Validity Challenges in Machine Learning Benchmarks

John Miller

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-180
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-180.html

August 3, 2022

Validity Challenges in Machine Learning Benchmarks

by

John Patterson Miller

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Moritz Hardt, Co-chair
Professor Benjamin Recht, Co-chair
Professor Joshua Blumenstock

Summer 2022

Validity Challenges in Machine Learning Benchmarks

Abstract

Validity Challenges in Machine Learning Benchmarks

by

John Patterson Miller

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Moritz Hardt, Co-chair

Professor Benjamin Recht, Co-chair

Over the last decade, machine learning practitioners in fields like computer vision and natural language processing have devoted vast resources to building models that successively improve performance numbers on a small number of prominent benchmarks. While performance on these benchmarks has steadily increased, real-world deployments of learning systems continue to encounter difficulties with robustness and reliability. The contrast between the optimistic picture of progress painted by benchmark results and the challenges encountered by real systems calls into question the *validity* of benchmark datasets, that is, the extent to which benchmark findings generalize to new settings. In this thesis, we probe the validity of machine learning benchmarks from several perspectives.

We first consider the *statistical validity* of machine learning benchmarks. Folk-wisdom in machine learning says that repeatedly reusing the same dataset for evaluation invalidates standard statistical guarantees and can lead to overoptimistic estimates of performance. We test this hypothesis via a dataset reconstruction experiment for the Stanford Question Answering Dataset (SQuAD). We find no evidence of overfitting from test-set reuse. This result is consistent with a growing literature which finds no evidence of so-called adaptive overfitting in benchmarks using image and tabular data. We offer a new explanation for this phenomenon based on the observed similarity between models being evaluated, and we formally show this type of model similarity offers improved protection against overfitting.

While statistical validity appears to be less of a concern, our experiments on SQuAD reveal that predictive performance estimates are extremely sensitive to small changes in the distribution of test examples, which threatens the *external validity* of such benchmarks. To understand the breadth of this issue, we conduct a large-scale empirical study of more 100,000 models across 60 different distribution shifts in computer vision and natural language processing. Across these many distribution shifts, we observe a common phenomenon:

small changes in the data distribution lead to large and uniform performance drops across models. Moreover, this drop is often governed by a precise linear relationship between the performance on the benchmark and performance on new data that holds across model architectures, training procedures, and dataset size. Consequently, sensitivity to distribution shift is likely an intrinsic property of existing benchmark datasets and not something that is easily addressed by algorithmic or modeling innovations.

Taken together, these results highlight the difficulties with using narrow, static benchmarks to build and evaluate systems deployed in a dynamic world. In the final part of the thesis, we present two new resources to improve the evaluation of such systems. In the context of algorithmic fairness, we present a new collection of datasets derived from US Census data that explicitly includes data across multiple years and all US states. This allows researchers to evaluate new models and algorithms in presence population changes due to temporal shift and geographic variation. In the context of causal inference, we introduce a simulation framework that repurposes dynamical system models from climate science, economics, and epidemiology for the evaluation of causal inference tools across a variety of data generating distributions both when the assumptions of such tools are satisfied and when they are not.

To the Millerbillies

Contents

Acknowledgments

This thesis is the product of the support, mentorship, and friendship of many people who have helped me along my journey and without whom none of this work would have happened.

First, I'd like to thank my advisors Moritz Hardt and Benjamin Recht. Throughout my PhD, Moritz has been a wellspring of guidance and perspective on research, academia, and life. His wide-ranging interests repeatedly pushed me outside my intellectual comfort zone, and joining him on explorations into deep learning theory, causal inference, and statistical validity substantially broadened my horizons as a scholar. I'd like to Ben for opening his research group to me, teaching me the power of the linear model, and pushing me to eschew theoretical abstraction for grounded empirical work. Ludwig Schmidt deserves credit as a de facto third advisor during the later years of my PhD. I learned much from Ludwig's relentess work ethic and willingness to get involved with the most unglamorous and minute details of a project. His strong demands for rigorous empirical work have permantently changed how I approach science. Thanks also to Josh Blumenstock for serving on my committee and Rediet Abebbe for her advice and comments during the final year of my PhD.

The work that appears in this thesis would not have come together without the efforts and brilliance of many collaborators and co-authors. Thank you to Yair Carmon, Frances Ding, Anca Dragan, Sara Fridovich-Keil, Pang Wei Koh, Karl Krauth, Horia Mania, Smitha Milli, Juanky Perdomo, Aditi Raghunathan, Rebecca Roelofs, Shiori Sagawa, Vaishaal Shankar, Yu Sun, Rohan Taori, and Tijana Zrnic.

The Berkeley community was more intellectually enriching and personally fulfilling than I could have possible expected. Thank you to my office mates and research group peers— Mihaela Curmei, Sarah Dean, Frances Ding, Chloe Hsu, Meena Jagadeesan, Lydia Liu, Smitha Milli, Juan Perdomo, Esther Rolf, Max Simchowitz, Yu Sun, Nilesh Tripuraneni, and Tijana Zrnic— for the many stimulating conversations, Free Speech Fridays, gym sessions, dinners, support, and commiseration during the last five years.

Before Berkeley, I'd like to thank Tim Roughgarden for giving me my first taste of computer science research as a CURIS student in the summer of 2014, Percy Liang for introducing me to machine learning and the countless hours he spent mentoring me as a naive undergraduate, and Kelvin Guu for his patience and generosity in teaching me how to do empirical work in machine learning.

Finally, I'd like to thank my family—the Millerbillies—for their love and support. My brother Bailey deserves special mention for the endless hours he spent listening to me stress about deadlines and gripe about academic life, as well as the GPU heat he endured as the caretaker of the group's GPU server. Thank you.

Chapter 1

Introduction

The benchmark dataset paradigm is central to modern machine learning practice. In computer vision, results on the ImageNet benchmark [62, 200] kicked off a decade of work and massive industrial investment in deep learning. In natural language processing, benchmark datasets like SQuAD [187] and GLUE [239] have garnered thousands of citations and are among the first testbeds for new ideas. In all areas of machine learning, it is now de rigueur for practitioners to compare the *quantitative performance* of algorithms on *common test sets*, using fixed and automatically calculated *evaluation metrics*. These benchmarks play a number of complex and important roles in the machine learning ecosystem. At a basic level, benchmarks provide data for model builders to train and compare models. More subtly, benchmark also implicitly formulate problems, organize research communities, drive funding and investment decisions, and serve as a ready interface between academic and industrial work [92].

A driving force behind the adoption of the benchmark dataset paradigm was the belief that focusing on benchmark evaluation would lead to progress in machine learning. The benchmark paradigm came into prominence in the 1980's under the influence of DARPA program manager Charles Wayne and Fred Jelinek as part of the common task framework. From the beginning, these men envisioned a focus on benchmark results would remove the element of "deceit-and-glamour" from evaluation and provide funding agencies with a concrete, numerical measure of progress over time [51, 139].

Over the last decade, Wayne and Jelinek's vision has been realized on a grand scale. Thousands of researchers build models for and compare results on an ever-growing set of new benchmarks in areas like computer vision [62, 200, 141, 123, 29, 75, 120, 14, 9, 50], natural language processing [187, 239, 71, 115, 242, 240], and robotics [33, 232, 40]. The field as a whole is increasingly empirical and evaluation-driven. New methods and techniques come equipped with few theoretical guarantees, and their ultimate justification is whether or not they improve performance numbers on a particular benchmark.

At first glance, this relentless focus on benchmarks has been an unmitigated success. In computer vision, since 2012, top-1 accuracy on ImageNet has risen dramatically from 63% [125] to 91% [255]. Similarly, in natural language processing, on the popular SQuAD

question answering benchmark, since 2016, F1 accuracy numbers have jumped from around 50 [187] to over 95 [252]. Collectively, the machine learning community is remarkably adept at hill-climbing on common benchmarks.

Despite the optimistic picture of progress portrayed in benchmark results, deployments of machine learning systems continue to fail in important applications. Learning systems to detect pneumonia or sepsis fail dramatically when trained on data from one hospital and used in another [256, 243]. Self-driving cars still have difficulty navigating new environments [206]. Speech recognition systems have significantly higher error rates on minority populations [119]. Models for home-price prediction can go wildly awry and cause nine-figure losses [219]. In essence, machine learning systems today are repeatedly deployed in high-stakes environments, and their performance underwhelms what one would expect from benchmark numbers alone.

In light of these conflicting observations, we revisit and interrogate the basic assumption of the benchmark paradigm: that progress on machine learning benchmarks leads to progress more generally. In this thesis, we probe the *validity* of machine learning benchmarks [156]. We first consider the *statistical validity* of benchmark results, that is, we test whether or not benchmark performance estimates reliably measure the performance of a model on the same distribution of examples as the benchmark test set. We then consider questions of *external validity*, that is, whether benchmark findings generalize to new experimental settings. Finally, we use insights from these experiments to construct new benchmarks and improve the state of empirical evaluation in algorithmic fairness and causal inference.

1.1 Statistical validity of benchmarks

Most machine learning benchmarks rely on the *holdout method* to evaluate models. Practitioners split their datasets into two parts: the training set used for model development and the "held-out"test set used for model evaluation. In principle, the test set is used only once and provides an unbiased estimate of the model's performance. In practice, the test set is repeatedly used as practitioners evaluate a model's performance on the test set and then use this feedback to tweak the model on the training set. This feedback invalidates the common statistical guarantees of the holdout method. Indeed, Hastie et al. [95] warn this practice can lead to overfitting and "the test set error of the final chosen model will underestimate the true test error, sometimes substantially." A primary worry about machine learning benchmarks is then that ever improving benchmark performance numbers are not real improvement, but rather the product of adaptive overfitting from test set reuse.

