section use the same unconstrained base model: a gradient boosted decision tree (GBM). We chose this model because it trains quickly and consistently achieved higher accuracy than other baseline models we considered (logistic regression and random forests) in the unconstrained setting; experiments using other base models also produced qualitatively similar results, so we focus on GBM in this paper. We use the implementation provided by Pedregosa et al. [98] and use `e`xponential loss, `num_`estimators 5, `max_`depth 5, and all other hyperparameters set to the default. These hyperparameters were chosen via a small grid search to maximize accuracy on the ACSIncome task. We use the implementation of LFR [130] from Bellamy et al. [15] with hyperparameters `k`=10, `Ax`=0.1, `Ay`=1.0, `Az` = 2.0, `maxiter`=5000, and `maxfun`=5000. The hyperparameters are the same as those used in the UCI Adult tutorial provided by Bellamy et al. [15]. For the in-processing method (ExpGrad) from Agarwal et al. [3], we use the implementation from Bird et al. [17] with the default hyperparameters, and for the post-processing method, we use the threshold adjustment method of Hardt et al. [51], which is also implemented in Bellamy et al. [15]. In Section 3.4, we use all of the methods to enforce demographic parity. We detail additional experiments enforcing equality of opportunity in Appendix 3.6.

Datasets. Throughout this section, we use the ACSIncome task described in Section 3.3 and Appendix 3.6. With the exception of the distribution shift across time experiments, we use the 2018 1-Year ACS PUMS data. For each state, we randomly split 80% of the dataset into a training split and use the remaining 20% as a test split. The US-Wide dataset is constructed by combining these training and testing sets over all 50 states and Puerto Rico. For the distribution shift across time experiments, we use the same procedure for the 2014-2017 1-Year ACS PUMS data.

Confidence intervals. To account for random variation in estimating model accuracies and violations of demographic parity and equality of opportunity, we report each of these metrics with appropriate confidence intervals. We report and plot accuracy numbers with 95% Clopper-Pearson intervals. We report and plot violations of demographic parity and equality of opportunity with 95% Newcombe intervals for the difference between two binomial proportions.

Compute environment. All of our experiments are run on CPUs on a cluster computer with 24 Intel Xeon E7 CPUs and 300 GB of RAM.

Additional experiments

In this section, we conduct the same set of experiments conducted in Section 3.4 on the 5 other prediction tasks we introduced in Section 3.3. Throughout we keep the experimental details (models, hyperparameters, etc) identical to those detailed in Appendix 3.6.

Intervention effect sizes across states

As in Section 3.4, we train an unconstrained gradient boosted decision tree (GBM) on each state, and we compare the accuracy and fairness criterion violation of this unconstrained model with the same model after applying one of three common fairness intervention: pre-processing (LFR), the in-processing fair reductions methods from Agarwal et al. [3] (ExpGrad), and the simple post-processing method that adjusts group-based acceptance thresholds to satisfy a constraint [51]. Figure 3.6 shows the result of this experiment for the ACSIncome prediction task for interventions to achieve equality of opportunity.

In Figure 3.7, we conduct the same experiment for demographic parity on four other ACS data tasks: ACSPublicCoverage, ACSEmployment, ACSMobility, and ACSTravelTime, respectively.

Geographic distribution shift

In Figure 3.8, we plot accuracy and equality of opportunity violation with respect to race for both an unconstrained GBM and the same model after applying a post-processing adjustment to achieve equality of opportunity on a natural suite of test sets: the in-distribution (same

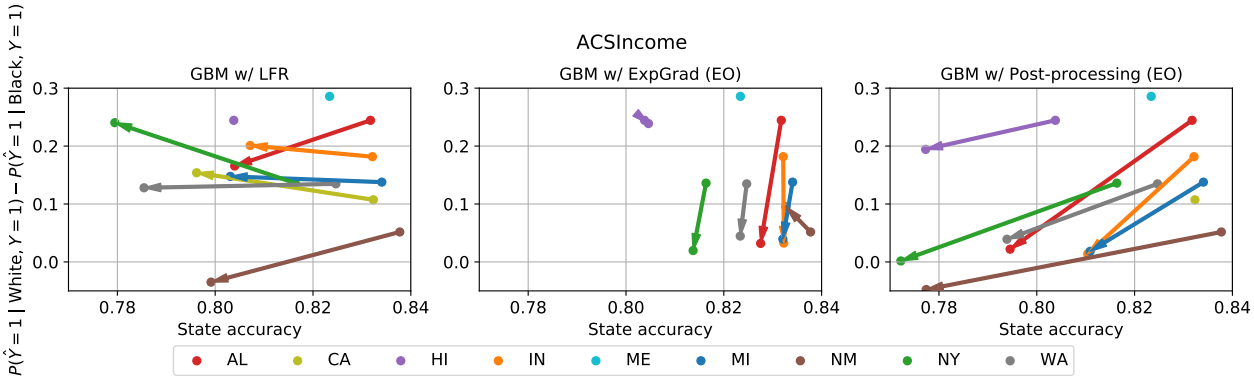


Figure 3.6: The effect size of fairness interventions varies by state. Each panel shows the change in accuracy and equality of opportunity violation (EO) on the ACSIncome task after applying a fairness intervention to an unconstrained gradient boosted decision tree (GBM). Each arrow corresponds to a different state distribution. The arrow base represents the (accuracy, EO) point corresponding to the unconstrained GBM, and the head represents the (accuracy, EO) point obtained after applying the intervention. The arrow for HI in the LFR plot and ME in all three plots is entirely covered by the start and end points.

state test set) and the out-of-distribution test sets for the 49 other states. This is the same experiment as in Section 3.4, but with equality of opportunity rather than demographic parity as the metric of interest. In Figures 3.9, 3.10 3.11, and 3.12 we conduct the same experiment for demographic parity on four other ACS data tasks: ACSPublicCoverage, ACSEmployment, ACSMobility, and ACSTravelTime, respectively.

Temporal distribution shift

In Figure 3.13, we plot model accuracy and equality of opportunity violation for a GBM trained on the ACSIncome task using US-wide data from 2014 and evaluated on the test sets for the same task drawn from years 2014-2018. This is the same experiment as conducted in Section 3.4; however, here we consider interventions to satisfy equality of opportunity rather than demographic parity. In Figure 3.14, we conduct repeat this experiment for interventions to satisfy demographic parity on 4 other ACS PUMS predictions tasks: ACSPublicCoverage, ACSMobility, ACSEmployment, and ACSTravelTime.

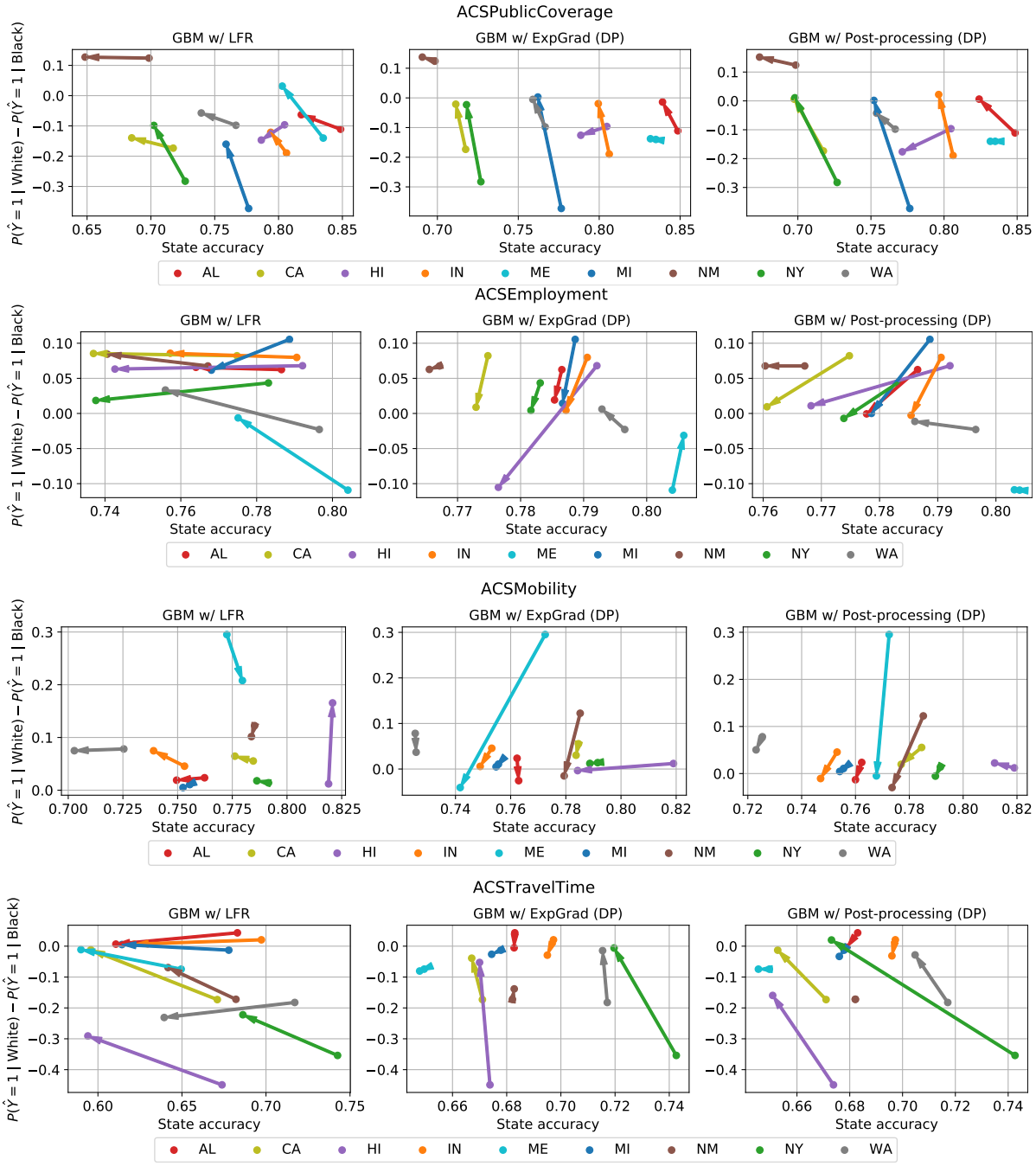


Figure 3.7: The effect size of fairness interventions varies by state. Each panel shows the change in accuracy and demographic parity violation (DP) on the ACSIncome task after applying a fairness intervention to an unconstrained gradient boosted decision tree (GBM). Each arrow corresponds to a different state distribution. The arrow base represents the (accuracy, DP) point corresponding to the unconstrained GBM, and the head represents the (accuracy, DP) point obtained after applying the intervention. When only a single point is visible, the entire arrow is covered by the point, representing an intervention that has essentially no effect.

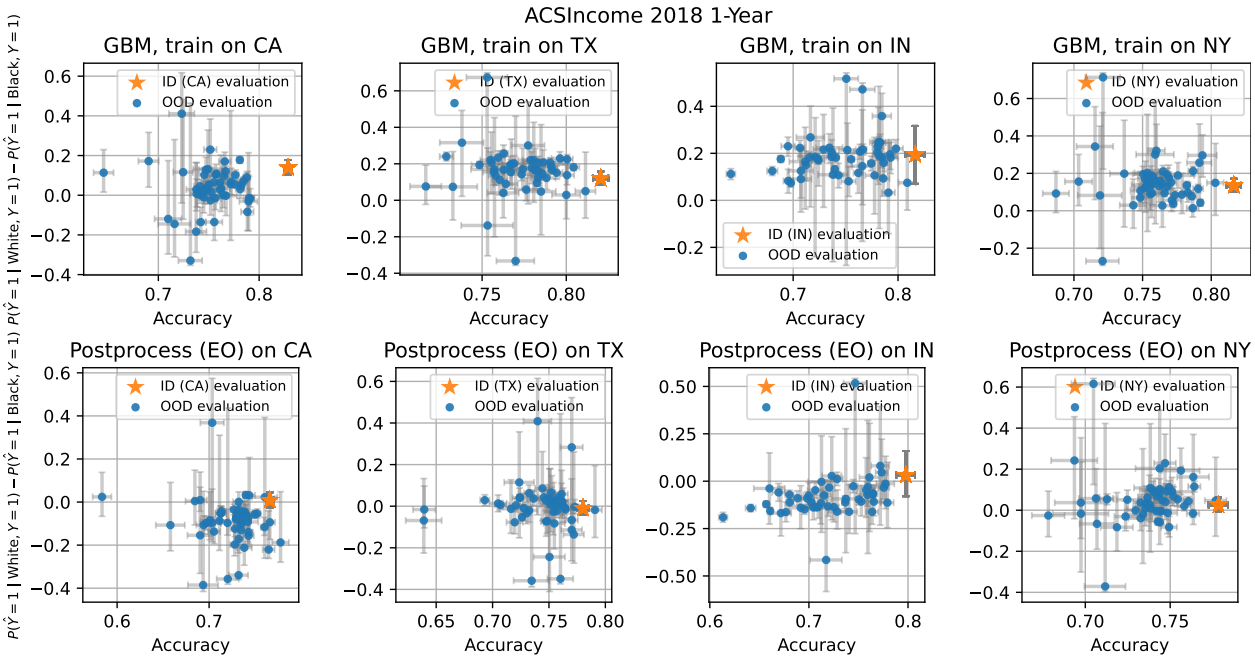


Figure 3.8: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSIncome task and evaluated both in-distribution (ID) on the same state and out-of-distribution (OOD) on the 49 other states in terms of accuracy and equality of opportunity violation. **Bottom:** Each panel shows an GBM with post-processing to enforce equality of opportunity on the state on which it was trained and evaluated both ID and OOD on all 50 states. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for equality of opportunity violation.

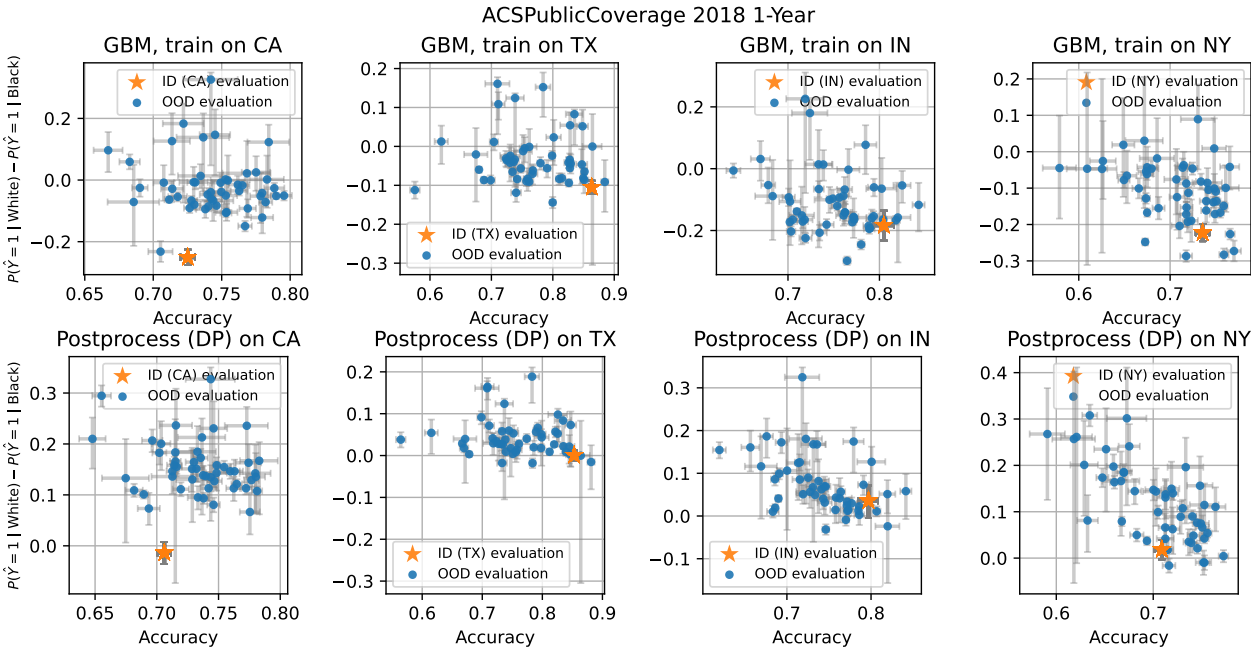


Figure 3.9: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSPublicCoverage task and evaluated both in-distribution (ID) on the same state and out-of-distribution (OOD) on the 49 other states in terms of accuracy and demographic parity violation. **Bottom:** Each panel shows an GBM with post-processing to enforce demographic parity on the state on which it was trained and evaluated both ID and OOD on all 50 states. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for demographic parity.

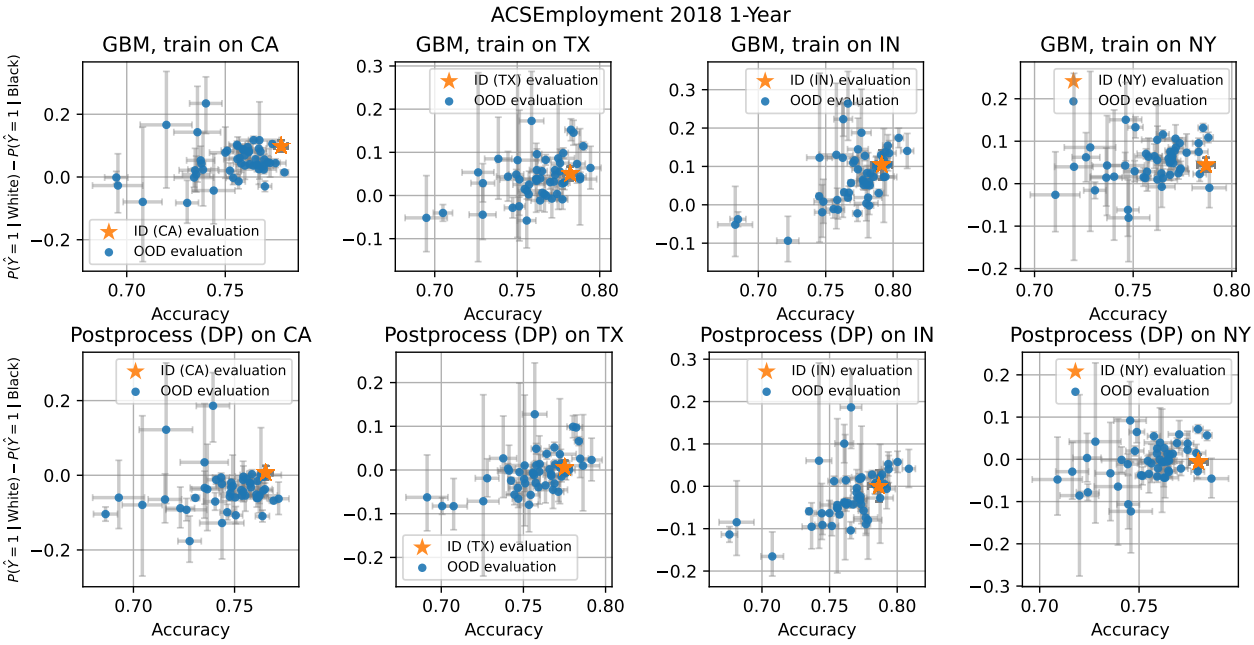


Figure 3.10: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSEmployment task and evaluated both in-distribution (ID) on the same state and out-of-distribution (OOD) on the 49 other states in terms of accuracy and demographic parity violation. **Bottom:** Each panel shows an GBM with post-processing to enforce demographic parity on the state on which it was trained and evaluated both ID and OOD on all 50 states. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for demographic parity.

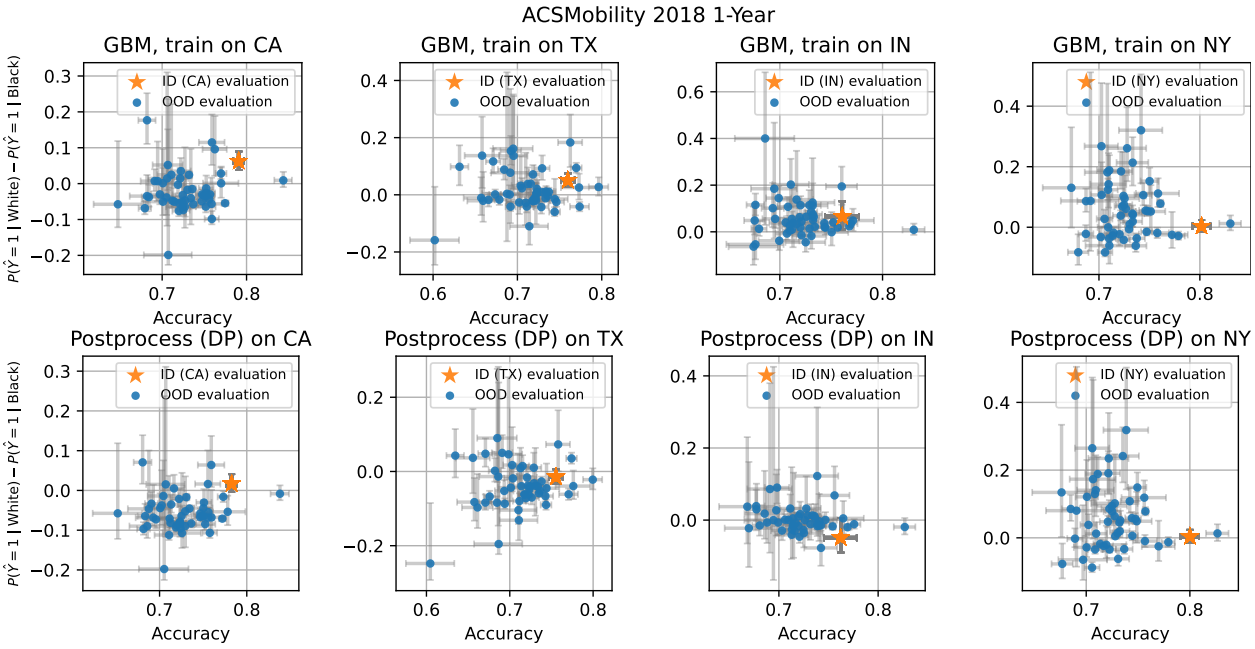


Figure 3.11: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSMobility task and evaluated both in-distribution (ID) on the same state and out-of-distribution (OOD) on the 49 other states in terms of accuracy and equality of opportunity violation. **Bottom:** Each panel shows an GBM with post-processing to enforce equality of opportunity on the state on which it was trained and evaluated both ID and OOD on all 50 states. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for demographic parity.

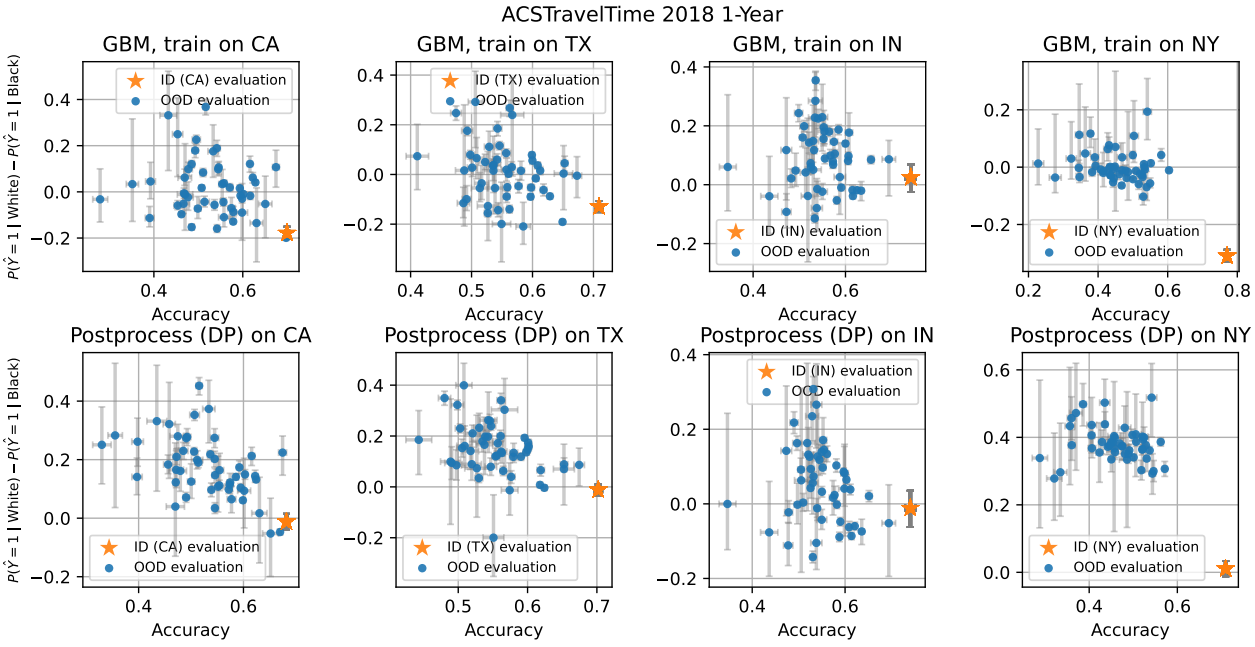


Figure 3.12: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSTravelTime task and evaluated both in-distribution (ID) on the same state and out-of-distribution (OOD) on the 49 other states in terms of accuracy and equality of opportunity violation. **Bottom:** Each panel shows an GBM with post-processing to enforce equality of opportunity on the state on which it was trained and evaluated both ID and OOD on all 50 states. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for demographic parity.

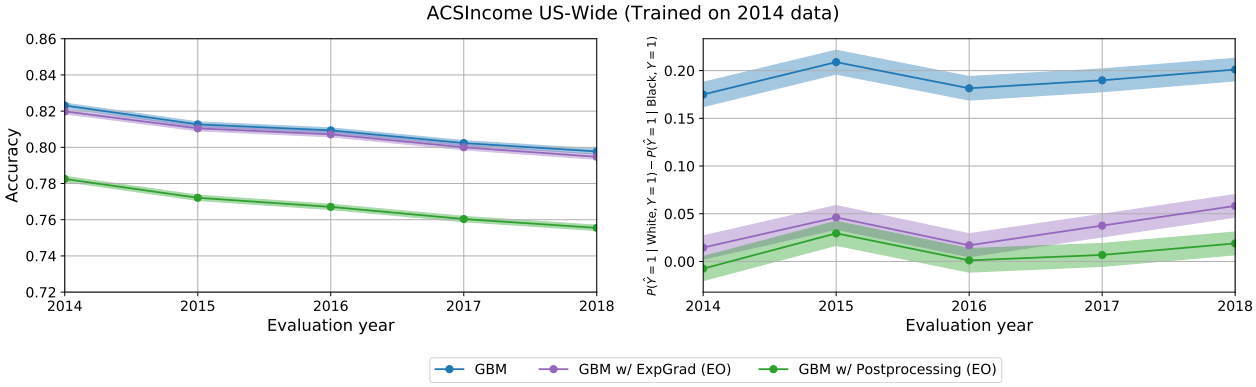


Figure 3.13: Fairness criteria are more stable over time than accuracy. **Left:** Models trained in 2014 on US-wide ACSIncome with and without fairness interventions to achieve equality of opportunity and evaluated on data in subsequent years. **Right:** Violations of equality of opportunity for the same collection of models. Although accuracy drops over time for most problems, violations of equality of opportunity remain essentially constant. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for equality of opportunity violations.

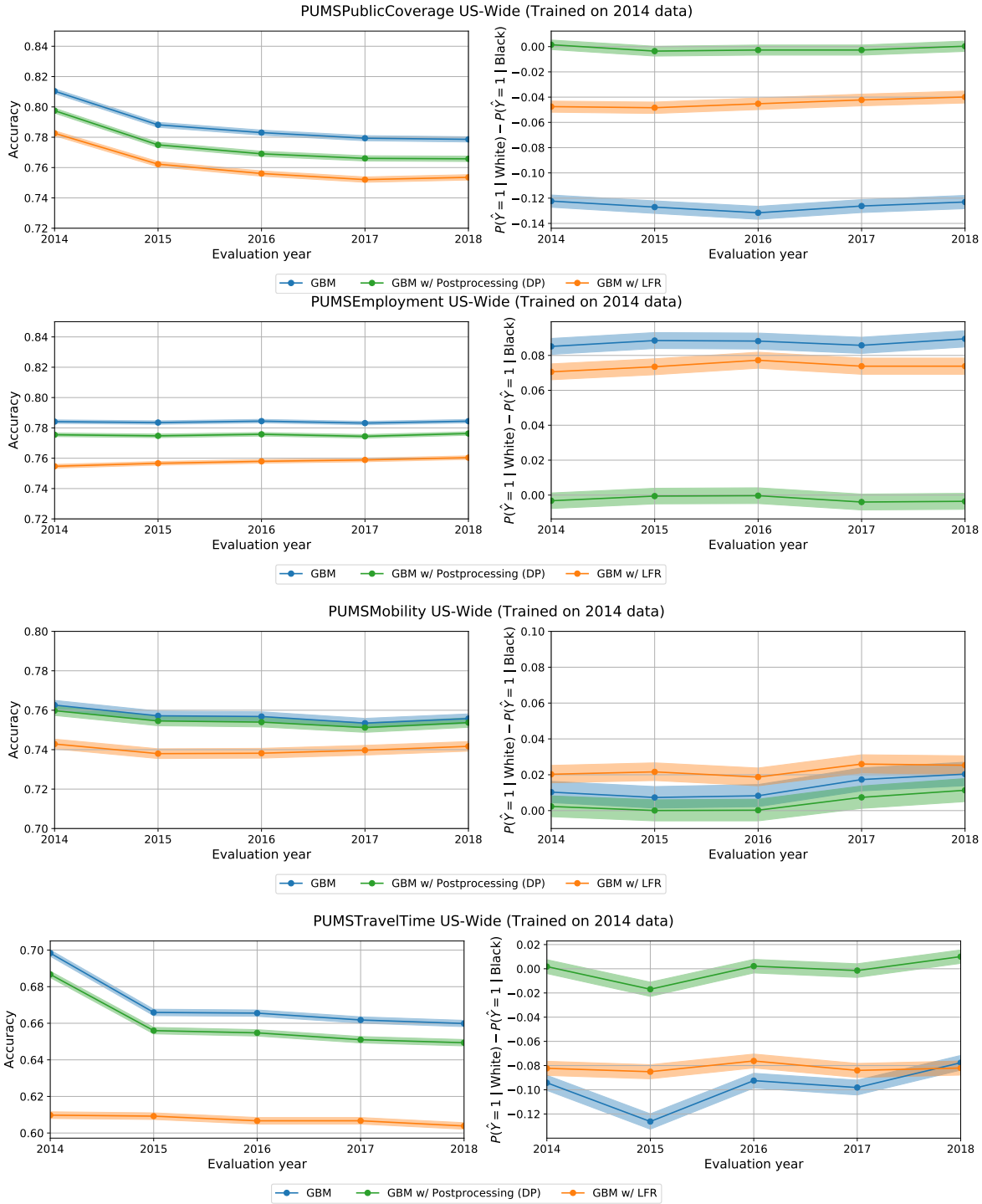


Figure 3.14: Fairness criteria are more stable over time than accuracy. **Left:** Models trained in 2014 on US-wide ACS data with and without fairness interventions to achieve demographic parity and evaluated on data in subsequent years. **Right:** Violations of demographic parity for the same collection of models. Although accuracy drops over time for most problems, violations of demographic parity remain essentially constant. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for demographic parity.

3.7 Datasheet

This datasheet covers both the prediction tasks we introduce and the underlying US Census data sources. However, due to the extensive documentation available about the US Census data we often point to relevant available resources rather than recreating them here. For the most up-to-date version of this datasheet, please refer to `https://github.com/zykls/folktables/blob/main/datasheet.md`.

Motivation

- **For what purpose was the dataset created?** Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

  The motivation for creating prediction tasks on top of US Census data was to extend the dataset ecosystem available for algorithmic fairness research as outlined in this paper.

- **Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?**

  The new prediction tasks were created from available US Census data sources by Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt.

- **Who funded the creation of the dataset?** If there is an associated grant, please provide the name of the grantor and the grant name and number.

  Frances Ding, Moritz Hardt, and John Miller were employed by the University of California for the duration of this research project, funded by grants administered through the University of California. Ludwig Schmidt was employed by Toyota Research throughout this research project.

- **Any other comments?**

  No.

Composition

- **What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?** Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.

  Each instance in our IPUMS Adult reconstruction represents an individual. Similarly, our datasets derived from ACS contains instances representing individuals. The ACS data our datasets are derived from also contain household-level information and the relationship between households and individuals.

- **How many instances are there in total (of each type, if appropriate)?**

  Our IPUMS Adult reconstruction contains 49,531 rows (see Section 3.2). Table 3.1 contains the sizes of our datasets derived from ACS.

- **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable)

  Both IPUMS Adult and our ACS datasets are samples of the US population. Please see Sections 3.2 & 3.3 and the corresponding documentation provided by the US Census Bureau. Note that the per-instance weights have to be taken into account if the sample is meant to represent the US population.