In Chapter 2, we test whether this type of adaptive overfitting has occurred on the popular SQuAD question answering benchmark [187]. To do this, we resurrect the original dataset construction pipeline, generate a new dataset with a distribution closely matching the original, and evaluate a testbed of models from the SQuAD leaderboard on the new dataset. Despite more than five years of heavy test set reuse, we find no evidence of adaptive overfitting.

Our findings on SQuAD are in line with similar dataset reproduction efforts in computer vision on ImageNet [190] and MNIST [249], as well as recent a meta-analysis of Kaggle competitions [195]. In Chapter 3, we seek to better understand the resilience of benchmarks to test set reuse. Concretely, we observe that many of the models evaluated on computer vision test sets like CIFAR-10 and ImageNet have extremely high similarity in their predictions, and we prove that this type of model similarity offers protection against overfitting.

1.2 External validity of benchmarks

Next, we turn our attention to the *external validity* of machine learning benchmarks. In particular, we study the extent to which the findings of a particular benchmark generalize to new populations and experimental contexts. To study this phenomenon rigorously in as many experimental settings as possible, we build new datasets and conduct large-scale evaluations of both classical and state-of-the-art deep learning models in natural language processing and computer vision.

In Chapter 2, using the same dataset construction pipeline, we generate three new tests for SQuAD using text data from different domains than the original Wikipedia articles and evaluate the ability of leaderboard models to generalize to new data. Across a broad range of models, we observe large and uniform performance drops. Interestingly, we observe an extremely precise *linear* relationship between model performance on the original test set and model performance on the new test set. Consequently, while model performance numbers themselves are brittle and unreliable, the relative ranking between models is often preserved.

In Chapter 4, we study this observed linear trend relationship more carefully. We conduct a large-scale evaluation of more than 100,000 models on more than 60 natural and synthetic distribution shifts on CIFAR-10 [123], ImageNet [63], the Stanford Wilds Benchmark [120], and a synthetic data testbed based on the YCB-Objects [40], among others. In most cases, we find both sensitivity to distribution shifts and the associated linear trend relationship holds across model architectures, model hyperparameters, training dataset sizes, optimization algorithms, and robustness interventions. The regularity of this phenomenon leads us to conjecture the linear trend is intimately related to properties of the benchmark datasets themselves and performance drops out of distribution are unlikely to be addressed with purely new algorithmic or modeling innovations.

Beyond distribution shifts of the types explored above, there are other challenges to external validity. The UCI Adult dataset [121] is one of the most widely used benchmarks in algorithmic fairness. In Chapter 5, we demonstrate that idiosyncratic details of dataset construction and choice of target variable limit it's utility for fairness research. Indeed, a standard logistic regression model has higher accuracy on the black and female subpopulations than the general population overall, which is a rather peculiar situation for fairness research.

Overall, one the key results of our inquiry is that while the reported results of machine learning benchmarks are typically statistically valid, benchmarks themselves face significant

challenges with external validity.

1.3 New resources for model evaluation

A primary source of external validity issues is that benchmarks themselves are often narrowly scoped and test only a limited range of experimental conditions. In the final part of this thesis, we attempt to address this issue by introducing two new experimental resources for evaluation. In Chapter 5, we build new datasets for algorithmic fairness from US Census data that rectify the issues we identified with UCI Adult and explicitly include temporal and geographic variation in populations. Finally, in Chapter 6, we introduce a new experimental framework that repurposes simulation environments from fields like economics and epidemiology to stress test the performance of causal inference tools in new environments and probe their robustness or lack thereof to violations of their key assumptions.

Chapter 2

Benchmark Case Study: The Stanford Question Answering Dataset

2.1 Introduction

In this chapter, we use the Stanford Question Answering Dataset (SQuAD) [187] as representative case study for validity issues facing modern machine learning benchmarks. Since its release in 2016, SQuAD has generated intense interest from the natural language processing community. At first glance, this intense interest has lead to impressive results. The best performing models in 2020 [65, 252] have F1 scores that are more than 40 points higher than the baseline presented by Rajpurkar et al. [187]. At the same time, it remains unclear to what extent progress on these benchmark numbers is a reliable indicator of progress more broadly.

The goal of building question answering systems is not merely to obtain high scores on the SQuAD leaderboard, but rather to *generalize* to new examples beyond the SQuAD test set. However, the competition format of SQuAD puts pressure on the validity of leaderboard scores. It is well-known that repeatedly evaluating models on a held-out test set can give overly optimistic estimates of model performance, a phenomenon known as *adaptive overfitting* [72]. Moreover, the standard SQuAD evaluation only measures model performance on new examples *from the same distribution*, i.e., paragraphs derived from Wikipedia articles. Nevertheless, we often use and deploy systems in settings different from the one in which they were trained. While Jia and Liang [113] demonstrated that SQuAD models are not robust to *adversarial* distribution shifts, one might still hope that the models are more robust to *natural* distribution shifts, for instance changing from Wikipedia to newspaper articles.

This state of affairs raises two important questions:

Are SQuAD models overfit to the SQuAD test set?
Are SQuAD models robust to natural distribution shifts?

In this work, we address both questions by replicating the SQuAD dataset creation

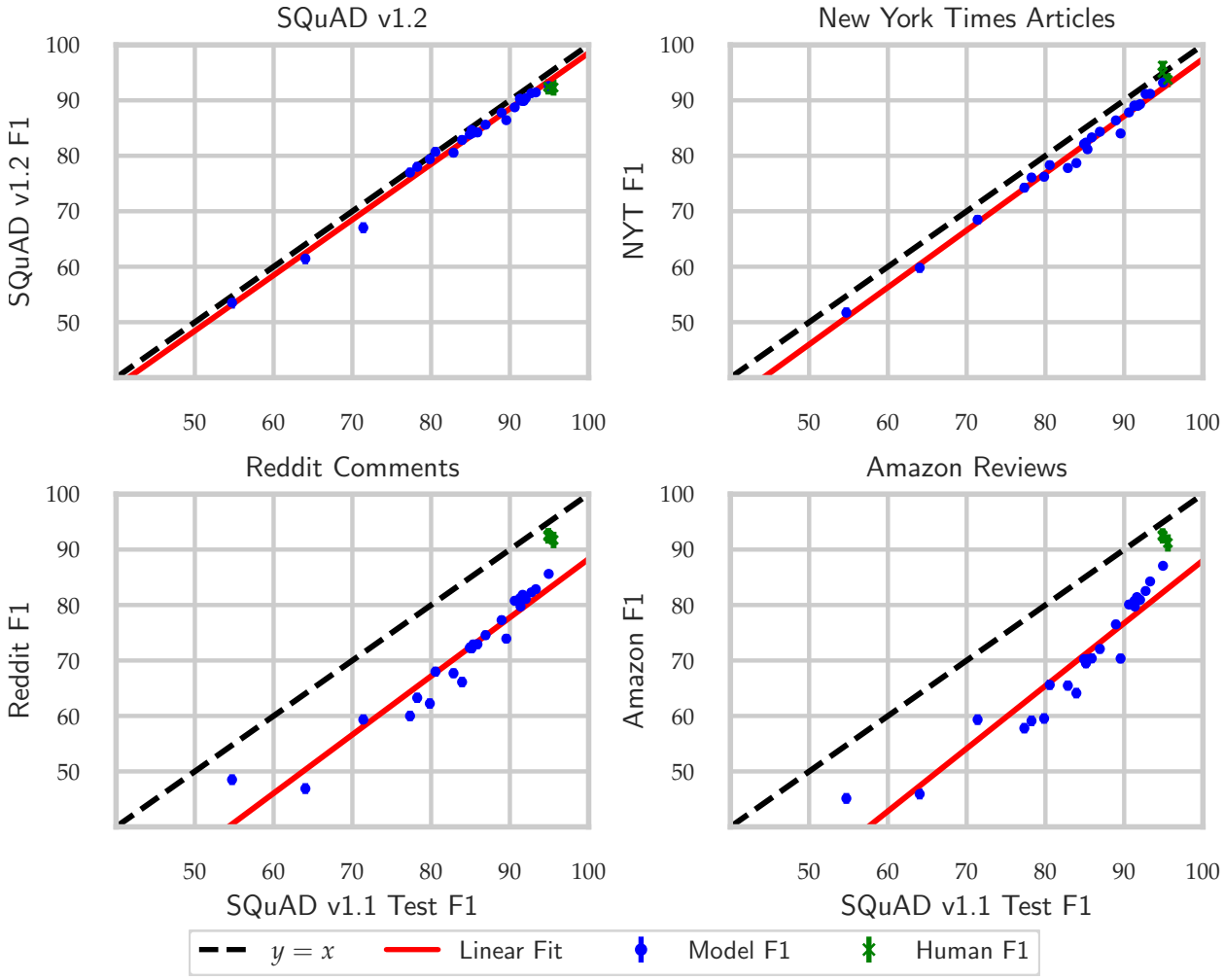


Figure 2.1: Model and human F1 scores on the original SQuAD v1.1 test set compared to our new test sets. Each point corresponds to a model evaluation, shown with 95% Student's-t confidence intervals (mostly covered by the point markers). The plots reveal three main phenomena: (i) There is no evidence of adaptive overfitting on SQuAD, (ii) all of the models suffer F1 drops on the new datasets, with the magnitude of the drop strongly depending on the corpus, and (iii) humans are substantially more robust to natural distribution shifts than the models. The slopes of the linear fits are 1.0, 1.03, 1.06, and 1.13, respectively. This means that every point of F1 improvement on the original dataset translates into roughly 1 point of improvement on our new datasets.

process and generating four new SQuAD test sets on both the original Wikipedia domain, as well as three new domains: New York Times articles, Amazon product reviews, and Reddit posts.

We first show that there is no evidence of adaptive overfitting on SQuAD. Across a large

collection of SQuAD models, there is little to no difference between the F1 scores from the original SQuAD test set and our replication. This even holds when comparing scores from the SQuAD *development* set (which was publicly released with answers) to our new test set. The lack of adaptive overfitting is consistent with recent replication studies in the context of image classification [190, 249]. These studies leave open the possibility that this phenomenon is specific to the data or models typical in computer vision research benchmarks. Our result demonstrates this same phenomenon also holds for natural language processing.

Beyond adaptive overfitting, we also demonstrate that SQuAD models exhibit robustness to some of our natural distribution shifts, though they still suffer substantial performance degradation on others. On the New York Times dataset, models in our test bed on average drop 3.0 F1 points. On the Reddit and Amazon datasets, the drop is on average 12.6 and 14.0 F1 points, respectively. All of our datasets were collected using the same data generation pipeline, so this degradation can be attributed purely to changes in the source text rather than differences in the annotation procedures across datasets.

We complement each of these experiments with a strong human baseline comprised of the author of this chapter and two collaborators. On the original SQuAD data, our human accuracy numbers are on par with the best SQuAD models [252] and significantly better than the Mechanical Turk baseline reported by Rajpurkar et al. [187]. On each of the test sets with distribution shift, human performance is essentially unchanged and significantly higher than the best SQuAD models.

Figure 2.1 summarizes the main results of our experiments. Humans show consistent behavior on all four test sets, while models are substantially less robust against two of the distribution shifts. Although there has been steady progress on the SQuAD leaderboard, there has been markedly less progress in this robustness dimension.

To enable future research, all of our new tests sets are freely available online.[1]

2.2 Background

In this section, we briefly introduce the SQuAD dataset and present a formal model for reasoning about performance drops between our test sets.

Stanford Question Answering Dataset

SQuAD is an extractive question answering dataset introduced by Rajpurkar et al. [187]. An example in SQuAD consists of a passage of text, a question, and one or more spans of text within the passage that answer the question. An example is given in Figure 2.2.

Model performance is evaluated using one of two metrics: exact match (EM) or F1. Exact match measures the percentage of predictions that exactly match at least one of the ground truth answers. F1 measures the maximum overlap between the tokens in the predicted span

[1] https://modestyachts.github.io/squadshifts-website/

Passage: "In our neighborhood, we were the small family, at least among the Irish and Italians… We could almost field a full baseball team. But the Flynns, they could put an entire football lineup… We loved Robert F. Kennedy's family: 11 kids, and Ethel looks great. Bobby himself was the seventh of nine."

Question: How many kids did Robert F. Kennedy have?
Answer: 11

Question: The author believes his family could fill a team of which sport?
Answer: baseball

Figure 2.2: Question and answer pairs from a sample passage in our New York Times SQuAD test set. Answers are text spans from the passage that answer the question.

and any of the ground truth answers, treating both the prediction and each answer as a bag of words. Both metrics are described formally in Appendix 2.8.

After releasing the original SQuAD v1.1 dataset, Rajpurkar, Jia, and Liang [186] introduced a new variant of the dataset, SQuAD 2.0, that includes unanswerable questions. Since SQuAD v1.1 has been public for longer and potentially subjected to more adaptivity, we focus on SQuAD v1.1 and refer to it as the SQuAD dataset throughout this chapter. The SQuAD v1.1 test set is not publically available. Therefore, while we use public test set evaluation numbers, we otherwise use the public SQuAD v1.1 development set for analysis.

A Model for Generalization

Although progress on SQuAD is measured through performance on a held-out test set, the implicit goal is not to achieve high F1 scores on the test set, but rather to *generalize* to unseen examples. Our experiments test the extent to which this assumption holds—if models with high leaderboard scores on the test set continue to perform well on new examples, whether from the same or different distributions.

To be more formal, suppose the original test set S is sampled from some underlying distribution $\mathcal{D}$, and consider a model f submitted to the SQuAD leaderboard. Let $L_S(f)$ denote the empirical loss of model f on the sample S, and let $L_{\mathcal{D}}(f)$ denote the corresponding population loss. In our experiment, we gather a new dataset of examples S' from a distribution $\mathcal{D}'$, potentially different from $\mathcal{D}$. We wish for the loss on the new sample, $L_{S'}(f)$ to be close

to the original, $L_S(f)$. Omitting f, we can decompose this gap into three terms [190].

$$L_S - L_{S'} = \underbrace{(L_S - L_{\mathcal{D}})}_{\text{Adaptivity gap}} + \underbrace{(L_{\mathcal{D}} - L_{\mathcal{D}'})}_{\text{Distribution gap}} + \underbrace{(L_{\mathcal{D}'} - L_{S'})}_{\text{Generalization gap}}$$

The *adaptivity gap* $L_S - L_{\mathcal{D}}$ measures how much adapting the model to the held-out test set S biases the estimate of the population loss. Since recent models are in part chosen on the basis of past test set information, the model f is not independent of S. Hence $L_S(f)$ can underestimate $L_{\mathcal{D}}(f)$, a phenomenon called *adaptive overfitting.* The *distribution gap* measures how much changing the distribution from $\mathcal{D}$ to $\mathcal{D}'$ affects the model's performance. Finally, the *generalization gap* $L_{S'} - L_{\mathcal{D}'}$ captures the difference between the sample and the population losses due to random sampling of S'. Since S' is sampled independently of the model, this gap is typically small and well-controlled by standard concentration results.

In the sequel, we empirically measure both the adaptivity gap and the distribution gap for a wide range of SQuAD models by collecting new test sets from a variety of distributions $\mathcal{D}'$. We first review related work that motivates our choice of SQuAD and natural distribution shifts.

2.3 Related work

Adaptive data analysis. Although repeated test-set reuse puts pressure on the statistical guarantees of the holdout method [72], a series of replication studies established there is no adaptive overfitting on popular classification benchmarks like MNIST [249], CIFAR-10 [190], and ImageNet [190]. Furthermore, Roelofs et al. [195] also found little to no evidence of adaptive overfitting in a host of classification competitions on the Kaggle platform. These studies either concern image classification or smaller competitions that have not been subject to intense, multi-year community scrutiny. Our work establishes similar results for natural language processing on a heavily studied benchmark.

A number of works have proffered explanations for why adaptive overfitting does not occur in the standard machine learning workflow [26, 153, 76, 264]. Complementary to these results, our work provides a new data point with which to validate and deepen our conceptual understanding of overfitting.

Datasets for question answering. Beyond SQuAD, a number of works have proposed datasets for question answering [193, 19, 115, 236, 71, 251, 127]. We focus our analysis on SQuAD for two reasons. First, SQuAD has been the focus of intense research for almost four years, and the competitive nature of the leaderboard format makes it an excellent example to study adaptive overfitting in natural language processing. Second, SQuAD requires all submissions to be uploaded to CodaLab [54], which ensures reproducibility and makes it possible to evaluate every submission on our new datasets using the same configuration and environment as the original evaluation.

Generalization in question answering. Given the plethora of question-answering datasets, Yogatama et al. [253] and Talmor and Berant [226] evaluate the extent to which models trained on SQuAD generalize to other question-answering datasets. In a similar vein, Fisch et al. [78] conduct a shared task competition that evaluates how well models trained on a collection of six datasets generalize to unseen datasets at test time. In both cases, the datasets encountered at test time vary across a number of dimensions: the question collection procedure, the origin of the input text, the question answering interface, the crowd worker population, etc. These differences are *confounding factors* that make it difficult to interpret performance differences across datasets. For example, human performance differs by 10 F1 points between SQuAD v1.1 and NewsQA [236]. In contrast, our datasets focus on a single factor of variation—the input text corpus. In this controlled setting, free of confounding factors, we observe non-trivial F1 drops across a large collection of models, while human F1 scores are essentially constant.

From a different perspective, both Jia and Liang [113] and Ribeiro, Singh, and Guestrin [192] consider robustness to *adversarial* dataset corruptions. While we instead focus on *natural* distribution shifts, we also evaluate adversarial distribution shift for our model testbed in Appendix 2.9. We find that robustness under natural and adversarial distribution shifts exhibits qualitatively different phenomena, making natural distribution shifts interesting to study in their own right.

2.4 Collecting new test sets

In this section, we describe our data collection methodology. Data collection primarily proceeds in two stages: curating passages from a text corpus and crowdsourcing question-answer pairs over the passages. In both of these stages, we take great care to replicate the original SQuAD data generating process. Where possible, we obtained and used the original SQuAD generation code kindly provided by Rajpurkar et al. [187]. We ran our dataset creation pipeline on four different corpora: Wikipedia articles, New York Times articles, Reddit posts, and Amazon product reviews.

Passage Curation

The first step in the dataset generation process is selecting the articles from which the passages or contexts are drawn.