- **What data does each instance consist of?** "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

  Each instance consists of features. IPUMS Adult uses the same features as the original UCI Adult dataset. Appendix 3.6 describes each feature in our new datasets derived from ACS.

- **Is there a label or target associated with each instance?** If so, please provide a description.

  Similar to UCI Adult, our IPUMS Adult reconstruction uses the income as label (where the continuous values as opposed to only the binarized values are now available). Appendix 3.6 describes the labels in our new datasets derived from ACS.

- **Is any information missing from individual instances?** If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

  Some features (e.g., the country of origin in IPUMS Adult) contain missing values. We again refer to the respective documentation from the US Census Bureau for details.

- **Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?** If so, please describe how these relationships are made explicit.

  Our versions of the datasets contain no relationships between individuals. The original data sources from the US Census contain relationships between individuals and households.

- **Are there recommended data splits (e.g., training, development/validation, testing)?** If so, please provide a description of these splits, explaining the rationale behind them.

  For IPUMS Adult, it is possible to follow the same train / test split as the original UCI Adult. In general, we recommend k-fold cross-validation for all of our datasets.

- **Are there any errors, sources of noise, or redundancies in the dataset?** If so, please provide a description.

  Our IPUMS Adult reconstruction contains slightly more rows than the original UCI Adult, see Section 3.2. Beyond IPUMS Adult, we refer to the documentation of CPS and ACS provided by the US Census Bureau.

- **Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?** If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

  Due to restrictions on the re-distribution of the original IPUMS and ACS data sources, we do not provide our datasets as standalone data files. Instead, we provide scripts to generate our datasets from the respective sources.

  Both the US Census Bureau and IPUMS aim to provide stable long-term access to their data. Hence we consider these data sources to be reliable. We refer to the IPUMS website and the website of the US Census Bureau for specific usage restrictions. Neither data source has fees associated with it.

- **Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor patient confidentiality, data that includes the content of individuals' non-public communications)?** If so, please provide a description.

  Our datasets are subsets of datasets released publicly by the US Census Bureau.

- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** If so, please describe why.

  No.

- **Does the dataset relate to people?** If not, you may skip the remaining questions in this section.

  Yes, each instance in our datasets corresponds to a person.

- **Does the dataset identify any subpopulations (e.g., by age, gender)?** If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.

  Our datasets identify subpopulations since each individual has features such as age, gender, or race. Please see the main text of our paper for experiments exploring the respective distributions.

- **Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?** If so, please describe how.

  To the best of our knowledge, it is not possible to identify individuals *directly* from our datasets. However, the possibility of reconstruction attacks combining data from the US Cenus Bureau (such as CPS and ACS) and other data sources are a concern and actively investigated by the research community.

- **Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?** If so, please provide a description.

  Our datasets contain features such as race, age, or gender that are often considered sensitive. This is by design since we assembled our datasets to test algorithmic fairness interventions.

- **Any other comments?**

  No.

Collection process

- **How was the data associated with each instance acquired?** Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

  The data was reported by subjects as part of the ACS and CPS surveys. The respective documentation provided by the US Census Bureau contains further information, see `https://www.census.gov/programs-surveys/acs/methodology/design-and-methodology.html` and also `https://www.census.gov/programs-surveys/cps/technical-documentation/methodology.html`.

- **What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?** How were these mechanisms or procedures validated?

The ACS relies on a combination of internet, mail, telephone, and in-person interviews. CPS uses in-person and telephone interviews. Please see the aforementioned documentation from the US Census Bureau for detailed information.

- **If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?**

  For the ACS, the US Census Bureau sampled housing units uniformly for each county. See ACS docs, Chapter 4 (`https://www2.census.gov/programs-surveys/acs/methodology/design_and_methodology/acs_design_methodology_report_2014.pdf`) for details.

  CPS is also sampled by housing unit from certain sampling areas, see Chapters 3 and 4 in `https://www.census.gov/prod/2006pubs/tp-66.pdf`.

- **Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?**

  The US Census Bureau employs interviewers for conducting surveys. According to online job information platforms such as `indeed.com`, an interviewer earns about $15 per hour.

- **Over what timeframe was the data collected?** Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.

  Both CPS and ACS collect data annually. Our IPUMS Adult reconstruction contains data from the 1994 CPS ASEC. Our new tasks derived from ACS can be instantiated for various survey years.

- **Were any ethical review processes conducted (e.g., by an institutional review board)?** If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

  Both ACS and CPS are regularly reviewed by the US Census Bureau. As a government agency, the US Census Bureau is also subject to government oversight mechanisms.

- **Does the dataset relate to people? If not, you may skip the remainder of the questions in this section.**

  Yes.

- **Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?**

  Data collection was performed by the US Census Bureau. We obtained the data from publicly available US Census repositories.

- **Were the individuals in question notified about the data collection?** If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

  Yes. A sample ACS form is available online: `https://www.census.gov/programs-surveys/acs/about/forms-and-instructions/2021-form.html`

  Information about the CPS collection methodology is available here: `https://www.census.gov/programs-surveys/cps/technical-documentation/methodology.html`

- **Did the individuals in question consent to the collection and use of their data?** If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

  Participation in the US Census American Community Survey is mandatory. Participation in the US Corrent Population Survey is voluntary and consent is obtained at the beginning of the interview: `https://www2.census.gov/programs-surveys/cps/methodology/CPS-Tech-Paper-77.pdf`

- **If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?** If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

  We are not aware that the Census Bureau would provide such a mechanism.

- **Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?** If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

  The US Census Bureau assesses privacy risks and invests in statistical disclosure control. See `https://www.census.gov/topics/research/disclosure-avoidance.html`. Our derived prediction tasks do not increase privacy risks.

- **Any other comments?**

  No.

Preprocessing / cleaning / labeling

- **Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** If so, please provide a description. If not, you may skip the remainder of the questions in this section.

We used two US Census data products – 1) we reconstructed UCI Adult from the Annual Social and Economic Supplement (ASEC) of the Current Population Survey (CPS), and 2) we constructed new prediction tasks from the American Community Survey (ACS) Public Use Microdata Sample (PUMS). Before releasing CPS data publicly, the Census Bureau top-codes certain variables and conducts imputation of certain missing values, as documented here: `https://www.census.gov/programs-surveys/cps/technical-documentation/methodology.html`. In our IPUMS Adult reconstruction, we include a subset of the variables available from the CPS data and do not alter their values.

The ACS data release similarly top-codes certain variables and conducts imputation of certain missing values, as documented here: `https://www.census.gov/programs-surveys/acs/microdata/documentation.html`. For the new prediction tasks that we define, we further process the ACS data as documented at the folktables GitHub page, `https://github.com/zykls/folktables`. In most cases, this involves mapping missing values (NaNs) to -1. We release code so that new prediction tasks may be defined on the ACS data, with potentially different preprocessing. Each prediction task also defines a binary label by discretizing the target variable into two classes; this can be easily changed to define a new labeling in a new prediction task.

- **Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** If so, please provide a link or other access point to the "raw" data.

  Yes, our package provides access to the data as released by the U.S. Census Bureau. The "raw" survey answers collected by the Census Bureau are not available for public release due to privacy considerations.

- **Is the software used to preprocess/clean/label the instances available?** If so, please provide a link or other access point.

  The software to is available at the folktables GitHub page, `https://github.com/zykls/folktables`.

- **Any other comments?**

  No.

Uses

- **Has the dataset been used for any tasks already?** If so, please provide a description.

  In this paper we create five new prediction tasks from the ACS PUMS data:

  1. ACSIncome: Predict whether US working adults' yearly income is above $50,000.
  2. ACSPublicCoverage: Predict whether a low-income individual, not eligible for Medicare, has coverage from public health insurance.

3. ACSMobility: Predict whether a young adult moved addresses in the last year.
4. ACSEmployment: Predict whether a US adult is employed.
5. ACSTravelTime: Predict whether a working adult has a travel time to work of greater than 20 minutes.

Further details about these tasks can be found at the folktables GitHub page, `https://github.com/zykls/folktables`, and in Appendix 3.6.

- **Is there a repository that links to any or all papers or systems that use the dataset?** If so, please provide a link or other access point.

  At the folktables GitHub page, `https://github.com/zykls/folktables`, any public forks to the package are visible, and papers or systems that use the datasets should cite the paper linked at that Github page.

- **What (other) tasks could the dataset be used for?**

  New prediction tasks may be defined on the ACS PUMS data that use different subsets of variables as features and/or different target variables. Different prediction tasks may have different properties such as Bayes error rate, or the base rate disparities between subgroups, that can help to benchmark machine learning models in diverse settings.

- **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?

  Both the CPS and ACS are collected through surveys of a subset of the US population, and in their documentation, they acknowledge that statistical trends in individual states may be noisy compared to those found by analyzing US data as a whole, due to small sample sizes in certain states. In particular, there may be very few individuals with particular characteristics (e.g. ethnicity) in certain states, and generalizing conclusions from these few individuals may be highly inaccurate. Further, benchmarking fair machine learning algorithms on datasets with few representatives of certain subgroups may provide the illusion of "checking a box" for fairness, without substantive merit.

- **Are there tasks for which the dataset should not be used?** If so, please provide a description.

  This dataset contains personal information, and users should not attempt to re-identify individuals in it. Further, these datasets are meant primarily to aid in benchmarking machine learning algorithms; Census data is often crucial for substantive, domain-specific work by social scientists, but our dataset contributions are not in this direction. Substantive

investigations into inequality, demographic shifts, and other important questions should not be based purely on the datasets we provide.

- **Any other comments?**

  No.

Distribution

- **Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** If so, please provide a description.

  The dataset will be available for public download on the folktables GitHub page, `https://github.com/zykls/folktables`.

- **How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?** Does the dataset have a digital object identifier (DOI)?

  The dataset will be be distributed via GitHub, see `https://github.com/zykls/folktables`. The dataset does not have a DOI.

- **When will the dataset be distributed?**

  The dataset will be released on August 1, 2021 and available thereafter for download and public use.

- **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

  The folktables package and data loading code will be available under the MIT license. The folktables data itself is based on data from the American Community Survey (ACS) Public Use Microdata Sample (PUMS) files managed by the US Census Bureau, and it is governed by the terms of use provided by the Census Bureau. For more information, see `https://www.census.gov/data/developers/about/terms-of-service.html`

  Similarly, the IPUMS adult reconstruction is governed by the IPUMS terms of use. For more information, see `https://ipums.org/about/terms`.

- **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.

  The folktables data and the adult reconstruction data are governed by third-party terms of use provided by the US Census Bureau and IPUMS, respectively. See `https://www.census.`

gov/data/developers/about/terms-of-service.html and https://ipums.org/about/terms for complete details. The IPUMS Adult Reconstruction is a subsample of the IPUMS CPS data available from cps.ipums.org These data are intended for replication purposes only. Individuals analyzing the data for other purposes must submit a separate data extract request directly via IPUMS CPS. Individuals should contact ipums@umn.edu for redistribution requests.

- **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.

  To our knowledge, no export controls or regulatory restrictions apply to the dataset.

- **Any other comments?**

  No.

Maintenance

- **Who is supporting/hosting/maintaining the dataset?**

  The dataset will be hosted on GitHub, and supported and maintained by the folktables team. As of June 2021, this team consists of Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt.

- **How can the owner/curator/manager of the dataset be contacted (e.g., email address)?**

  Please send issues and requests to folktables@gmail.com.

- **Is there an erratum?** If so, please provide a link or other access point.

  An erratum will be hosted on the dataset website, https://github.com/zykls/folktables.

- **Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?** If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

  The dataset will be updated as required to address errors and refine the prediction problems based on feedback from the community. The package maintainers will update the dataset and communicate these updates on GitHub.

- **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?** If so, please describe these limits and explain how they will be enforced.

  The data used in folktables is based on data from the American Community Survey (ACS) Public Use Microdata Sample (PUMS) files managed by the US Census Bureau. The

data inherits and will respect the corresponding retention policies of the ACS. Please see `https://www.census.gov/programs-surveys/acs/about.html` for more details. For the Adult reconstruction dataset, the data is based on Current Population Survey (CPS) released by IPUMS and thus inherits and will respect the corresponding retention policies for the CPS. Please see `https://cps.ipums.org/cps/` for more details.

- **Will older versions of the dataset continue to be supported/hosted/maintained?** If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

  Older versions of the datasets in folktables will be clearly indicated, supported, and maintained on the GitHub website. Each new version of the dataset will be tagged with version metadata and an associated GitHub release.

- **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

  Users wishing to contribute to folktables datasets are encouraged to do so by submitting a pull request on the website `https://github.com/zykls/folktables/pulls`. The contributions will be reviewed by the maintainers. These contributions will be reflected in new version of the dataset and broadcasted as part of each Github release.

- **Any other comments?**

  No.

Chapter 4

Identifying Biases in Protein Language Models

4.1 Introduction

Proteins are the building blocks and workhorses of life, performing essential roles in human and ecosystem health. Inspired by natural language processing, many protein language models (PLMs) have been trained to model the distribution of naturally occurring protein sequences [5, 36, 79, 80, 106]. PLMs have been successfully used to predict protein 3D structure [79], catalytic activity [39], and other biophysical properties [20, 57], generally with supervision for fine-tuning. Excitingly, without needing additional supervision, *likelihoods* from PLMs have been shown to correlate with protein fitness, i.e. desirable qualities such as catalytic activity, stability, and binding affinity [85, 91, 92].

Because of this correlation with fitness, PLM likelihoods are increasingly used in protein design. They have been used to screen for potentially beneficial mutations [59], to design libraries of protein candidates with higher hit rates than previously state-of-the-art synthetic libraries [113], and to efficiently evolve human antibodies without any additional supervision [55].

In this work we find that likelihoods from popular PLMs have a species bias: likelihoods of naturally occurring protein sequences are systematically higher in certain species, which can be detrimental for some protein design applications. We describe the extent of this species bias, show that it arises from imbalanced representation of different evolutionary taxa in training data, measure the impact of the bias on protein design, and develop a post-hoc bias mitigation strategy that reduces these impacts.

To our knowledge, this work is the first to identify subgroups that protein language models are systematically biased against, even though these subgroups are present in training data. Similarly to how new methods were developed to identify and quantify biases in language and vision models [18, 111, 118], our contributions include a generalizable framework for assessing bias in protein language models through Elo rating comparisons between subgroups.

In Section 4.3 we identify and quantify species bias, showing that across the many different proteins we study, certain species almost always have higher PLM likelihoods for their protein sequences than other species. For example, in the data we collect, fruit fly proteins have higher likelihoods than the *C. elegans* (roundworm) versions of the same proteins 92% of the time, even though there is no biological reason for fruit fly proteins to be uniformly "fitter" or more canonical. We find consistent species bias in the commonly used Progen2 and ESM2 model families, across several model sizes.

Next, in Section 4.4 we show that the bias can be largely explained by representation of different branches of the evolutionary tree in protein databases. Understanding the bias requires analyzing the hierarchical organization of the tree of life, rather than considering each species as independent of other species. We find that each individual species' representation in these sequence databases only has a 0.2–0.25 Spearman correlation with the per-species bias, while the sample count incorporating *evolutionarily close* species achieves a 0.6–0.75 Spearman correlation.

In Section 4.5 we examine the implications for protein design. The bias causes protein designs to gravitate towards sequences from arbitrarily favored species, which can lead to worse outcomes. For example, proteins from heat-tolerant microbes are indispensable tools for research and industrial applications because of their stability at high temperatures. However, they are under-represented in sequence databases. Using them as starting points for protein design guided by PLM likelihoods, we find that a majority of designs lose thermostability. Similarly, proteins from salt-tolerant microbes tend to lose their salt tolerance after PLM-guided design.

Finally, in Section 4.6 we mitigate the bias by training an auxiliary model to correct sequences' PLM likelihoods. This likelihood correction strategy reduces the loss in thermostability and salt tolerance in protein designs, and provides a proof of concept for many possible avenues of bias mitigation.

Overall, we find biases of unexpectedly large magnitude in PLMs, with detrimental impacts on PLM-guided design. Looking forwards, these results suggest that protein designers should use PLM likelihoods carefully and consider whether the species bias should be corrected for a given application. We additionally argue that in the long-term, the protein design field would benefit from more deliberate curation of training data, potentially tailored to different contexts.