Wikipedia. We sampled 48 articles uniformly at random from the same list of 10,000 Wikipedia articles as Rajpurkar et al. [187], ensuring that there is no overlap between our articles and those in the SQuAD v1.1 training or development sets. To minimize distribution shift due to temporal language variation, we extracted the text of the Wikipedia articles from around the publication date of the SQuAD v1.0 dataset (June 16, 2016). For each article, we extracted individual paragraphs and stripped out images, figures, and tables using the same

Table 2.1: Dataset statistics of our four new test sets compared to the original SQuAD 1.1 development and test sets.

Dataset	Total Articles	Total Examples
SQuAD v1.1 Dev	48	10,570
SQuAD v1.1 Test	46	9,533
New Wikipedia	48	7,938
New York Times	797	10,065
Reddit	1969	9,803
Amazon	1909	9,885

data processing code as Rajpurkar et al. [187]. Then, we subsampled the resulting paragraphs to match the passage length statistics of the original SQuAD dataset.[2] See Appendix 2.11 for a detailed comparison of the paragraph distribution of the original SQuAD dev set and our new SQuAD test set.

New York Times. We sampled New York Times articles from the set of all articles published in 2015 using the NYTimes Archive API. We scraped each article using the Wayback Machine, using the same snapshot timestamp as our Wikipedia dataset, and removing foreign language articles. Since the average paragraph length for NYT articles is significantly shorter than the average paragraph length for Wikipedia articles, we merged each NYT paragraph with its subsequent paragraph with some probability. Then we subsampled the merged paragraphs to match the character length statistics of the original SQuAD v1.1 dataset.

Reddit Posts. We sampled Reddit posts from the set of all posts across all subreddits during the month of January 2016 in the Pushshift Reddit Corpus [13]. Then we restricted the set of posts to those marked as "safe for work" and manually removed inappropriate posts from the remaining ones. We concatenated each post's title with its body, removed Markdown, and replaced all links with the string `LINKREMOVED`. We then subsampled the posts to match the passage length statistics of the original SQuAD v1.1 dataset.

Amazon Product Reviews. We sampled Amazon product reviews belonging to the "Home and Kitchen" category from the dataset released by McAuley et al. [157]. As in the previous datasets, we then subsampled the reviews to match the passage length statistics of SQuAD v1.1.

[2]The minimum 500 character per paragraph rule mentioned in Rajpurkar et al. [187] was adopted midway through their data collection, and hence the original dataset also includes shorter paragraphs [185].

Crowdsourcing Question-Answer Pairs

We employed crowdworkers on Amazon Mechanical Turk (MTurk) to ask and answer questions on the passages in each dataset. We followed a nearly identical protocol to the original SQuAD dataset creation process. We used the same MTurk user interface, task instructions, MTurk worker qualifications, time per task, and hourly rate (adjusted for inflation) as Rajpurkar et al. [187]. For full details and examples of the user interface, refer to Appendix 2.11.

For each paragraph, one crowdworker first asked and answered up to five questions on the content of the paragraph. Then, we obtained at least two additional answers for each question using separate crowdworkers.

There are two points of discrepancy between our crowdsourcing protocol and the one used to create the original SQuAD dataset. First, we interfaced directly with MTurk rather than via the Daemo platform.[3] Second, in the original SQuAD task, workers asked and answered questions for an entire article, whereas in our task workers asked and answered questions for at most five paragraphs at a time. Although each difference is a potential source of distribution shift, in Section 2.5 we show that the effect of these changes is negligible—all models achieve roughly the same scores on both the original and new Wikipedia datasets.

After gathering question and answer pairs for each paragraph, we apply the same post-processing and data cleaning as SQuAD v1.1. We adjusted answer whitespace for consistency, filtered malformed answers, and removed all documents that had less than an average of two questions per paragraph after filtering. Table 2.1 summarizes the overall statistics of our datasets.

Human Evaluation

Although both SQuAD and our new test sets have answers from MTurk workers, it is not clear whether these answers represent a compelling human baseline. At minimum, workers are not familiar with the typical style of answers in SQuAD (e.g., how much detail to include), and they receive no feedback on their performance. To obtain a stronger human baseline, the author and two collaborators also answered approximately 1,000 questions on each of the four new test sets and the original SQuAD development set, following the same procedure and using the same UI as the MTurk workers. To take feedback into account, each participant first labelled 500 practice examples from the training set and compared their answers with the ground truth.

2.5 Main results

We use the four new datasets generated in the previous part to test for adaptive overfitting on SQuAD and probe the robustness of SQuAD models to natural distribution shifts.

[3]The Daemo platform has been discontinued.

Table 2.2: Comparision of model F1 scores on the original SQuAD test set and our new Wikipedia test set. Rank refers to the relative ordering of the 25 models in our testbed using the original SQuAD v1.1 F1 scores, new rank refers to the ordering using the new Wikipedia test set scores, and Δ rank is the relative difference in ranking from the original test set to the new test set. The confidence intervals are 95% Student-T intervals. A complete table with data for the entire model testbed, references, and analogous data for EM scores is in Appendix 2.10.

New-Wiki F1 Score Summary					
Name	Rank	SQuAD	New-Wiki	New Rank	Δ Rank
XLNET-123	1	94.9	92.6 [92.1, 93.1]	1	0
Tuned BERT-1seq Large	2	93.3	91.4 [90.9, 92.0]	2	0
BERT-Large Baseline	3	92.7	91.3 [90.7, 91.8]	3	0
BiDAF+SelfAttn+ELMo	13	85.9	84.2 [83.5, 85.0]	14	-1
Jenga	18	82.8	80.6 [79.7, 81.4]	19	-1
AllenNLP BiDAF	22	77.3	77.0 [76.1, 77.9]	22	0

We evaluated a broad set of 25 models submitted to the SQuAD leaderboard, including state-of-the-art models like XLNet [252] and BERT [65], as well as older, but popular models like BiDAF [208]. All of the models were submitted to the CodaLab platform, and we evaluate every model using the exact same configuration (model weights, hyperparameters, command-line arguments, execution environment) as the original submission. Tables 2.2 and 2.3 contain a brief summary of the results for key models.

Adaptive Overfitting

The SQuAD models in our testbed come from a long sequence of papers that incrementally improve F1 and EM scores over a period of several years. Consequently, if there is adaptive overfitting, we should expect the later models to have large drops in F1 scores because they are the result of more interaction with the testset. In this case, the higher F1 scores are partially the result of a larger adaptivity gap, and we would expect that, as the observed scores L_S continue to rise, the population scores L_D would begin to plateau.

To check for adaptive overfitting, we plot the SQuAD v1.1 test F1 scores against F1 scores our new Wikipedia test set. Figure 2.1 in Section 2.1 provides strong evidence against the adaptive overfitting hypothesis. Across the entire model collection, the F1 scores on the new test set closely replicate the original F1 scores. The observed linear fit is in contrast to the concave curve one would expect from adaptive overfitting. We use 95% Student's-t confidence intervals, which make a large-sample Gaussian assumption, to capture the error in the new test F1 scores due to random variation. No such confidence intervals are available

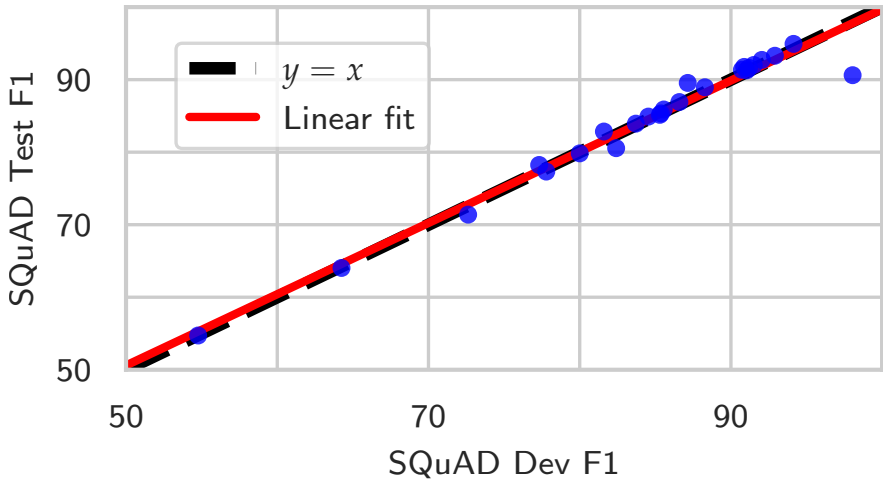


Figure 2.3: Comparison of F1 scores between the SQuAD v1.1 dev set and the SQuAD v1.1 test set. Despite heavy use of the dev set during model development, the dev set and test set scores closely match, with the exception of a single model that explicitly trained on the dev set [179]. The slope of the linear fit is 0.99.

for the original test set scores since the test set is not publicly available. A similar plot for EM is provided in Appendix 2.10.

Not only is there little evidence for adaptive overfitting on the test set, there is also little evidence of adaptive overfitting on the development set. In Figure 2.3, we plot F1 scores on the SQuAD v1.1 development set against F1 scores on the SQuAD v1.1 test set. With the exception of a single model, the F1 scores on the dev set closely match the scores on the test set, despite the fact that the development set is aggressively used during model selection. Moreover, the model that does not lie on the linear trend line, `Common-sense Governed BERT-123`, was directly trained on the development set [179].

Robustness to Natural Distribution Shifts

Given the correspondence between the old and new Wikipedia test set F1 scores, the adaptivity gap and the distribution gap are small or non-existent. Consequently, there is minimal distribution shift stemming from our data generation pipeline. This allows us to probe the sensitivity of the SQuAD models to a set of controlled distribution shifts, namely the choice of text corpus. Since all of the datasets are constructed with the same preprocessing pipeline, crowd-worker population, and post-processing, the datasets are free of confounding factors that would otherwise arise when comparing model performance across different datasets.

Figure 2.1 in Section 2.1 shows F1 scores of on the SQuAD v1.1 test set versus the F1 scores on each of the new test sets for each model. All models experience an F1 drop on the new test sets, though the magnitude is both test-set dependent and smaller than might

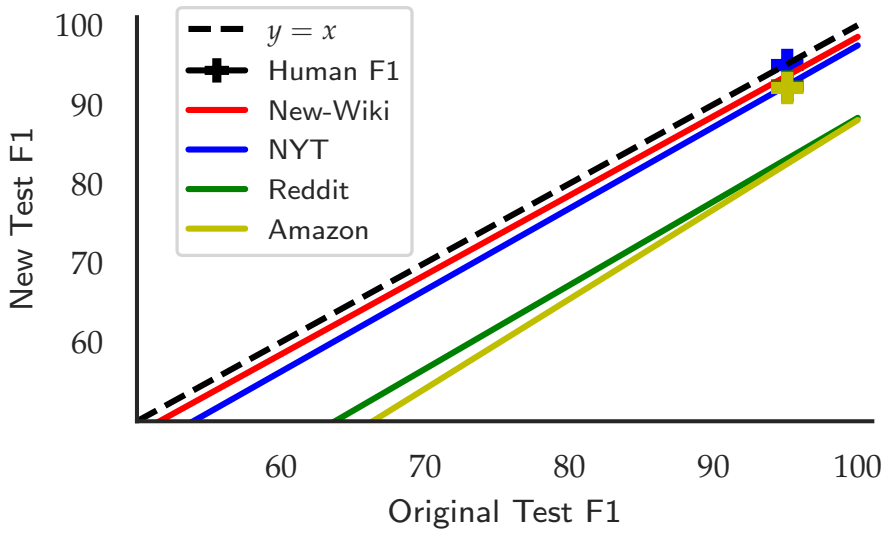


Figure 2.4: Comparison of the linear fits for each of our new test sets. Markers indicate average human performance on each dataset. By varying the dataset, we can vary the gap between perfect robustness and the observed linear fit and change the difficulty of the distribution shift problem. In each case, human performance is higher than the linear fit, suggesting a substantial difference in robustness.

Table 2.3: Comparision of model F1 scores on the original SQuAD test set and our new Amazon test set. Rank refers to the relative ordering of the 25 models in our testbed using the original SQuAD v1.1 F1 scores, new rank refers to the ordering using the Amazon test set scores, and Δ rank is the relative difference in ranking from the original test set to the new test set. The confidence intervals are 95% Student-T intervals. A complete table with data for the entire model testbed, the New York Times and Reddit datasets, and EM scores scores is in Appendix 2.10.

Amazon F1 Score Summary

Name	Rank	SQuAD	Amazon	New Rank	Δ Rank
XLNET-123	1	94.9	87.1 [86.5, 87.7]	1	0
Tuned BERT-1seq Large	2	93.3	84.3 [83.6, 84.9]	2	0
BERT-Large Baseline	3	92.7	82.5 [81.8, 83.2]	3	0
BiDAF+SelfAttention+ELMo	13	85.9	70.4 [69.5, 71.3]	12	1
Jenga	18	82.8	65.5 [64.6, 66.4]	18	0
AllenNLP BiDAF	22	77.3	57.8 [56.8, 58.7]	23	-1

otherwise be expected. On New York Times, for instance, the top performing XLNet only drops around 1.5 F1 points, whereas it drops around 8 F1 points on Amazon and 9 F1 points on Reddit. Table 2.3 summarizes the F1 scores for a select set of models. Full results for all models, datasets, and EM scores are given in Appendix 2.10.

F1 scores on the original SQuAD test set are highly predictive of F1 scores on the new test sets, and, interestingly, the relationship is well-captured by a linear fit even under distribution shifts. Similar to Recht et al. [190], we observe the linear fits are better under a probit scaling of F1 scores, see Appendix 2.10. The gap between perfect robustness, $y = x$, and the observed linear fits varies with the dataset in a way that roughly corresponds to problem difficulty: 3.0 F1 points for New York Times and 12.6 points for Reddit and 14.0 F1 for Amazon. This relationship is summarized in Figure 2.4. In Appendix 2.10, we further show the magnitude of the gap is controlled by another measure of task difficulty– agreement between human answers—and we can systematically vary the size of the gap by varying this measure of task specificity. In each case, higher performance on SQuAD v1.1 translates into higher performance on these *natural* distribution shift instances, in contrast to *adversarial* distribution shifts. We discuss this relation further in Appendix 2.9.

Despite the robustness demonstrated by the models, on all of the distribution test sets, human performance is substantially higher than model performance and well above the linear fits described in Figure 2.4. This rules out the possibility that the shift in F1 scores are entirely due to an increase in noise in the data. Moreover, it points towards substantial room for improvement for models on our new test sets.

2.6 Further analysis

In this section, we further explore the properties of our new test sets. We first study the extent to which common measures of dataset difficulty can explain the performance drops on our new test sets. Then, we evaluate whether training models with more, and more diverse, data improves robustness to our distribution shifts.

Are The New Test Sets Harder Than The Original?

One hypothesis for the performance drops observed in Section 2.5 is that our new dataset are harder in some sense. For instance, the diversity of answers may be greater among Reddit comments than Wikipedia articles. To better understand this, we compare the original SQuAD development set along with our four new test sets using the three difficulty measures introduced in Rajpurkar et al. [187].

Answer diversity. Following Rajpurkar et al. [187], we automatically categorize each answer into numerical and non-numerical answers, named entities, and constituents using Spacy [106] and the constituency parser from Kitaev and Klein [117]. Histograms of answer types for each data are shown in Figure 2.5. Since the original pipeline is not available,

our implementation differs slightly from Rajpurkar et al. [187] and we include results on the SQuAD dev set for comparison. Both the original and our new Wikipedia test set have very similar answer type histograms. The distribution shift datasets have slight variations in the answer distributions. For instance, NYT has more person answers, whereas Amazon has more adjective phrases. However, there is little that systematically explains the performance differences between the datasets. In Appendix 2.10, we show a simple model that predicts F1 scores on our new test sets using model F1 scores on each of the answer types and the relative frequency of each answer type in our new test sets explains little of the performance differences across test sets.

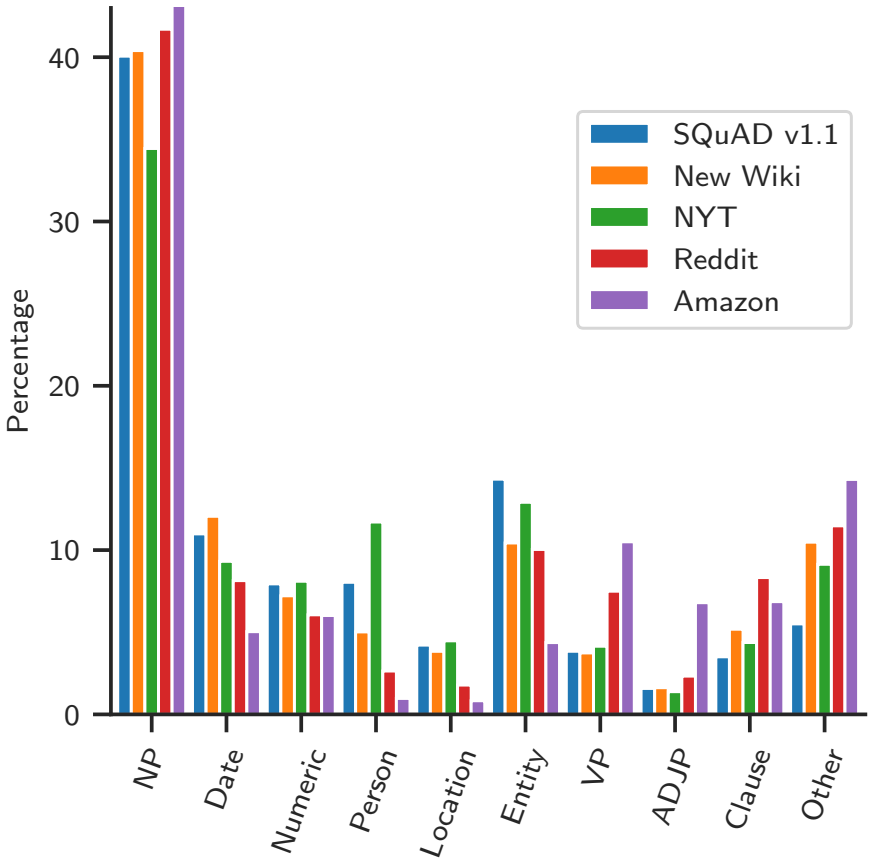


Figure 2.5: Comparison of the types of answers in the original and new datasets. We automatically partition our answers into the same categories as Rajpurkar et al. [187]. Although there are differences between the datasets, e.g. New York Times has more person answers, the four datasets are very similar, and there is little that systematically explains the performance drops.