4.2 Related work

Mismatches between pre-training and downstream tasks Self-supervised pre-training produces strong results in both natural language processing (NLP) and protein modeling. However, recent work has shown that standard pre-training objectives can imbue language models with properties that are undesirable for downstream tasks, such as lower accuracy when correct outputs contain infrequent words [84], self-delusions [95], and more [78]. In this

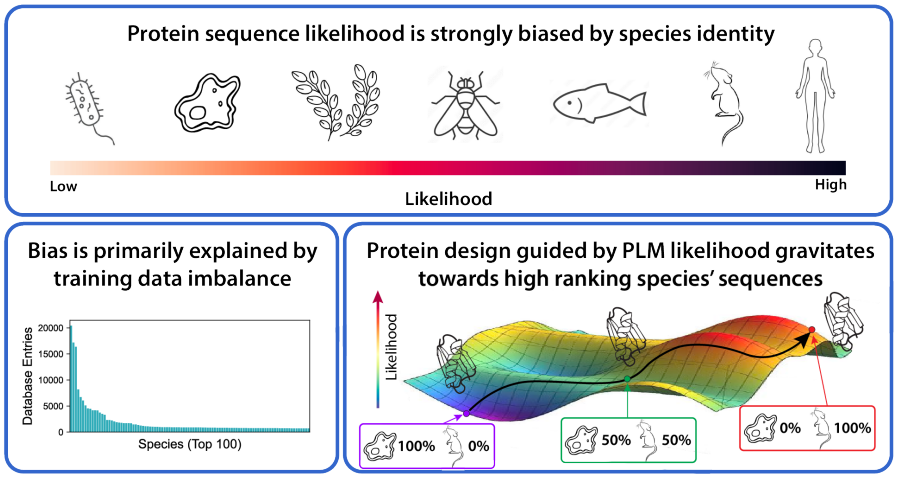


Figure 4.1: **Overview of this chapter's main findings.** We find that 1) under popular PLMs, the likelihood of a protein sequence from certain species (e.g. humans, mice, and *E. coli*) is much higher than other species, 2) this bias arises from training data imbalance between different branches of the evolutionary tree of life, and 3) protein design guided by PLM likelihood systematically introduces mutations that increase similarity to high ranking species' sequences.

work we show how *protein* LMs inherit related properties from their pre-training, and show their impacts on the unique application of protein design.

Training data bias Biases in training data are reflected in downstream models. Under-represented subgroups can suffer lower accuracy due to insufficient weight in the training data [21, 25, 61, 111], and socially undesirable biases in data are often amplified by models [18, 23, 118]. Various papers have studied how re-weighting or curating datasets can mitigate these biases [109, 121, 128, 131], even finding that *overall* performance is improved by over-weighting minority groups and actively increasing diversity in datasets [46, 71, 107]. In each of these settings, identifying the precise subgroups models are biased against and connecting training representation to downstream harms has required careful experimental design. This work presents the case for applying similar scrutiny to protein sequence datasets–we identify species bias in PLMs, suggesting the existence of other, yet uncharacterized biases that affect the usefulness of PLMs.

Protein language models Many protein language models (PLMs) have been trained using transformers [36, 37, 43, 54, 79, 85, 91, 106], CNNs [129], and other architectures [6]. PLMs can generate sequences that successfully fold into functional proteins [5, 80, 123]. To

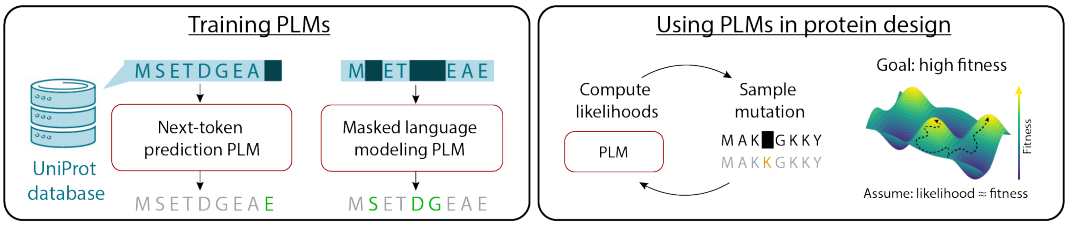


Figure 4.2: **Overview of PLM training and use in protein design.** Left: PLMs are trained on amino acid sequences from protein databases (most commonly UniProt) with either next-token prediction or masked language modeling tasks. Right: After training, PLMs may be used directly for protein design by picking a starting sequence and then iteratively generating a set of possible mutations, computing likelihoods of sequences with those mutations, and sampling based on these likelihoods. This designs sequences with high PLM likelihoods, which hopefully corresponds to high fitness.

modify existing proteins with enhanced properties, PLM likelihoods have been used as a proxy for fitness in the absence of experimental measurements for supervision [41, 55, 85]. If experimental measurements are available, various strategies can combine this supervision with PLMs to improve fitness prediction [56, 93]. Recent work has identified a tendency for PLMs and other protein models to classify a sequence as lower fitness if it has more mutations from a naturally occurring sequence (i.e., is more out-of-distribution), referred to as sequence similarity bias [112]. Our work focuses on a different bias arising from species identity, which is present even for protein sequences present in PLM training data, and which can compound the effect of sequence similarity bias.

4.3 PLM likelihoods are higher for sequences from certain species

We now turn to empirically investigating what factors affect PLM likelihoods. We collect a dataset of orthologous[1] sequences across the tree of life and across diverse protein types, and we compute PLM likelihoods for each sequence. Unsurprisingly, some protein types have much higher overall likelihoods than others (due to intrinsic disorder, conservation, etc.), but surprisingly, we find that some *species* also have much higher likelihoods than others (across proteins), and that this generalizes across PLMs with different training objectives and data sampling.

[1]Orthologs are genes/proteins in different species that evolved from a common ancestral gene/protein by speciation, and, in general, orthologs retain the same function during the course of evolution.

Dataset creation To create our protein dataset, we started with the top 100 most sequenced species in the UniProt database [29], filtered for redundancy, then augmented this list with additional model organisms that had whole genomes sequenced, resulting in 133 species total. Next we collected all protein sequences in the Swiss-Prot database [9] (the human-annotated subset of UniProt) associated with any of the species in our list. Based on their annotations, we divided the proteins into orthologous sets to be able to compare orthologs to each other. The vast majority of sequences were bacterial, so to create a balanced dataset with many points of comparison between eukaryotes and bacteria, we restricted our attention to proteins with at least 15 eukaryotic orthologs, resulting in 203 distinct protein types, and a total of 7545 sequences in our dataset, 40% being eukaryotic.

PLMs we study We focus on two families of PLMs in this work: the Progen2 suite [91] (in 5 sizes: xlarge, BFD90, large, base, and medium) and the ESM2 suite [79] (in 3 sizes: 15B, 3B, and 650M). These models are among the most popular for downstream use and achieve the best performance among PLMs on many benchmark tasks in ProteinGym [92]. Progen2 is an autoregressive transformer trained with next-token prediction on the UniRef90 database (a curated subset of UniProt clustered at 90% sequence identity). ESM2 has a bidirectional transformer architecture and is trained with the masked language modeling objective on data collected in a two-tiered sampling scheme: first randomly select a UniRef50 database member, and then sample a training data point from the UniRef90 cluster that member belongs to. For ESM models, we compute a pseudo-likelihood by masking each token in the sequence, as in Lin et al. [79].

Results

Variance explained by species identity We first investigate what factors explain PLM likelihoods in our dataset. We compute linear regressions of PLM likelihood against species, protein-type, and both at once, and report R^2 values in Table 4.1. We also compute the fraction of variance explained by the species after controlling for protein type. Protein type explains some of the variance, as expected, since proteins vary in prevalence, conservation, and other factors that intuitively affect likelihood. Surprisingly, species identity also explains a significant amount of the variance in PLM likelihoods; for example, for likelihoods from Progen2-xlarge, species accounts for 50% of the variance by itself, and 67% of the variance after controlling for protein type. This suggests that likelihoods have a species bias that holds consistently across the diverse universe of proteins.

We next quantify the bias associated with each species without assuming a linear model of likelihoods. Since each protein is only found in a subset of species, species cannot be fairly compared by a simple average likelihood score. To solve this, we use the Elo rating system, described below, to summarize how often one species has higher likelihoods than another. Note this provides a general framework for assessing bias against subgroups when we must handle "missing data" in the real world (in contrast to settings where we can construct synthetic experiments holding all variables equal except subgroup).

Table 4.1: **Variance in likelihood explained by species and protein type.** $R^2_{\text{Species|Protein}}$ is the fraction of variance explained by species identity, after controlling for protein type. (Example $R^2_{\text{Species|Protein}}$ derivation: 0.67 = (0.81 - 0.42)/(1 - 0.42).)

Model	R^2_{Species}	R^2_{Protein}	R^2_{Both}	$R^2_{\text{Species|Protein}}$
Progen2-xlarge	0.50	0.42	0.81	0.67
Progen2-BFD90	0.49	0.51	0.85	0.69
Progen2-large	0.46	0.60	0.87	0.67
Progen2-base	0.25	0.64	0.84	0.55
Progen2-medium	0.44	0.59	0.86	0.66
ESM2-15B	0.25	0.42	0.60	0.32
ESM2-3B	0.26	0.46	0.63	0.32
ESM2-650M	0.19	0.62	0.72	0.26

Quantifying species bias via Elo The Elo rating system was developed to calculate the relative skill levels of players in zero-sum games [38]. The difference in two players' Elo ratings directly translates to the probability of one player winning in a match against the other; for example, the 400 Elo difference between a chess grandmaster and a candidate master implies that the grandmaster is expected to win 90% of matches. In our setting, each time two species have different sequences of the same protein type, we count this as a "match", where the winner is the species with the higher likelihood for their sequence. If a species has multiple sequences of the same protein, its median likelihood is used to determine the match result. All species start with a baseline Elo rating of 1500, and each pair-wise matchup updates the winner's rating upwards and the loser's downwards in a stochastic gradient descent-like step. We use the standard Elo update algorithm with $K = 32$ and average results over 50 permutations of the matchups to ensure results are robust [19].

Figure 4.3 plots Elo ratings for each species in our dataset, annotated by phylogenetic taxa. Elo ratings vary widely across species. Using Progen2-xlarge likelihoods, the 25th percentile species (*A. baylyi*) has an Elo rating of 1235 while the 75th percentile species (*S. glossinidius*) has an Elo rating of 1745. This Elo difference of 510 implies that *S. glossinidius* has a higher likelihood for its orthologs 95% of the time. Similarly, if we use ESM2-15B pseudo-likelihoods to compute Elo ratings, there is a 220 Elo difference between the 25th and 75th percentile species, which implies an 80% chance of a higher likelihood. Both models thus have a significant species bias, with Progen2's being somewhat larger.

Progen2 and ESM2 also show largely similar biases: Elo ratings from Progen2-xlarge and ESM2-15B have a Pearson correlation of 0.83 (see Appendix 4.8 for correlations for all pairs of PLMs). Figure 4.3 further shows that the species bias has some interpretable trends: within eukaryotes, animals have the highest Elo ratings, and within animals, mammals do.

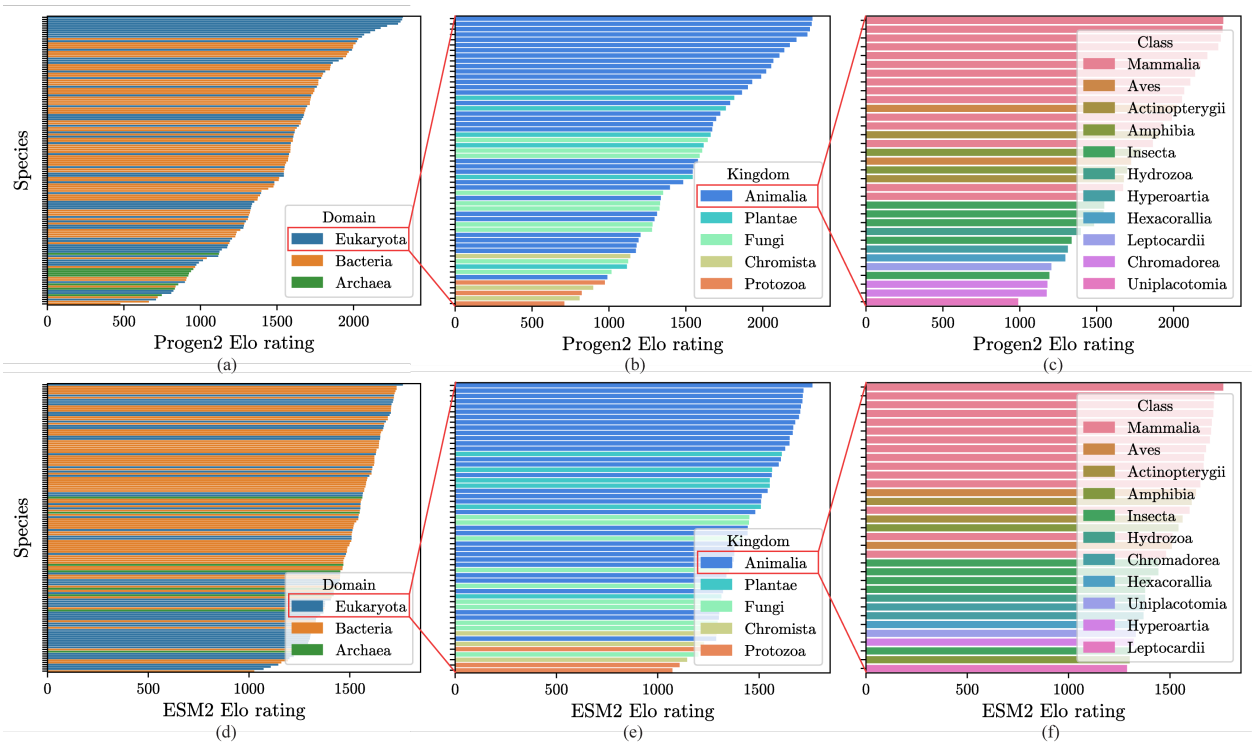


Figure 4.3: **Elo ratings for different species.** Elo ratings computed from Progen2-xlarge and ESM2-15B (top and bottom).

This species bias motivates understanding how it arises, which we study in the next section.

4.4 Bias is largely explained by species representation in sequence databases

We investigate what factors explain the species bias and find that species representation in popular sequence databases plays a major role. We test an initial hypothesis that Elo ratings will correlate with the number of sequences a species has in a database, and find that this only explains a small part of the bias. We next note that likelihoods may be influenced not only by a given species' sample counts, but also by evolutionarily-close species' counts. Taking this second factor into account, we posit a second hypothesis: *Elo ratings will correlate with an "effective" sequence count weighted by evolutionary distance*, and we show evidence for this second hypothesis.

We assess the initial hypothesis by plotting Elo rating against sequence counts in the SwissProt database in Figure 4.4 (a) and (c). Although a few species, such as *H. sapiens*,

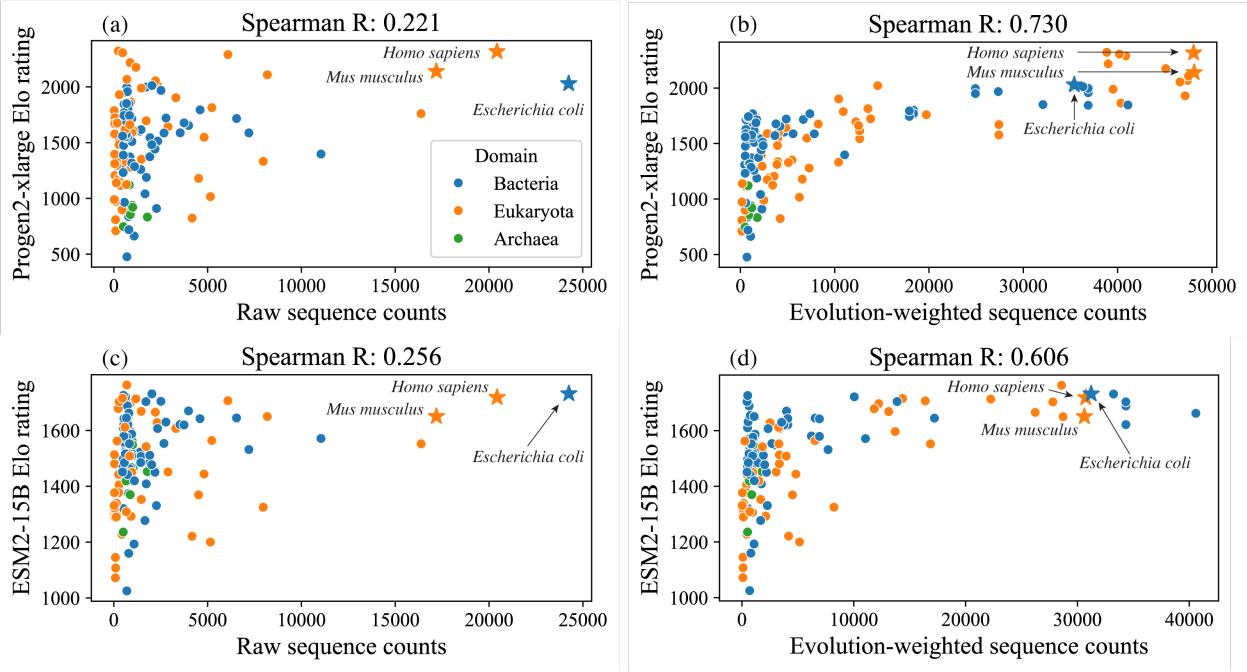


Figure 4.4: **Species Elo ratings plotted against their Swiss-Prot sequence counts and evolution-weighted sequence counts.** Top: Elo ratings computed from Progen2-xlarge likelihoods. Bottom: Elo ratings computed from ESM2-15B pseudolikelihoods. Correlation using raw sequence counts (left) is low, while correlation using evolution-weighted sequence counts (right) is high.