Syntactic divergence. We also stratify our datasets using the automatic syntactic divergence measure of Rajpurkar et al. [187]. Syntactic divergence measures the similarity between the syntactic dependency tree structure of both the question and answer sentences and provides another metric of example difficulty. In Figure 2.6, we compare the histograms of syntactic divergence for the SQuAD dev set and our new test sets. All of the datasets have similar histograms, though both the Reddit and Amazon datasets have slightly more examples with small syntactic divergence. As Rajpurkar et al. [187] note, however, examples with small syntactic divergence are not necessarily easier if there are many other candidate answers with small divergence.

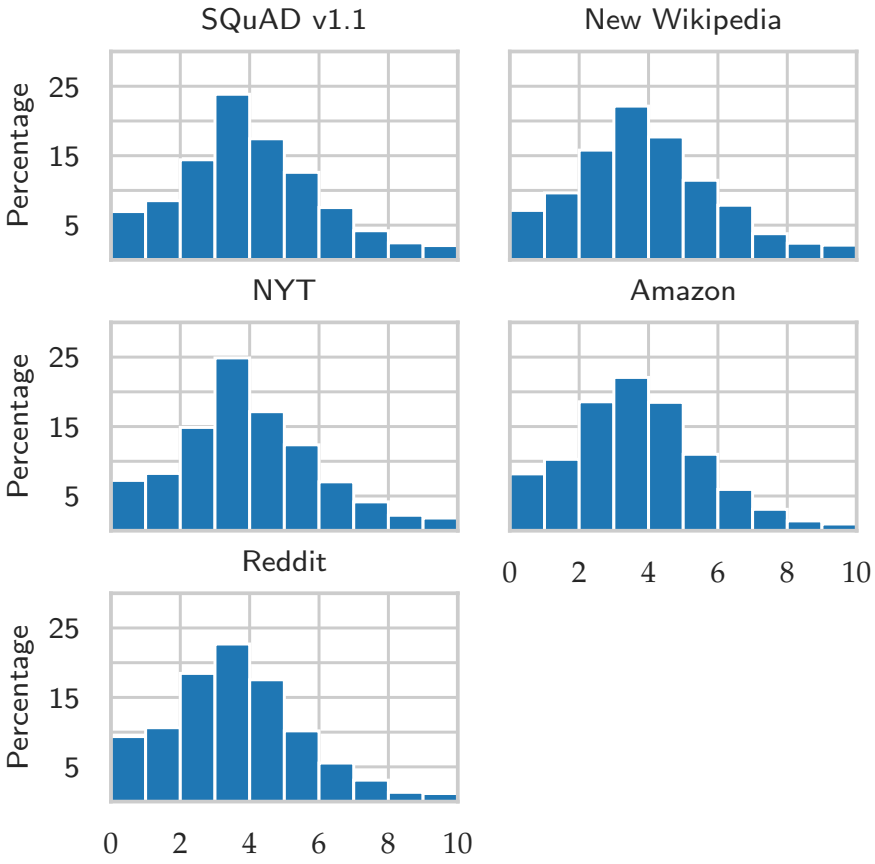


Figure 2.6: Histograms of syntactic divergence between question and answer sentences for both the original and new datasets. All of the datasets have a similar distribution of syntactic divergence, though the Reddit and Amazon datasets have more question-answers pairs with small (1-2) syntactic divergence.

Reasoning required. Finally, we compare our new test sets in terms of the reasoning required to answer each question-answer pair, using the same non-mutually exclusive cat-

egories as Rajpurkar et al. [187]. For each dataset, as well as the SQuAD dev set, we randomly sampled and manually labeled 192 examples. The results for each dataset are presented in Table 2.4. Both the Amazon and Reddit dataset have more examples requiring world knowledge to resolve lexical variation, while the New York Times dataset has more examples requiring multisentence reasoning.

Table 2.4: Manual comparison of the reasoning required to answer each question-answer pair on a random sample of 192 examples from each dataset using the categories from Rajpurkar et al. [187]. The Reddit and Amazon datasets have more examples requiring world knowledge to resolve lexical variation, whereas the New York Times dataset requires more multi-sentence reasoning.

Reasoning Type	SQuAD	New Wiki	NYT	Reddit	Amazon
Lexical Variation (Synonomy)	37.2	38.0	34.1	38.0	43.4
Lexical Variation (World Knowledge)	10.9	6.2	10.1	22.5	18.6
Syntactic Variation	49.6	46.5	50.4	55.0	50.4
Multiple Sentence Reasoning	12.4	11.6	17.1	12.4	10.1
Ambiguous	0.8	2.3	0.8	1.6	0.0

Are Models Trained with More Data More Robust to Natural Distribution Shifts?

High performance on our new datasets requires models to generalize to data distributions that may be different from those on which they were trained. Our primary evaluation only concerns the robustness on SQuAD models, and a natural follow-up is whether models trained on more data, or explicitly trained for out-of-distribution question-answering, perform better on our new test sets.

To test this claim, we evaluated a collection of models from the Machine Reading for Question Answering (MRQA) 2019 Shared Task on Generalization [78]. In the shared task, models were trained on 6 question-answering datasets, including SQuAD v1.1, and then evaluated on 12 held-out datasets. The datasets simultaneously differed not just in the passage distribution, as in our experiments, but also in confounders like the question distribution, and the relationship between questions and passages.

In Figure 2.7, we plot the F1 scores of MRQA models on the SQuAD v1.1 dataset against the F1 scores on each of our new test sets, along with the linear fits from Figure 2.1. On the Reddit and Amazon test sets, the best MRQA model in our testbed, Delphi [146], achieves higher F1 scores than any SQuAD model and is substantially above the linear fit. However, many of the models trained on more data exhibit no improved robustness. In addition, all of the models are still substantially below the human F1 scores and robustness.

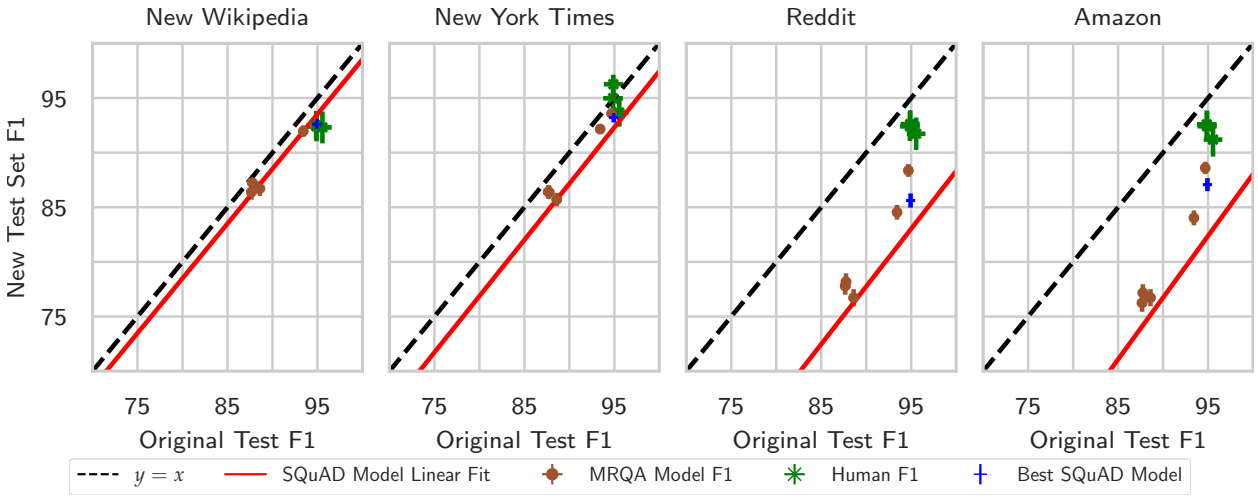


Figure 2.7: Model from the MRQA Shared Task 2019, trained on 5 datasets beyond SQuAD, and human F1 scores on the original SQuAD test set and each of our new test sets. The error bars are 95% Student's-t confidence intervals. Although the MRQA models still lag human performance and robustness across datasets, these models, particularly those with high F1 scores on the original SQuAD, exhibit increased robustness and generalization across each of the datasets compared to models that are only trained on SQuAD.

2.7 Discussion

Despite years of test set reuse, we find no evidence of adaptive overfitting on SQuAD. Though limits on the rate of resubmission and the bit precision of test set scores may partially explain the test set results, such explanations do not apply to the development set. Our findings demonstrate natural language processing benchmarks like SQuAD continue to support progress much longer than than reasoning from first principles might have suggested.

While SQuAD models generalize surprisingly well to new examples from the same distribution, results on our three natural distribution shift datasets suggest robustness to distribution shifts remains a problem. On each of our new test sets, a strong human baseline is unchanged, but SQuAD models suffer non-trivial and nearly uniform performance drops. While question answering models have made substantial progress on SQuAD, there has been much less progress towards closing the robustness gap under non-adversarial distribution shifts. This highlights the need to move beyond model evaluation in the standard, i.i.d. setting, and explicitly incorporate distribution shifts into evaluation. We hope our new test sets offer a helpful starting point.

2.8 Appendix: Evaluation metrics

In this section, we formally define the evaluation metrics used throughout our experiments. Let $(p, q, (a^1, \ldots, a^n)$ denote a passage p, a question q, and a set of n answers $(a^1, \ldots, a^n)$. Let S denote the sampled dataset, let f denote some model, and $f(p, q) = \hat{a}$ be its predicted answer.

F1 Score. F1 measures the average overlap between the prediction and the ground-truth answer. Given answer a and prediction $\hat{a}$, consider a and $\hat{a}$ as bags of words (sets), and let $v(a, \hat{a})$ be their associated F1 score, i.e. the harmonic mean of precision and recall between the two sets. Then,

$$\text{F1}(f) = \frac{1}{|S|} \sum_{(p,q,(a^1,\ldots,a^n))\in S} \max_{i=1,\ldots,n} v(a^i, f(p, q)).$$

Exact match. Exact match measures the percentage of predictions that exactly match any one of the ground truth answers.

$$\text{ExactMatch}(f) = \frac{1}{|S|} \sum_{(p,q,(a^1,\ldots,a^n))\in S} \max_{i=1,\ldots,n} \mathbb{1}\{f(p, q) = a^i\}.$$

All of our results are reported using the evaluation script provided by Rajpurkar et al. [187], which ignores punctation and the articles "a", "an", and "the" when computing the above metrics.

2.9 Appendix: Comparing natural and adversarial distribution shifts

To contrast natural and adversarial distribution shifts, we evaluated all of the models in our testbed against the adversarial attacks described in Jia and Liang [113] on the original SQuAD v1.1 dataset.

AddSent. In the `AddSent` attack, for every passage, question, and answer pair (p, q, a), Jia and Liang [113] procedurally generate up to five new sentences to append to the passage p that do not contradict the correct answer. Each of the sentences are generated to be similar to the correct answer, and ungrammatical or contradictory sentences are removed by crowdworkers. This results in a set of new examples $(\tilde{p}_1, q, a), \ldots, (\tilde{p}_5, q, a)$ for each original example. The adversary evaluates the model f on each of the 5 examples and picks the one that gives the lowest score, $\min_{i=1,\ldots,5} s(f(\tilde{p}_i, q), a)$, where s is the scoring function (exact

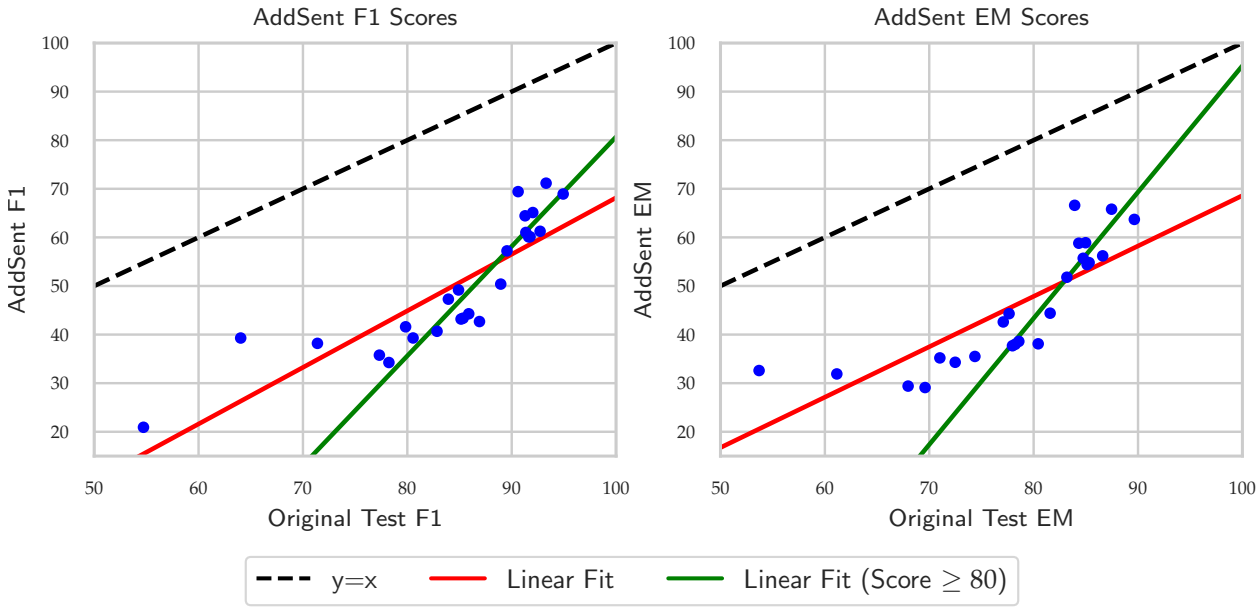


Figure 2.8: Comparison of F1 and EM scores on the original SQuAD test set versus the *adversarial* `AddSent` attack from Jia and Liang [113]. In contrast to *natural* distribution shifts, the trend is no longer clearly linear, and models with the same F1 scores on SQuAD v1.1 can have substantially different robustness properties. For F1 scores, the slope of the linear fit is 1.28, and, for models with F1 scores greater than 80, the slope of the linear fit is 2.27. For EM, the slopes are 1.04 and 2.6.

match or F1). In Figure 2.8, we compare F1 and EM scores on the original SQuAD v1.1 test set with F1 and EM scores against the adversarial `AddSent` attack.

In contrast to our natural distribution shift examples, we observe qualitatively different phenomenon. A linear fit no longer well captures the relationship between original test F1 scores and adversarial test F1 scores. For instance, a collection of models with 90 F1 scores on the original test distribution have a spread of 10 F1 points on the adversarial distributions. Furthermore, improvements on the original test set need not translate to improved robustness. Models between 60 and 80 F1 on SQuAD v1.1. have the same 40 F1 under `AddSent`. Moreover, the models with F1 scores greater than 80 have a linear fit with slope 2.27, suggesting that each 1 points F1 improvement on the original SQuAD v1.1. dataset translates into 2.27 point of F1 in terms of adversarial loss. This slope is significantly higher than those observed in our new test sets.

AddOneSent. The `AddOneSent` attack similar to the `AddSent` attack. However, rather than take the worst of the 5 altered passages, it randomly selects one of the five on which to evaluate the model. In Figure 2.9, we compare F1 and EM scores on the original SQuAD v1.1 test set with F1 and EM scores against the adversarial `AddSent` attack. Since this attack

does not require model access or evaluations, it is closer in spirit to the natural distribution shifts we consider. However, we observe much the same phenomenon as we see with `AddSent`. The linear model is no longer a good fit, and there is a threshold phenomenon around 80 F1 and EM whereby the slope sharply increases.

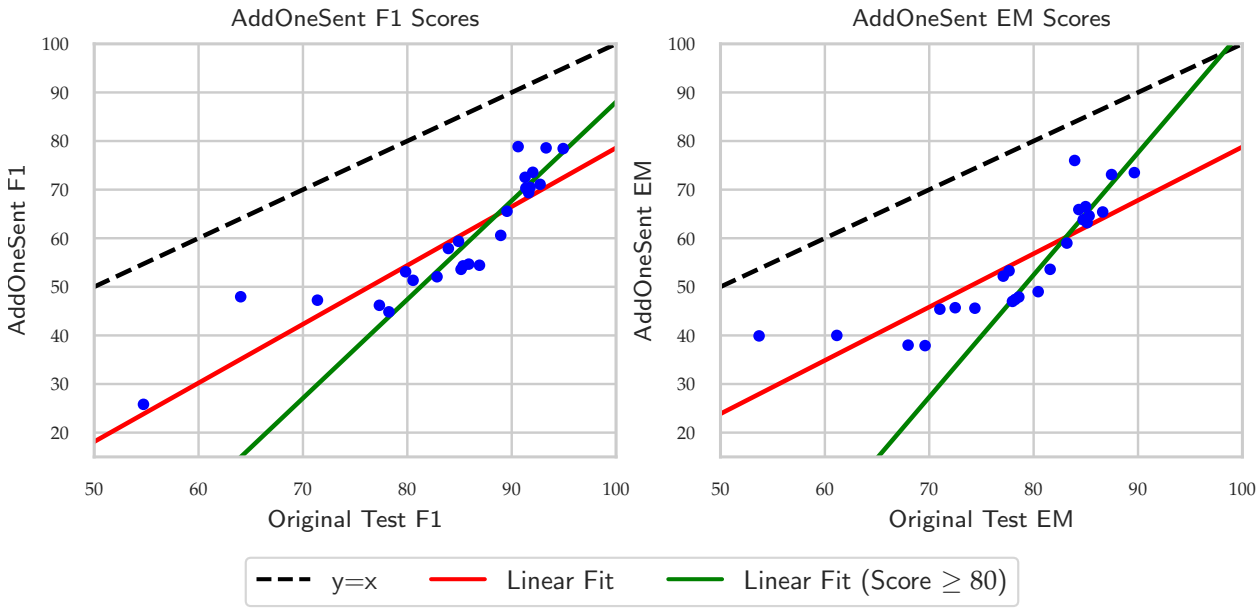


Figure 2.9: Comparison of F1 and EM scores on the original SQuAD test set versus the *adversarial* `AddOneSent` attack from Jia and Liang [113]. We observe similar phenomenon as with `AddSent`. In contrast to our *natural* distribution shifts, the linear model is no longer a good description of the data. For F1 scores, the slope of the linear fit is 1.21, and, for models with F1 scores greater than 80, the slope of the linear fit is 2.03. For EM, the slopes are 1.10 and 2.51.

2.10 Appendix: Detailed results

In this appendix, we present complete detailed results for our main distribution shift experiments.