M. musculus, and *E. coli* show the expected trend of high sequence counts and high Elo ratings, most other species are not fit well. Overall, these raw sequence counts achieve Spearman correlations of only **0.221** and **0.256** with Progen2-xlarge and ESM2-15B Elo ratings, respectively.

One factor not captured by the initial hypothesis is sequence similarity due to evolution. Figure 4.5 displays the phylogenetic tree connecting the species in our dataset, annotated with sequence counts and Elo ratings. We see that a few model organisms have extremely large sequence counts, while Elo ratings are spread more diffusely across species. Many species with few sequence counts nonetheless have a high Elo rating, often when the species shares a recent common ancestor with a highly sequenced model organism. We conjecture that the sequence count from a given species contributes to the "effective" sequence count for another species in proportion to the sequence similarity between the two species' orthologs (see Appendix 4.8 for additional intuition). Assuming a Poisson model of mutations accumulated over time, sequence similarity between two species' orthologs is directly related to their

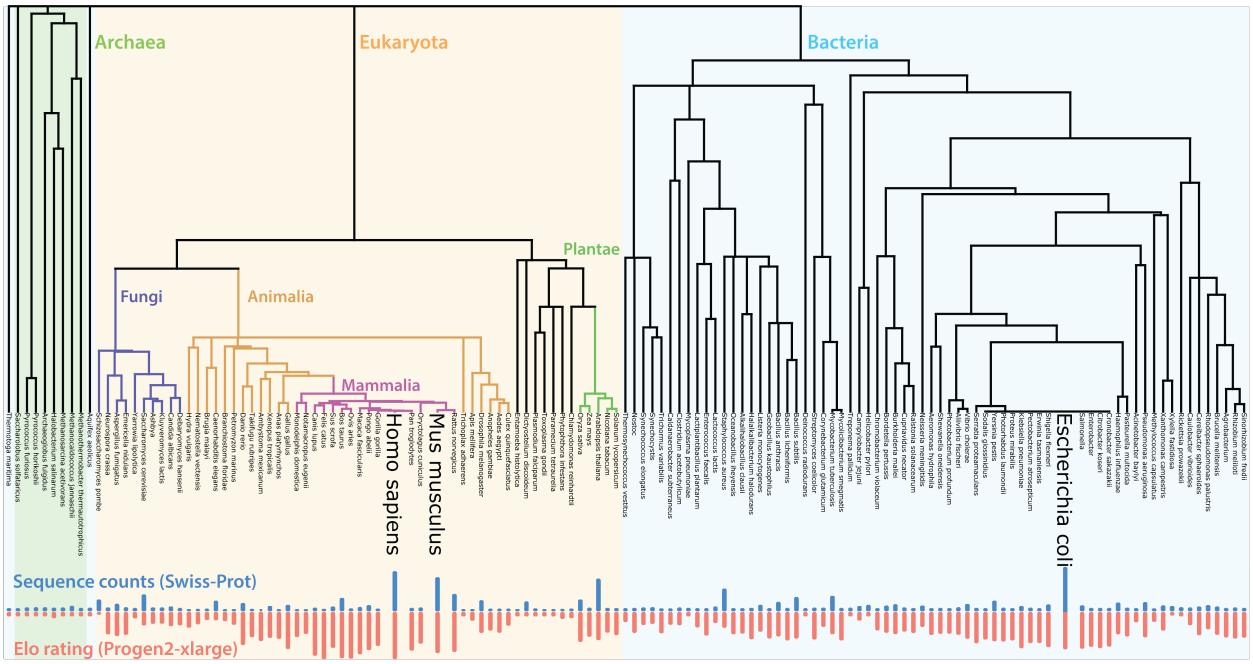


Figure 4.5: **Phylogenetic tree annotated with sequence counts and Elo ratings.** The tree's vertical axis represents time: number of years to the last common ancestor between two species is proportional to the distance from the leaves to the last common node between them.

evolutionary closeness. Thus we posit our second hypothesis that Elo ratings will correlate with an evolution-weighted sequence count, $n^{\text{evolution-weighted}}$, which we define as follows:

$$n_i^{\text{evolution-weighted}} = \sum_j n_j e^{-\frac{d(i,j)}{\alpha}},$$

where n_j is the raw sequence count for species j, $d(i,j)$ is the time to last common ancestor between species i and j collected from the TimeTree of Life resource [66], and $\alpha \in \mathbb{R}_{\geq 0}$ is a hyperparameter used to scale d appropriately. Under the assumption[2] that mutations occur at a fixed rate, $e^{-\frac{d(i,j)}{\alpha}}$ gives the expected overlap in sequence between two species' orthologs, to approximate the effective sequence counts they contribute to each other.

Figure 4.4(b, d) plot Elo rating against evolution-weighted sequence counts, and we see that this achieves much higher Spearman correlations than our initial hypothesis: **0.730** and **0.606**

[2]Mutation rates in fact vary across species and proteins, so as more species' proteomes are annotated, one could compute a more precise effective sequence count using exact sequence similarity between proteomes, and using this in place of the $e^{-\frac{d(i,j)}{\alpha}}$ term.

for Progen2-xlarge and ESM2-15B Elo ratings, respectively. Thus species representation, with the addition of evolutionary distance, explains a large fraction of the species bias in both PLMs.

It is interesting that both PLMs contain biases with this pattern because their training data differs in important respects. Recall that Progen2 was trained on random samples of UniRef90 (UniProt clustered at 90% sequence identity), while ESM2 used a two-tiered sampling scheme starting with UniRef50 (clustered at 50% identity). One motivation for using UniRef50 clusters is to avoid over-sampling similar, evolutionarily-close sequences. However, our results show that both sampling strategies create a similar species bias, likely because representative sequences for both UniRef50 and UniRef90 clusters are chosen first by annotation quality and second by whether they are from a model organism [116]. Additionally, species may have many similar sequences within each cluster, increasing the chances that one of them will be sampled.

4.5 PLM species bias affects protein design

Next, we investigate the impacts of species bias on protein design with PLMs. Since sequences from high Elo species have higher likelihoods, they may be basins of attraction when designing proteins to optimize likelihood. We test this prediction by simulating a simple protein design workflow and find that designs indeed systematically drift towards sequences from high Elo species.

We then study how this drift affects protein properties. The magnitude of drift is largest when the starting protein is from a low Elo species, and many of the species with the lowest Elo ratings are extremophiles that have adapted to hostile environments such as extreme heat, cold, pH, salinity, and even radiation. Despite their low Elo, these species' proteins often have unique properties that are useful for a variety of applications, such as research tools, therapeutics, and environmental bioremediation. This suggests that the typical use of likelihood for design will be detrimental when trying to enhance these protein's properties. We show that this is true for our simulated design workflow. We specifically study thermophilic (heat-loving) and halophilic (salt-loving) species and find that initially thermostable sequences have much lower predicted stability after design, and initially salt-tolerant sequences have lower predicted tolerance after design.

In this section, we first describe our design setup, next show that designs have species drift, and finally show that designs diminish heat and salt tolerance in extremophiles.

Overview of simulated design

In most protein design applications, scientists propose a *set* of candidate designs, with the goal of both high protein fitness and diversity in the set. We follow the design methodology in Zhu et al. [133] and Fannjiang et al. [41], who show that optimal tradeoffs between fitness

and diversity are achieved by sampling designs from sequence distributions that maximize entropy while satisfying constraints on mean fitness.

Formally, let $\mathcal{X}$ denote the set of all amino acid sequences of length L and let $\mathcal{P}$ denote set of all distributions over $\mathcal{X}$. We aim to sample from the sequence distribution, $p^\star$, that solves the following:

$$\begin{aligned} \arg\max_{p \in \mathcal{P}} \quad & H(p) \\ \text{subject to} \quad & \mathbb{E}_p[f(x)] \geq \tau, \quad \text{support}(p) \subseteq \textsc{HomologyRegion}(x_0) \end{aligned} \tag{4.1}$$

where $H(p)$ is the entropy of p, $f(x)$ is the sequence log-likelihood under some PLM, $\tau \in \mathbb{R}$ is a predicted fitness target threshold, and $\textsc{HomologyRegion}(x_0) \subseteq \mathcal{X}$ denotes the region of sequence space that is plausibly homologous to the starting sequence x_0 (to ensure we are designing sequences that still have similar function). Specifically, $x \in \textsc{HomologyRegion}(x_0)$ if and only if the top result returned by a Protein BLAST search [24] for x is an ortholog of x_0.

The distribution, $p^\star$, that solves (4.1) has a likelihood of the following form:

$$p^\star(x) \propto \begin{cases} \exp[\lambda f(x)] & \text{if } x \in \textsc{HomologyRegion}(x_0) \\ 0 & \text{otherwise,} \end{cases} \tag{4.2}$$

for Lagrange multiplier λ which has a one-to-one correspondence with the threshold τ in (4.1). Higher λ increases the average predicted fitness at the cost of lower diversity.

The distribution $p^\star$ is intractable to compute directly, so we use MCMC techniques to sample from it. From the starting sequence x_0, we iteratively propose mutations with Gibbs sampling for 10,000 steps, using Progen2-xlarge likelihoods for the predicted fitness $f(x)$. We set $\lambda = 1$ and sample 3 designs with different random seeds for each x_0. In total we collect 1092 designs using many different species' orthologs as starting sequences, as detailed in the following sections.

Protein design has a species drift

First we show that designs systematically drift towards high Elo species' orthologs, regardless of starting point. We generate designs for 20 different proteins, each with 15–20 different species' orthologs as starting points, with the species representing the full range of Elo ratings.

To quantify species drift, we define the *similarity-weighted Elo* of a sequence to reflect the average Elo of species who have orthologs similar to that sequence. Formally:

$$\text{Similarity-Weighted-Elo}(x) = \frac{1}{\sum_j s(x, x_j)} \sum_j \Big(s(x, x_j) \cdot \text{Elo}(j) \Big),$$

where x_j is the ortholog of x from species j, $s(x, x_j)$ is the sequence similarity between x and x_j, and $\text{Elo}(j)$ is the Elo rating of species j. We use $s(x, x_j) = (1 - \text{Levenshtein}(x, x_j))^2$, where

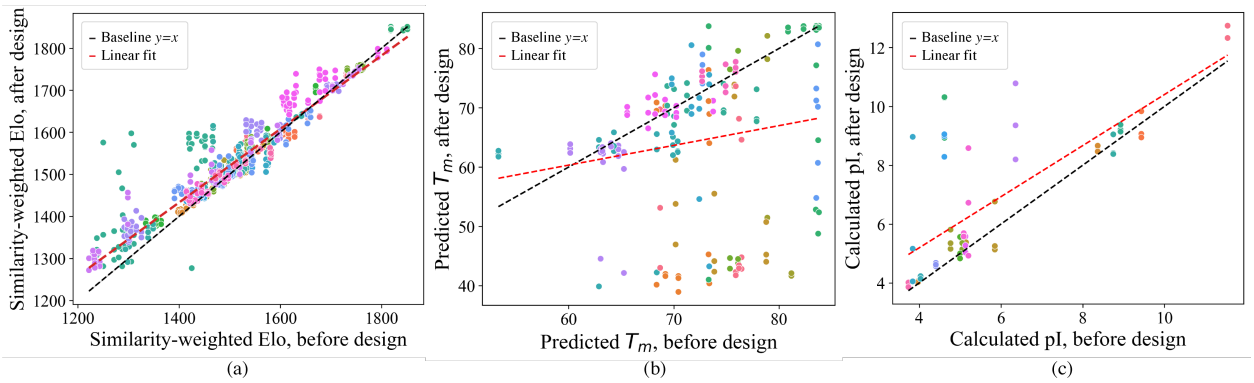


Figure 4.6: **Protein properties before and after protein design.** (a) Similarity weighted Elo is higher after design. (b) Predicted melting temperature (T_m) is lower (i.e. proteins are less stable) after design. (c) Calculated isoelectric point (pI) is higher (i.e. proteins are less tolerant to salt) after design. Each dot represents one protein design, and color indicates protein type (see Appendix 4.8).

the Levenshtein distance between two sequences is the minimum number of single-character edits required to change one sequence into another; other choices of s lead to similar results (Appendix 4.8).

Figure 4.6a plots the similarity-weighted Elo of a sequence before and after design. We see that final design sequences tend to increase their similarity-weighted Elo, and this increase is statistically significant under a paired sample t-test ($t = 15.2$, $p = 7.7\text{e-}47$). This drift holds across the spectrum of Elo ratings and is most prominent for low Elo starting points. This is consequential because many low Elo species are rich sources of useful proteins, as we discuss further in the following section.

Protein design diminishes extremophile properties

In this section we show that the unique adaptations in extremophile orthologs tend to be diminished after PLM likelihood-based design. Many of the species with the lowest Elo ratings are extremophiles. Despite their low Elo, these species' proteins are a valuable resource for developing novel biotechnology, with applications including research tools, therapeutics, and environmental bioremediation. For instance, thermophilic (heat-loving) species have evolved proteins with high thermostability (the ability to remain folded at high temperatures), necessary for many industrial uses of engineered proteins [87]. Similarly, halophilic (salt-loving) species have evolved proteins to be more acidic (negatively charged) to prevent aggregation in salty environments, and this salt tolerance is crucial to efforts such as biofuel production [31]. Since these thermophilic and halophilic adaptations occur at the individual

protein level, they provide perfect test cases to study the effects of protein design.

Effects on thermostability To examine protein design's effects on thermostability, we sample designs from 13 different proteins, each with 3–7 thermophilic species' orthologs as starting points. We assess the thermostability of each starting sequence and final design *in silico* using the protein melting temperature (T_m) predictor DeepSTABp [60].

Figure 4.6b plots the predicted melting temperature after design vs. before design. We see that designs tend to decrease their predicted melting temperatures, and we find that this decrease is statistically significant under a paired sample t-test ($t = -7.1$, $p = 3.5\text{e-}11$). 63% of designs have lower predicted melting temperatures than their starting sequence, and in one-third of those cases, the predicted decrease is over 20°C. This thermostability decrease is consequential for many applications. For example, engineering strains to successfully ferment bioethanol at 15°C higher temperatures makes a much wider set of raw materials economically feasible for biofuel production [86].

Effects on salt tolerance To examine protein design's effects on salt tolerance, we sample designs from 18 different proteins, each with 1–3 halophilic species' orthologs as starting points. We assess salt tolerance of a sequence through their isoelectric point (pI) [64], as proteins in halophilic species have lower pI than other species' orthologs in order to remain stable at high salt concentrations [49].

Figure 4.6c plots the calculated isoelectric point (pI) after design vs. before design. We see that designs tend to increase their pI, and this increase is statistically significant under a paired sample t-test ($t = 4.6$, $p = 2.0\text{e-}5$). 83% of designs have higher pI than their starting sequence, and the average pI increase of 4.6 may be consequential–the difference in pH between vinegar and neutral water is approximately 4-5.

Since we assess sequences' thermostability and salt tolerance *in silico*, these results rely on the accuracy of predictive models. As a sanity check for whether these results would replicate in experimental assays, we analyze the final design sequences and find that many have extremely high sequence similarity (>90%) to mammalian orthologs of the original protein. These orthologs have been experimentally verified to have lower melting temperatures and lower salt tolerance (see Appendix 4.8 for more results on convergence to orthologs). Thus, even though we only have access to predicted properties, these results suggest that protein design pushes sequences into a basin of attraction around sequences from well-represented species, often diminishing any unique properties.

4.6 Bias mitigation

Finally, we investigate strategies for mitigating PLM species bias. Because most downstream users of PLMs have limited computational budgets insufficient to re-train a PLM, we focus our attention on the setting of *post-hoc* bias mitigation given a biased PLM. However, in

Table 4.2: **Bias mitigation reduces losses in thermostability and salt tolerance.**

Model	Mean (SE) ΔT_m after design ↑	Mean (SE) Δ pI after design ↓
Uncorrected	-26.34 (2.41)	3.25 (0.49)
KNN corrected	**-18.39** (3.05)	2.74 (0.52)
RF corrected	-19.78 (2.48)	**1.06** (0.45)

Section 4.7, we also discuss bias mitigation strategies relevant for PLM *pre-training* that may be fruitful avenues for future work.

Here we describe a post-hoc bias mitigation strategy based on the assumption that all natural orthologs of the same protein should ideally have the same likelihood. Formally, for every sequence x, we define its ideal log-likelihood $f^*(x)$ as $\max_{y \in \text{orthologs}(x)} f(y)$, which we can compute with our dataset. Our strategy is to train an auxiliary model, $g : \mathbb{R}^D \to \mathbb{R}$, which takes as input the PLM embedding of a sequence x and predicts $f^*(x) - f(x)$, the additive correction term to make x attain its ideal log-likelihood. The functional form of g is flexible; here we implement K-nearest neighbors (KNN) and random forest (RF) regressors to demonstrate the capabilities of even low-cost auxiliary models.

We train the auxiliary model g on a subset of our dataset, test it on held-out sequences, and evaluate the species bias on this test set. The results are shown in Table 4.3. The model is able to generalize to sequences from unseen proteins and species, reducing species bias significantly.

Further, we incorporate the auxiliary model within our design sampling algorithm to test whether bias mitigation improves designs. Specifically, we replace the log-likelihood $f(x)$ used in Eq. 4.2 with $f(x) + g(\text{PLM embedding}(x))$, and otherwise follow the same algorithm to generate designs from thermophilic and halophilic species. The results are shown in Table 4.2. Both the KNN and RF correction models substantially mitigate the effects of species bias in protein design: the melting temperatures of final designs are higher and the isoelectric points of final designs are lower, compared to using uncorrected likelihoods. This reinforces our findings in Section 4.5 that species bias is a *causal* contributor to loss in thermostability and salt-tolerance. However, neither model completely eliminates losses in these properties, so there remains room for improvement for novel post-hoc mitigation strategies.

4.7 Discussion

In this work we identify and quantify a species bias in PLM likelihoods, trace its origins to uneven sequence sampling across the tree of life, and document its effect of pushing protein designs toward sequences from favored species. This design bias is most likely to be detrimental when the starting point comes from an organism under-represented in sequence databases, and we demonstrate that likelihood-guided design can reduce the thermostability

of proteins from heat tolerant species and reduce the salt-tolerance of proteins from species that thrive at high salinity.

We also show a proof of concept bias mitigation strategy through training an auxiliary model to correct PLM likelihoods, post-hoc. This auxiliary model helps to preserve the thermostability and salt-tolerance of extremophile sequences, but not perfectly. Other post-hoc strategies worth future investigation may include new design algorithms that adjust the acceptance rate of proposed mutations based on whether the mutation moves towards a common or uncommon ortholog.

Additionally, since training data representation is driving the bias, curating the pre-training data for PLMs is a highly promising direction for bias mitigation. Two possible sampling strategies to reduce bias include: sampling sequences from branches of the tree of life according to desired frequencies for each branch, and sampling sequences inversely proportional to their effective representation in databases (i.e., taking into account the prevalence of highly similar sequences).

However, it is also possible that in some settings the bias will happen to align with design goals. For example, antibody therapeutics are often produced from non-human sources and can generate immunogenic responses in humans [81]. It would be interesting to test PLM capabilities for humanizing antibodies such that they do not elicit an immune response and thus become safe for therapeutic use.

More broadly, this work highlights the importance of data curation for biological datasets. Training PLMs has only been possible due to decades-long efforts from scientists to standardize sequence information in huge, public databases. While the databases at first primarily served as a resource for protein information queries, today they are additionally treated as defining *distributions* over natural protein sequences. As databases and models grow, it is critical to understand biases present in the data collection process, evaluate whether mitigation of these biases is warranted, and leverage the rich annotations and meta-data in these databases to curate training data tailored to downstream use-cases.

4.8 Supplementary Materials

Broader Impacts

Protein design is an active, rapidly changing research field research with applications across medicine, biotechnology, and environmental sustainability. PLMs are one of many computational tools leveraged in protein design, and it is crucial to understand their strengths, weaknesses, and what information they encode. This paper highlights an unintentional bias learned by PLMs that can be detrimental in certain protein design settings. Mitigating this bias may help PLMs become more effective at accelerating protein design, with far-reaching beneficial impacts for human and environmental health.

At the same time, advancements in PLMs and other biological design tools, such as AlphaFold, have raised the concern that they may be used for the development of biological

weapons or other harmful technology. Research based on this paper that improves PLM performance could increase such risks. We hope that by understanding PLMs more thoroughly and collaborating with stakeholders to identify key biosecurity control points, such as physically manufacturing synthetic DNA [10], progress in protein design can achieve its beneficial potential while mitigating risks.

Stylized model for PLM bias

In this section, we provide a brief conceptual example to illustrate why PLMs learn a species bias. PLMs are sequence models with architectures and training objectives inspired by language models trained on text data (Figure 4.2). Most if not all protein language models are trained on samples from the UniProt protein sequence database, with either an autoregressive (AR) next-token prediction task or a masked language modeling task. For the stylized results in this section, we focus on the autoregressive loss shown below:

$$\mathcal{L}_{\text{AR}} = \mathop{\mathbb{E}}_{x\sim X}\big[-\log p(x)\big] = \mathop{\mathbb{E}}_{x\sim X}\left[\sum_i -\log p\left(x_i|x_{<i}\right)\right] \tag{4.3}$$

where x is a protein sequence, X is the distribution of sequences defined by the training set and i indexes a single token in x.

Consider a stylized model in which protein sequences are represented as vectors $x \in \mathbb{R}^D$, and protein sequences from each species are Gaussian distributed. As a result, the set of all protein sequences forms a mixture of Gaussians. In this setting, if the training data has k_s samples from species s, a PLM that optimizes likelihood (as in Eq. 4.3) will learn a mixture of Gaussians with weight $\frac{k_s}{n}$.

This stylized setting highlights two trends with likelihoods. First, in the case that different species' Gaussian distributions have minimal overlap, species that are sampled more, i.e. have larger k_s, will have higher likelihoods (Figure 4.7a). Second, if some species' distributions cluster together and overlap significantly, additive effects will make all those species' sequences have higher likelihoods (Figure 4.7b).

We will see both of these effects empirically. In the UniProt database, a few species have far more sequences recorded than others. These species, as well as species in close evolutionary proximity to them, have the highest likelihoods.

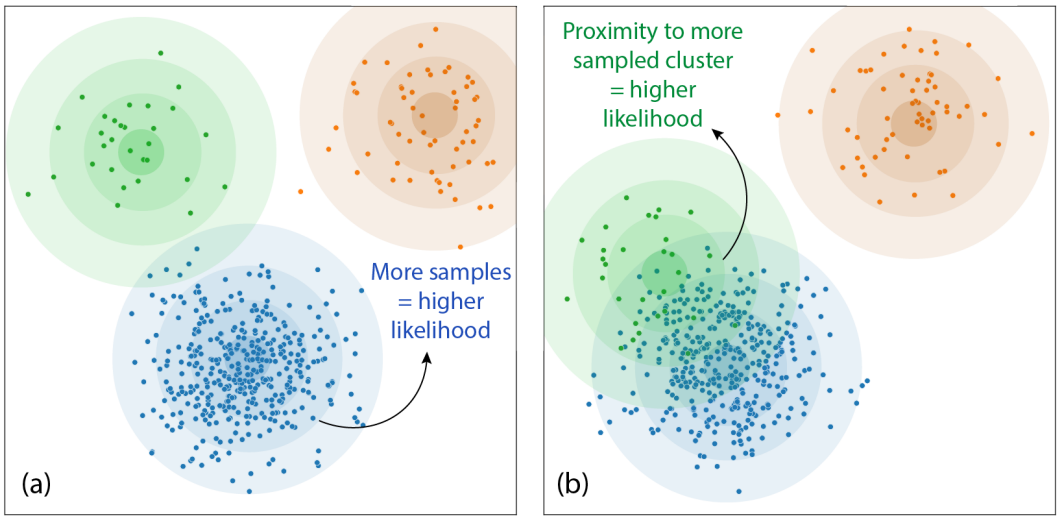


Figure 4.7: **Illustration of biases induced by maximum likelihood training.** Left: in well-separated settings, species with the highest prevalence in the training data will have the highest likelihood. Right: when there is overlap between distributions, species clustered together will increase each others' likelihoods.

Elo ratings from different PLMs

Figure 4.8 plots the correlation between Elo ratings computed from different PLMs. We see that PLMs within the same family have nearly identical Elo ratings, and PLMs across families also correlate highly.

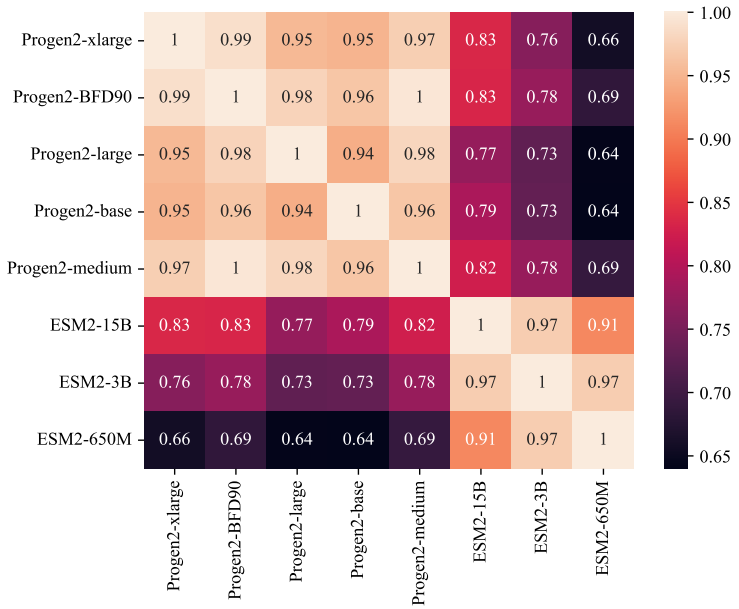


Figure 4.8: **Heatmap of the Pearson correlation between Elo scores from different PLMs.**

Dataset creation details

We started with the most represented species in the Swiss-Prot database (available from the UniProt statistics page) filtered for redundancy, then added organisms with annotated genomes from the NCBI Genome Server to include at least one organism from each listed category (e.g. Fish, Insects). We also required each organism to have at least 100 entries in the Swiss-Prot database, which resulted in 133 final species. We next used the UniProt REST API to download all Swiss-Prot sequences from each of the selected species. We divided the proteins into orthologous sets based on the protein name and function annotations. The majority of sequences collected were bacterial, so to create a more balanced dataset that allowed for comparison between eukaryotes and bacteria, we kept a protein in our dataset if they had at least 15 eukaryotic orthologs.

Robustness to different database sequence counts

In the main text, Figure 4.4 shows the correlation between Elo scores and raw sequence counts, and between Elo scores and evolution-weighted sequence counts, using the number of SwissProt entries per species as the raw sequence count. Here we show that we find similar results by using other choices of raw sequence counts. Figure 4.9 shows results with UniRef90 sequence counts. Figure 4.10 shows results with sequence counts tallied from two-tiered sampling from UniProt: first sample a representative member of UniRef50, then sample a protein sequence from the UniRef90 cluster that the member belongs to. In all cases, correlation with Elo ratings is significantly higher with evolution-weighted sequence counts.

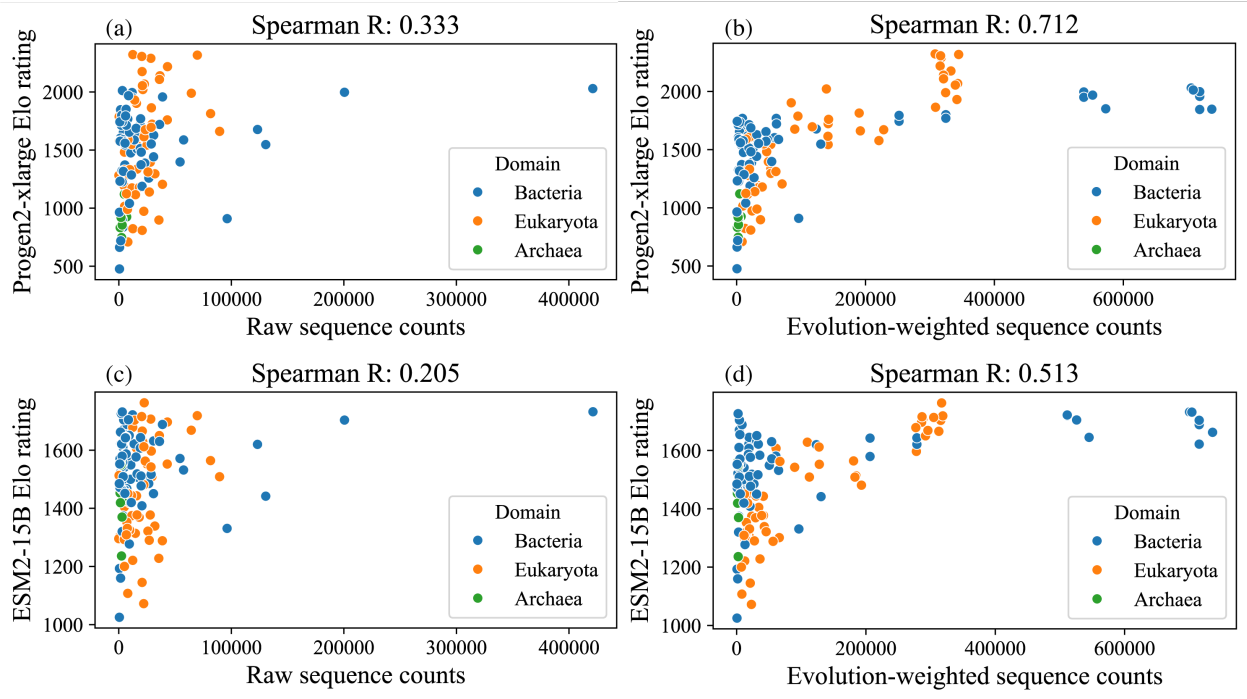


Figure 4.9: **Species Elo ratings plotted against their UniRef90 sequence counts and evolution-weighted sequence counts.** Top: Elo ratings computed from Progen2-xlarge likelihoods. Bottom: Elo ratings computed from ESM2-15B pseudolikelihoods. Correlation using raw sequence counts (left) is low, while correlation using evolution-weighted sequence counts (right) is higher.

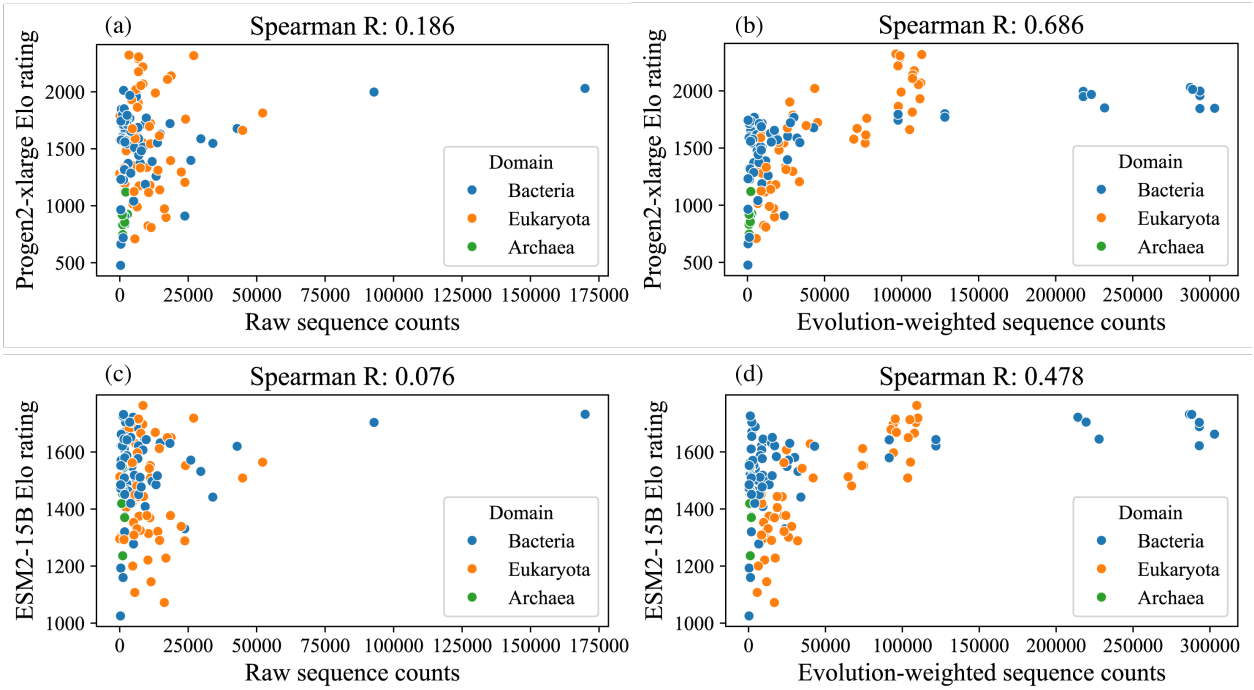


Figure 4.10: **Species Elo ratings plotted against sequence counts from two-tiered sampling: first sample a representative from UniRef50 and then sample a sequence from the UniRef90 cluster of the representative.** Top: Elo ratings computed from Progen2-xlarge likelihoods. Bottom: Elo ratings computed from ESM2-15B pseudolikelihoods. Correlation using raw sequence counts (left) is low, while correlation using evolution-weighted sequence counts (right) is higher.

Details on protein design experiments

Details for similarity-weighted Elo

For the results in the main text, we compute similarity-weighted Elo with $s(x, x_j) = (1 - \text{Levenshtein}(x, x_j))^2$. Here we show that results are robust to other choices of s. We use the Bio.Align package to score the optimal alignment between sequences with the Smith-Waterman algorithm, using the BLOSUM62 matrix to score the penalties for each amino acid substitution. Similar amino acids receive a smaller edit penalty compared to more chemically distinct amino acids, and we use the maximum alignment score as the similarity s. In contrast, the Levenshtein distance assigns an equal penalty to all substitutions. Figure 4.11 plots similarity-weighted Elo after design vs. before design. We see that for both similarity metrics, similarity-weighted Elo increases after design.

Proteins were selected for design only if they had at least 15 orthologs in Swiss-Prot and if the Elo ratings of the species with orthologs spanned at least 200. Under the computational constraints of how many designs could be generated, proteins were chosen to cover different types of functions (e.g. enzymes, structural proteins, etc.)

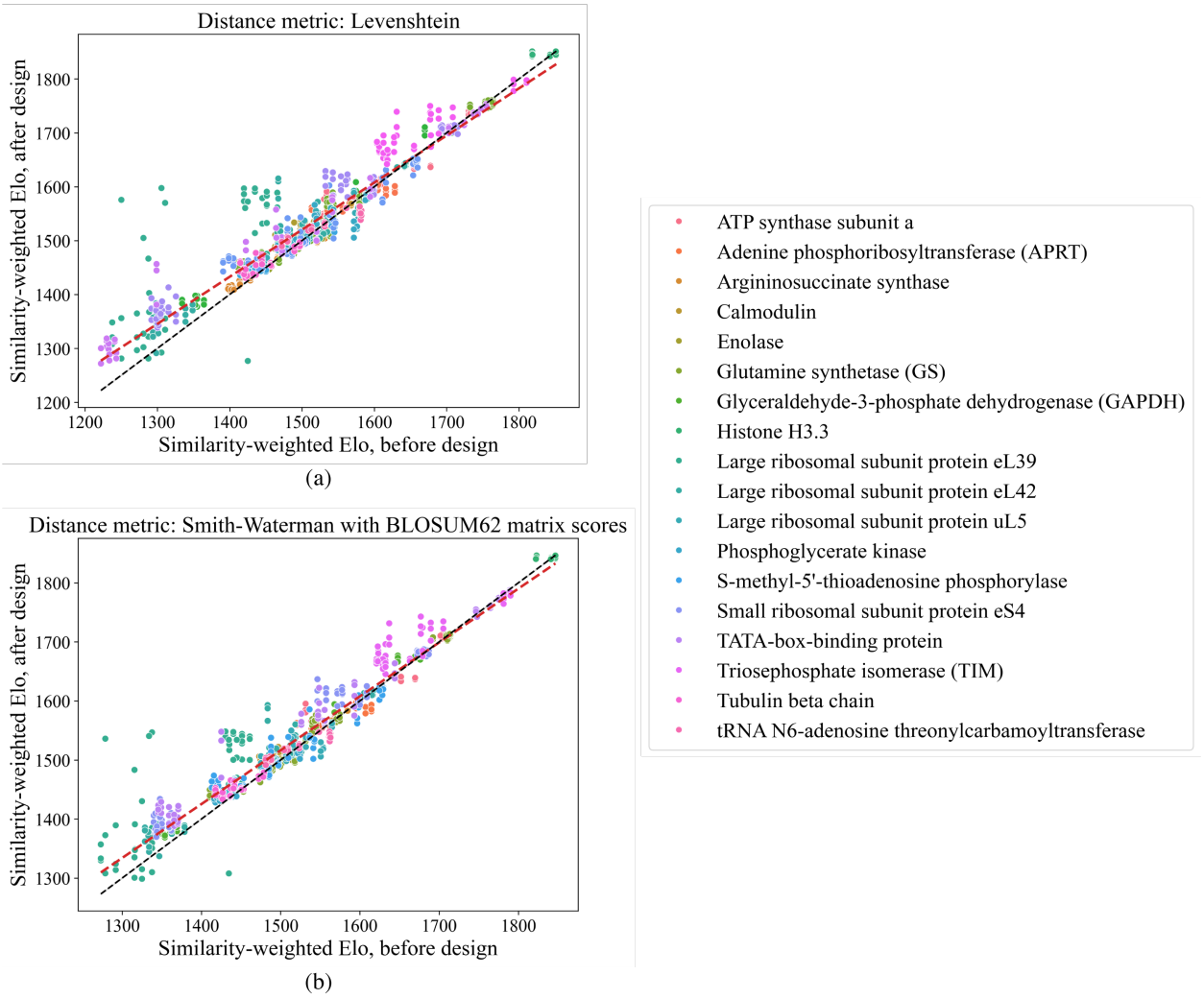


Figure 4.11: **Similarity-weighted Elo before and after design.** Top: similarity-weighted Elo computed using Levenshtein distance. Bottom: similarity-weighted Elo computed using the Smith-Waterman algorithm to compute a local alignment under BLOSUM62 matrix scores for amino acid substitutions.

Details for thermostability

We study the 7 thermophilic species out of the original 133 species collected in our dataset: *Methanocaldococcus jannaschii*, *Archaeoglobus fulgidus*, *Methanothermobacter thermautotrophicus*, *Methanosarcina acetivorans*, *Pyrococcus furiosus*, *Pyrococcus horikoshii*, and *Saccharolobus solfataricus*. Figure 4.12 plots the same data as Figure 4.6b with the legend added. Proteins were selected for design only if they had at least 3 orthologs from thermophilic species, and under the computational constraints of how many designs could be generated, proteins were chosen to cover different types of functions (e.g. enzymes, structural proteins, etc.)

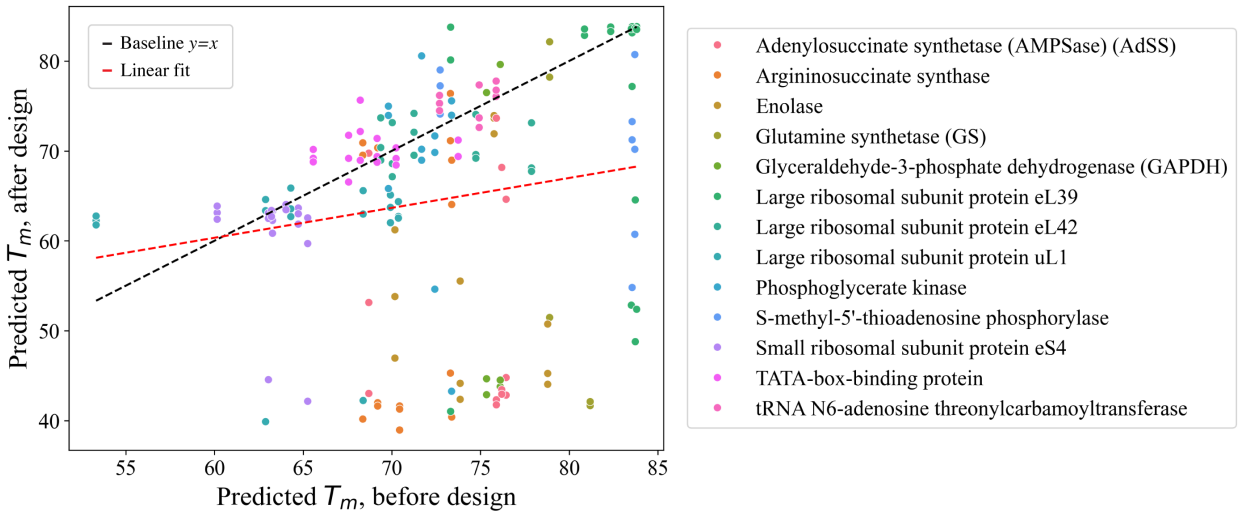


Figure 4.12: **Predicted melting temperature (T_m) after design vs. before design.**

Details for salt tolerance

We study the 3 halophilic species out of the original 133 species collected in our dataset: *Halobacterium salinarum*, *Alkalihalobacillus clausii*, and *Halalkalibacterium halodurans*. Figure 4.13 plots the same data as Figure 4.6c with the legend added. The proteins used for design in this experiment are all ribosomal because ribosomal proteins are generally basic, and thus provide a focused test case to study whether halophilic species' orthologs become more basic after design.

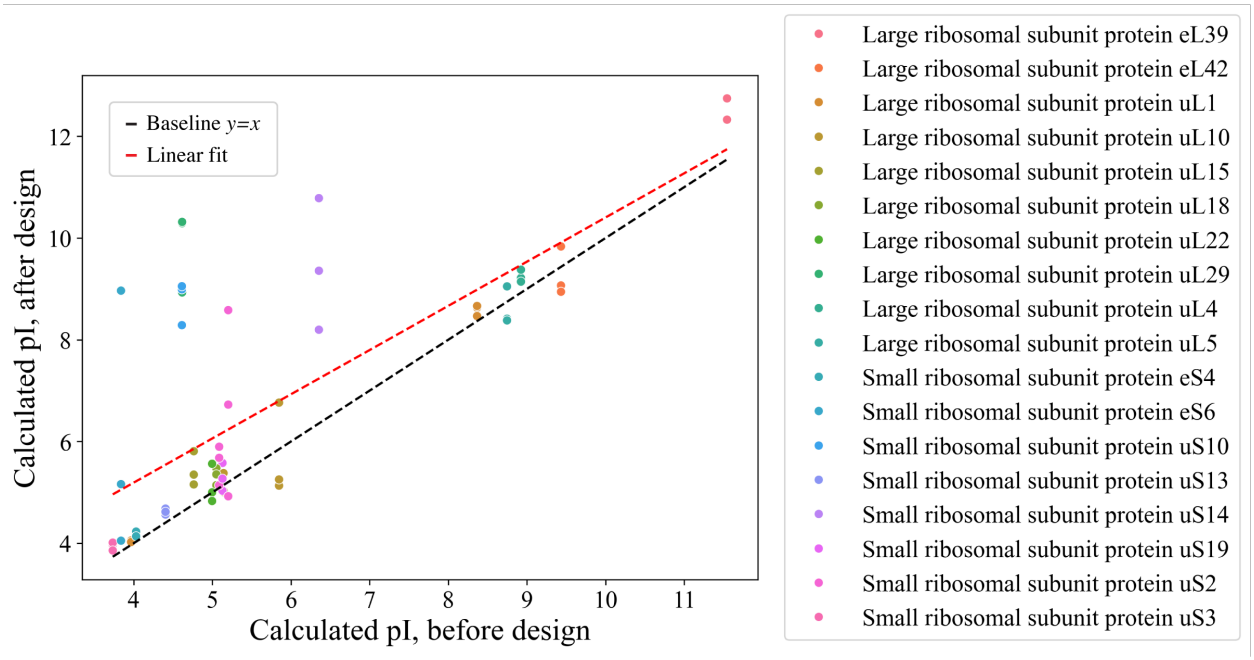


Figure 4.13: **Calculated isoelectric point (pI) after design vs. before design.**

Convergence to high Elo orthologs after design

We find that some protein designs result in sequences nearly identical to orthologs from high Elo species. To quantify this, we compute the fraction of designs that "converge" to a ortholog in our dataset, when we set the threshold for convergence to be 90%, 95%, and 98% sequence identity. Figure 4.14 plots these convergence frequencies. We see that the convergence rate varies significantly between different proteins, and that convergence happens much more often to a higher Elo species' ortholog compared to a lower Elo species' ortholog.

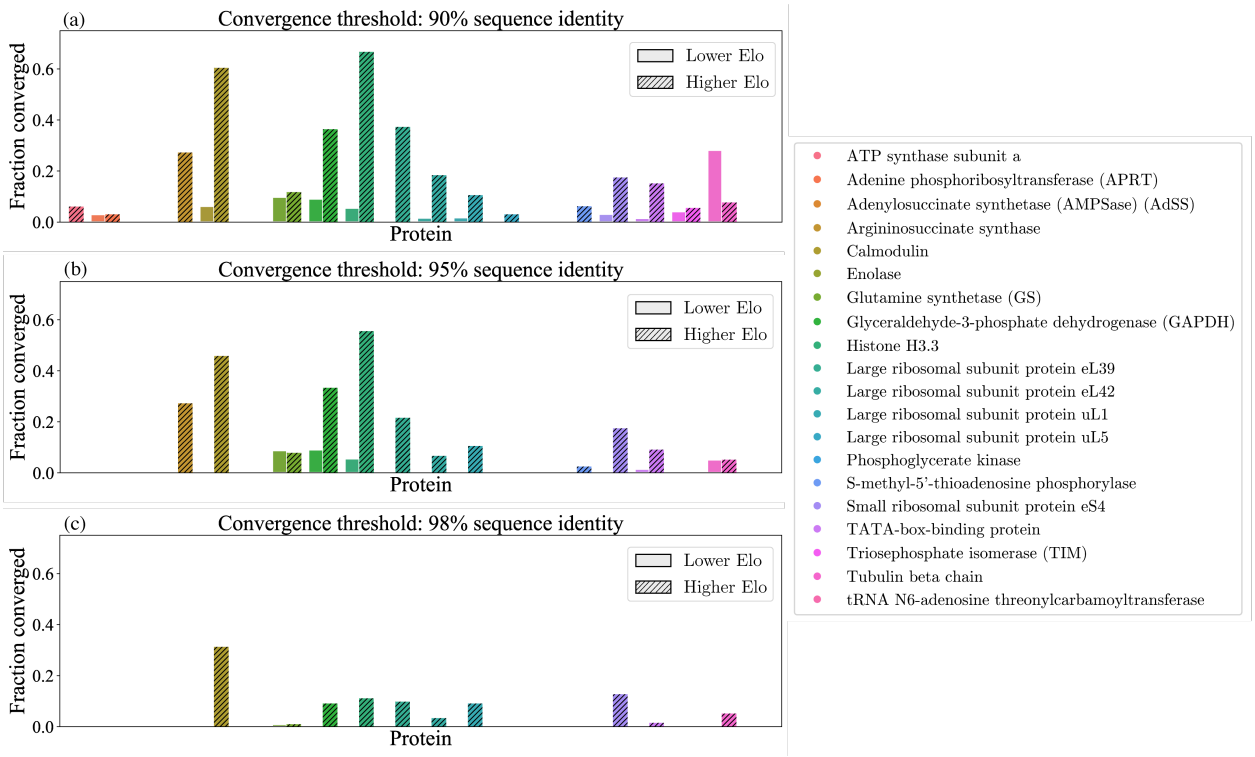


Figure 4.14: **Frequency of convergence to naturally-occurring orthologs from a different species.** Convergence to a lower Elo species' ortholog is represented by the solid bars, and convergence to a higher Elo species' ortholog is represented by the shaded bars. The threshold for convergence is 90% sequence identity, 95% sequence identity, and 98% sequence identity for the top, middle, and bottom, respectively.

Details on bias mitigation experiments

We mitigate bias by training an auxiliary model to correct sequences' PLM likelihoods. We train two types of models: 1) K-nearest neighbors regressor, using

```
n_neighbors=5, weights='uniform'
```

We evaluate this model on its ability to generalize out of distribution, i.e., to reduce bias in likelihoods for sequences that belong to protein families or species not seen during training. To quantify bias with a single scalar, one natural metric is the spread of the distribution of Elo ratings across different species, specifically the interquartile range (IQR), i.e., the difference between the 75th percentile and the 25th percentile Elo rating.

We split our dataset into a train set and a test set, train a correction model on the train set, and then for each sequence in the test set, add its predicted correction to its original likelihood to get a "corrected" likelihood. We then compute Elo ratings on these corrected likelihoods and calculate the IQR of the Elo rating distribution. To test the generalizability of the bias mitigation, we ensure that the train and test sets are meaningfully different – we show results where they are split based on protein identity, species identity, or both variables.

Table 4.3: **Auxiliary likelihood correction model successfully mitigates bias.** For each of the train-test splits (protein, species, and both), we show the test set Elo IQR Mean (SE) based on likelihoods from the uncorrected, K-Nearest Neighbors corrected, and Random Forest corrected models. Lower scores indicate less bias.

Model	Protein split	Species Split	Protein and species split
Uncorrected	505.7 (18.2)	449.0 (18.2)	473.4 (13.2)
KNN corrected	**168.5** (6.4)	**136.0** (5.3)	**160.8** (7.1)
RF corrected	229.7 (4.2)	173.5 (8.4)	210.2 (8.5)

We report the mean and standard error of IQRs in Table 4.3 over 10 random train-test splits in each category. These results show that the disparity in Elo ratings can be significantly reduced by training an auxiliary likelihood correction model. The K-Nearest Neighbors model appears to be more effective than the Random Forest model at reducing the bias, and it would be interesting to also further test whether finetuning a PLM to correct for the bias would perform as well or better.

Compute

To collect PLM likelihoods for sequences in our dataset, 2 A100 GPUs on our internal cluster were used for 24 hours to run inference on the Progen2 and ESM2 models we study. For our

protein design experiments, each design run used either 1 A100 or 1 A4000 GPU (dependent on whether the sequences demanded the memory available in an A100), and took between 1 hour and 8 hours to complete. In total we approximate that the experiments required 30 days on an A4000 GPU and 14 days on an A100 GPU.

Licenses

The Progen2 model is available at `https://github.com/enijkamp/progen2` under the BSD-3-Clause license.

The ESM2 model is available at `https://github.com/facebookresearch/esm` under the MIT license.

The UniProt database (`https://www.uniprot.org/help/copyright`) provides access under the CC BY 4.0 License.

Bibliography

[1] Rediet Abebe, Kehinde Aruleba, Abeba Birhane, Sara Kingsley, George Obaido, Sekou L Remy, and Swathi Sadagopan. Narratives and counternarratives on data sharing in africa. In *Proc. of the ACM Conference on Fairness, Accountability, and Transparency*, pages 329–341, 2021.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 31:9505–9515, 2018.

[3] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[4] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2018.

[5] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pages 2023–09, 2023.

[6] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[7] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[8] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2017.

[9] Amos Bairoch and Rolf Apweiler. The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic acids research*, 28(1):45–48, 2000.

[10] David Baker and George Church. Protein design meets biosecurity, 2024.

[11] Michelle Bao, Angela Zhou, Samantha Zottola, Brian Brubach, Sarah Desmarais, Aaron Horowitz, Kristian Lum, and Suresh Venkatasubramanian. It's compaslicated: The messy relationship between rai datasets and algorithmic fairness benchmarks. *arXiv preprint arXiv:2106.05498*, 2021.

[12] Solon Barocas and Andrew D. Selbst. Big data's disparate impact. *California Law Review*, 104, 2016.

[13] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

[14] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, 2017.

[15] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

[16] Ruha Benjamin. *Race after Technology*. Polity, 2019.

[17] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep. MSR-TR-2020-32*, 2020.

[18] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in Neural Information Processing Systems*, 2016.

[19] Meriem Boubdir, Edward Kim, Beyza Ermis, Sara Hooker, and Marzieh Fadaee. Elo uncovered: Robustness and best practices in language model evaluation. *arXiv preprint arXiv:2311.17295*, 2023.

[20] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38 (8):2102–2110, 2022.

[21] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Fairness, Accountability and Transparency*, pages 77–91, 2018.

[22] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *In Proc. IEEE ICDMW*, pages 13–18, 2009.

[23] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[24] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. Blast+: architecture and applications. *BMC bioinformatics*, 10:1–9, 2009.

[25] Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? *Advances in neural information processing systems*, 31, 2018.

[26] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[27] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

[28] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data, 2018.

[29] The UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 11 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac1052. URL `https://doi.org/10.1093/nar/gkac1052`.

[30] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

[31] Lobna Daoud and Mamdouh Ben Ali. Chapter 5 - halophilic microorganisms: Interesting group of extremophiles with important applications in biotechnology and environment. In Richa Salwan and Vivek Sharma, editors, *Physiological and Biotechnological Aspects of Extremophiles*, pages 51–64. Academic Press, 2020. ISBN 978-0-12-818322-9. doi: https://doi.org/10.1016/B978-0-12-818322-9.00005-8. URL `https://www.sciencedirect.com/science/article/pii/B9780128183229000058`.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[33] Frances Ding and Jacob Noah Steinhardt. Protein language models are biased by unequal sequence sampling across the tree of life. *bioRxiv*, 2024. doi: 10.1101/2024.03.07.584001. URL `https://www.biorxiv.org/content/early/2024/03/12/2024.03.07.584001`.

[34] Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. *Advances in Neural Information Processing Systems*, 34, 2021.

[35] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[36] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[37] Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling. *bioRxiv*, pages 2023–01, 2023.

[38] Arpad E Elo. *The Rating of Chess Players: Past and Present*. Ishi Press International, 1978.

[39] Hyunuk Eom, Kye Soo Cho, Jihyeon Lee, Stephanie Kim, Sukhwan Park, Hyunbin Kim, Jinsol Yang, Young-Hyun Han, Juyong Lee, Chaok Seok, et al. Discovery of highly active kynureninases for cancer immunotherapy through protein language model. *bioRxiv*, pages 2024–01, 2024.

[40] Virginia Eubanks. *Automating inequality: How high-tech tools profile, police, and punish the poor*. St. Martin's Press, 2018.

[41] Clara Fannjiang, Micah Olivas, Eric R Greene, Craig J Markin, Bram Wallace, Ben Krause, Margaux M Pinney, James Fraser, Polly M Fordyce, Ali Madani, et al. Designing active and thermostable enzymes with sequence-only predictive models. In *NeurIPS 2022 Workshop on Learning Meaningful Representations of Life*, 2022.

[42] Yunzhen Feng, Runtian Zhai, Di He, Liwei Wang, and Bin Dong. Transferred discrepancy: Quantifying the difference between representations. *arXiv preprint arXiv:2007.12446*, 2020.

[43] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[44] Sarah Flood, Miriam King, Renae Rodgers, Steven Ruggles, and J. Robert Warren. Integrated Public Use Microdata Series, Current Population Survey: Version 8.0 [dataset], 2020. Minneapolis, MN: IPUMS, `https://doi.org/10.18128/D030.V8.0`.

[45] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

[46] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[47] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *arXiv:1803.09010*, 2018.

[48] Mary L Gray and Siddharth Suri. *Ghost work: how to stop Silicon Valley from building a new global underclass*. Eamon Dolan Books, 2019.

[49] Nina Gunde-Cimerman, Ana Plemenita**v**s, and Aharon Oren. Strategies of adaptation of microorganisms of the three domains of life to high salt concentrations. *FEMS microbiology reviews*, 42(3):353–375, 2018.

[50] Moritz Hardt and Benjamin Recht. *Patterns, predictions, and actions: A story about machine learning*. `https://mlstory.org`, 2021.

[51] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Proc. 29th NIPS*, pages 3315–3323, 2016.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[53] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[54] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. Rita: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

[55] Brian L Hie, Varun R Shanker, Duo Xu, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 2023.

[56] Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature biotechnology*, 40(7): 1114–1122, 2022.

[57] Milind Jagota, Chengzhong Ye, Carlos Albors, Ruchir Rastogi, Antoine Koehl, Nilah Ioannidis, and Yun S Song. Cross-protein transfer learning substantially improves disease variant prediction. *Genome Biology*, 24(1):182, 2023.

[58] Eun Seo Jo and Timnit Gebru. Lessons from archives: strategies for collecting sociocultural data in machine learning. In *Fairness, Accountability, and Transparency*, pages 306–316, 2020.

[59] Sean R Johnson, Xiaozhi Fu, Sandra Viknander, Clara Goldin, Sarah Monaco, Aleksej Zelezniak, and Kevin K Yang. Computational scoring and experimental evaluation of enzymes generated by neural networks. *bioRxiv*, pages 2023–03, 2023.

[60] Felix Jung, Kevin Frey, David Zimmer, and Timo Mühlhaus. Deepstabp: A deep learning approach for the prediction of thermal protein stability. *International Journal of Molecular Sciences*, 24(8):7444, 2023.

[61] Giona Kleinberg, Michael J Diaz, Sai Batchu, and Brandon Lucke-Wold. Racial underrepresentation in dermatological datasets leads to biased machine learning models and inequitable healthcare. *Journal of biomed research*, 3(1):42, 2022.

[62] Ronny Kohavi and Barry Becker. Uci adult data set. *UCI Meachine Learning Repository*, 5, 1996.

[63] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529, 2019.

[64] Lukasz P Kozlowski. Ipc–isoelectric point calculator. *Biology direct*, 11(1):1–16, 2016.

[65] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[66] Sudhir Kumar, Michael Suleski, Jack M Craig, Adrienne E Kasprowicz, Maxwell Sanderford, Michael Li, Glen Stecher, and S Blair Hedges. Timetree 5: an expanded resource for species divergence times. *Molecular Biology and Evolution*, 39(8):msac174, 2022.

[67] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*, 2020.

[68] Aarre Laakso and Garrison Cottrell. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical psychology*, 13(1):47–76, 2000.

[69] Pat Langley. The changing science of machine learning, 2011.

[70] Matthew L Leavitt and Ari Morcos. Towards falsifiable interpretability research. *arXiv preprint arXiv:2010.12016*, 2020.

[71] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.577. URL `https://aclanthology.org/2022.acl-long.577`.

[72] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence, 2015.

[73] Amanda Levendowski. How copyright law can fix artificial intelligence's implicit bias problem. *Wash. L. Rev.*, 93:579, 2018.

[74] Shaofeng Li, Shiqing Ma, Minhui Xue, and Benjamin Zi Hao Zhao. Deep learning backdoors. *arXiv preprint arXiv:2007.08273*, 2020.

[75] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *Feature Extraction: Modern Questions and Challenges*, pages 196–212. PMLR, 2015.

[76] Ruofan Liang, Tianlin Li, Longfei Li, Jing Wang, and Quanshi Zhang. Knowledge consistency between neural networks and beyond. In *International Conference on Learning Representations*, 2019.

[77] Thomas Liao, Rohan Taori, Deborah Raji, and Ludwig Schmidt. Are we learning yet? a meta review of evaluation failures across machine learning. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL `https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/757b505cfd34c64c85ca5b5690ee5293-Paper-round2.pdf`.

[78] Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. Limitations of autoregressive models and their alternatives. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5147–5173, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.405. URL `https://aclanthology.org/2021.naacl-main.405`.

[79] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

[80] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pages 1–8, 2023.

[81] Claire Marks, Alissa M Hummer, Mark Chin, and Charlotte M Deane. Humanization of antibodies using a machine learning approach on large-scale repertoire data. *Bioinformatics*, 37(22):4041–4047, 2021.

[82] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[83] R Thomas McCoy, Junghyun Min, and Tal Linzen. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, 2020.

[84] R Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*, 2023.

[85] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

[86] Roni Miah, Ayesha Siddiqa, Udvashita Chakraborty, Jamsheda Ferdous Tuli, Noyon Kumar Barman, Aukhil Uddin, Tareque Aziz, Nadim Sharif, Shuvra Kanti Dey, Mamoru Yamada, et al. Development of high temperature simultaneous saccharification and fermentation by thermosensitive saccharomyces cerevisiae and bacillus amyloliquefaciens. *Scientific Reports*, 12(1):3630, 2022.

[87] H Pezeshgi Modarres, MR Mofrad, and AJRA Sanati-Nezhad. Protein thermostability engineering. *RSC advances*, 6(116):115252–115270, 2016.

[88] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, pages 5727–5736, 2018.

[89] Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, 2018.

[90] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[91] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

[92] Pascal Notin, Aaron W Kollasch, Daniel Ritter, Lood Van Niekerk, Steffan Paul, Han Spinner, Nathan J Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[93] Pascal Notin, Ruben Weitzman, Debora Susan Marks, and Yarin Gal. Proteinnpt: Improving protein property prediction and design with non-parametric transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[94] Mimi Onuoha. The point of collection. *Data & Society: Points*, 2016.

[95] Pedro A Ortega, Markus Kunesch, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Joel Veness, Jonas Buchli, Jonas Degrave, Bilal Piot, Julien Perolat, et al. Shaking the foundations: delusions in sequence models for interaction and control. *arXiv preprint arXiv:2110.10819*, 2021.

[96] Frank Pasquale. *The black box society*. Harvard University Press, 2015.

[97] Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *arXiv preprint arXiv:2012.05345*, 2020.

[98] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[99] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proc. 14th SIGKDD*. ACM, 2008.

[100] Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*, 2018.

[101] Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? *arXiv preprint arXiv:2006.16923*, 2020.

[102] Thomas Pumir, Amit Singer, and Nicolas Boumal. The generalized orthogonal procrustes problem in the high noise regime. *Information and Inference: A Journal of the IMA*, Jan 2021. ISSN 2049-8772. doi: 10.1093/imaiai/iaaa035. URL `http://dx.doi.org/10.1093/imaiai/iaaa035`.

[103] Ruchir Puri. Mitigating bias in artificial intelligence (ai) models – ibm research, Feb 2019. URL `https://www.ibm.com/blogs/research/2018/02/mitigating-bias-ai-models/`.

[104] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2019.

[105] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pages 6076–6085, 2017.

[106] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[107] Esther Rolf, Theodora T Worledge, Benjamin Recht, and Michael Jordan. Representation matters: Assessing the importance of subgroup allocations in training data. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9040–9051. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/rolf21a.html`.

[108] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[109] Hee Jung Ryu, Hartwig Adam, and Margaret Mitchell. Inclusivefacenet: Improving face attribute detection with race and gender diversity. *arXiv preprint arXiv:1712.00193*, 2017.

[110] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[111] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and HV Jagadish. Representation bias in data: A survey on identification and resolution techniques. *ACM Computing Surveys*, 2023.

[112] Ada Y Shaw, Hansen B Spinner, Sarah Gurev, Jung-Eun Shin, Nathan Rollins, and Debora S Marks. Removing bias in sequence models of protein fitness. *bioRxiv*, pages 2023–09, 2023.

[113] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):2403, 2021.

[114] Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax, 2017.

[115] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[116] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[117] Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah Goodman. Investigating transferability in pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1393–1401, 2020.

[118] Rohan Taori and Tatsunori Hashimoto. Data feedback loops: Model-driven amplification of dataset biases. In *International Conference on Machine Learning*, pages 33883–33920. PMLR, 2023.

[119] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SJzSgnRcKX`.

[120] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[121] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

[122] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019.

[123] Robert Verkuil, Ori Kabeli, Yilun Du, Basile IM Wicky, Lukas F Milles, Justas Dauparas, David Baker, Sergey Ovchinnikov, Tom Sercu, and Alexander Rives. Language models generalize beyond natural proteins. *bioRxiv*, pages 2022–12, 2022.

[124] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.

[125] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[126] Liwei Wang, Lunjia Hu, Jiayuan Gu, Zhiqiang Hu, Yue Wu, Kun He, and John Hopcroft. Towards understanding learning representations: To what extent do different neural networks learn the same representation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 9584–9593. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/5fc34ed307aac159a30d81181c99847e-Paper.pdf`.

[127] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`.

[128] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 547–558, 2020.

[129] Kevin K Yang, Nicolo Fusi, and Alex X Lu. Convolutions are competitive with transformers for protein sequence pretraining. *bioRxiv*, pages 2022–05, 2022.

[130] Richard Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, pages III–325, 2013.

[131] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1323. URL `https://aclanthology.org/D17-1323`.

[132] Ruiqi Zhong, Dhruba Ghosh, Dan Klein, and Jacob Steinhardt. Are larger pretrained language models uniformly better? comparing performance at the instance level. *arXiv preprint arXiv:2105.06020*, 2021.

[133] Danqing Zhu, David H. Brookes, Akosua Busia, Ana Carneiro, Clara Fannjiang, Galina Popova, David Shin, Kevin C. Donohue, Li F. Lin, Zachary M. Miller, Evan R. Williams, Edward F. Chang, Tomasz J. Nowakowski, Jennifer Listgarten, and David V. Schaffer. Optimal trade-off control in machine learning–based library design, with application to adeno-associated virus (aav) for gene therapy. *Science Advances*, 10(4):eadj3786, 2024. doi: 10.1126/sciadv.adj3786. URL `https://www.science.org/doi/abs/10.1126/sciadv.adj3786`.