Exact Match Scatterplots

Similar to Figure 2.1 in Section 2.1, we compare the EM scores of all models in our testbed on the SQuAD v1.1 test set versus the EM scores of all models on each of the new test sets. The results are shown in Figure 2.10. In each case, we observe a more pronounced drop than the F1 scores with average drops of 4.7, 4.7, 18.3, and 20.7 for each of the new Wikipedia,

New York Times, Reddit, and Amazon datasets, respectively. However, the primary trends are the same. In particular, we observe little evidence of overfitting on Wikipedia (the linear model nicely describes the data and the slope is approximately one), and we observe a similar ranking of magnitudes of the drop on each of the other three datasets, New York Times exhibits a small drop, followed by larger drops on Reddit and Amazon.

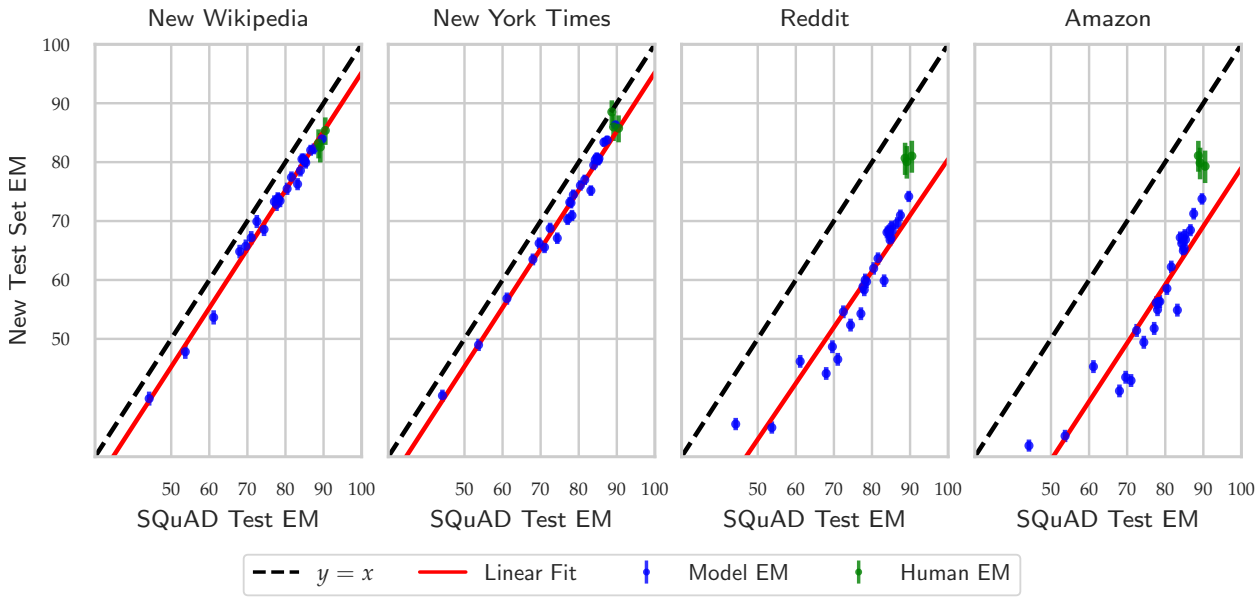


Figure 2.10: Model and human EM scores on the original SQuAD test set compared to our new test sets (shown with 95% Clopper-Pearson confidence intervals). The slopes of the linear fits are 1.00, 1.00, 0.95, and 0.99, respectively.

Probit Fits

In many cases, a linear model of F1 or EM scores is not a good fit when the scores span a wide range. In these cases, we find that a probit model describes the data better. Figure 2.11 shows the F1 scores for the Amazon dataset on both the linear scale used throughout the data and a probit scale obtained by transforming all of the F1 scores with the inverse Gaussian CDF. We observe a better linear fit for our data. Figures 2.12 and Figures 2.13 show similar probit models for each of our new datasets.

Labeler Agreement

Answer Category Shifts

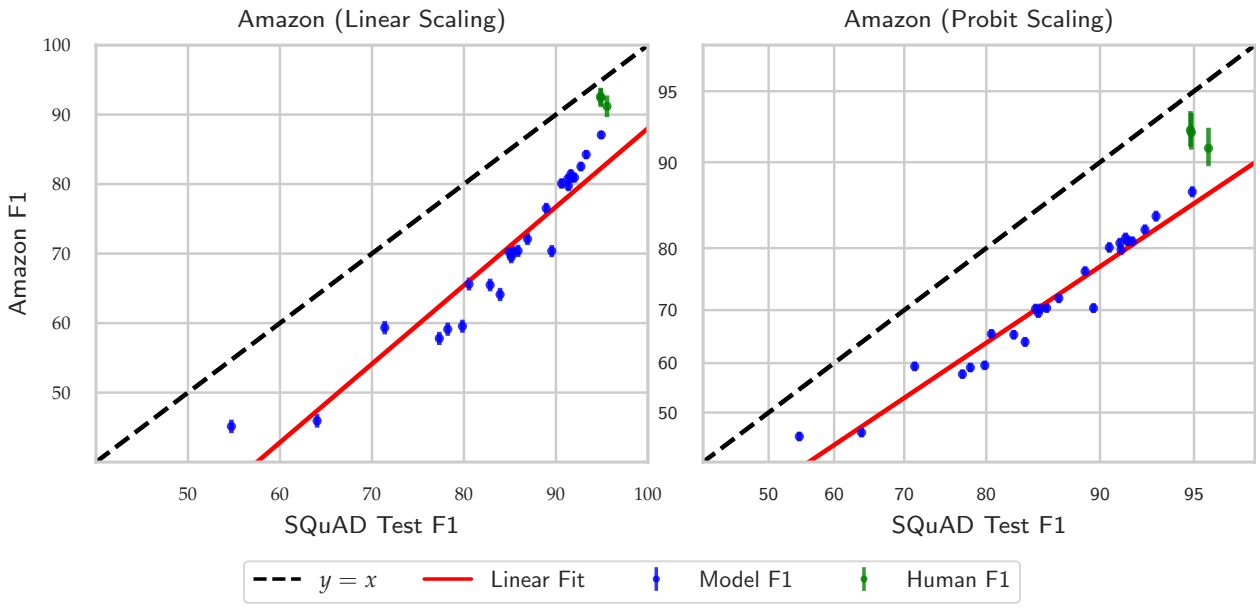


Figure 2.11: Comparison of model and human F1 scores on the original SQuAD v1.1 test set and our new Amazon test set. Each datapoint corresponds to one model in the testbed and is shown with 95% Student's-T confidence intervals. The left plot shows the model F1 scores under a linear scaling, whereas the right plot uses an *probit scale*. In other words, model F1 score x appears at $\Phi^{-1}(x)$, where Φ^{-1} is the inverse Gaussain CDF. Visual inspection shows the linear fit is much better in the probit domain.

Figure 2.12: Probit scaling of model and human F1 scores on the original SQuAD test set compared to F1 scores on our new test sets. The slopes of the linear fit are 0.93, 0.94, 0.82, and 0.89, respectively.

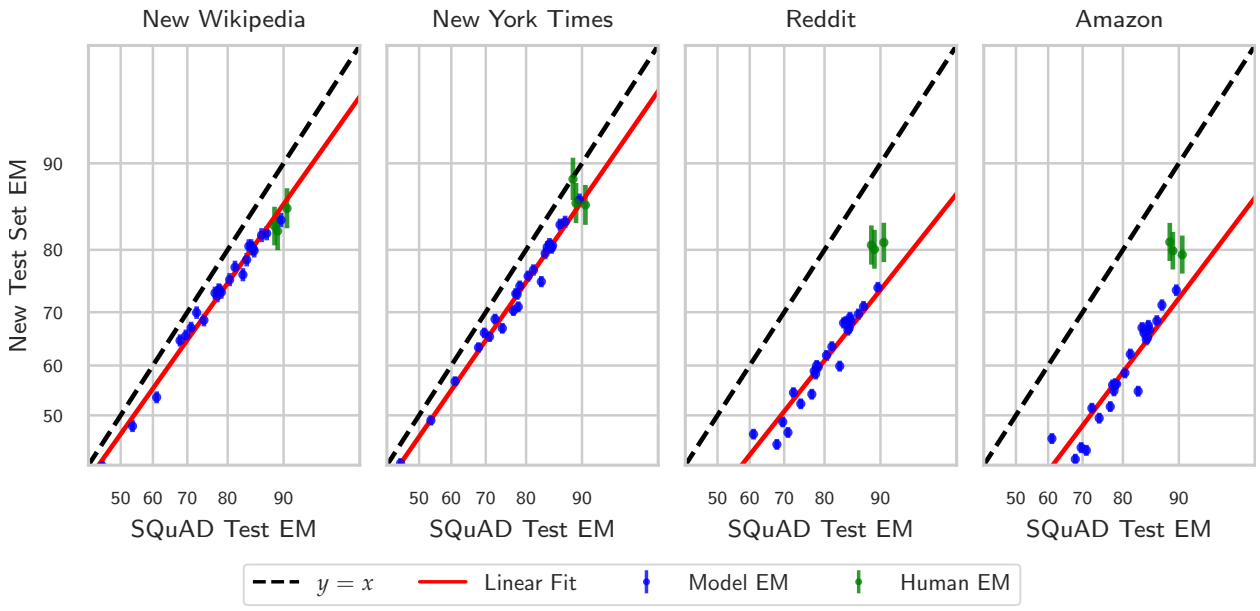


Figure 2.13: Probit scaling of model and human EM scores on the original SQuAD test set compared to EM scores on our new test sets. The slopes of the linear fit are 0.91, 0.93, 0.82, and 0.86, respectively.

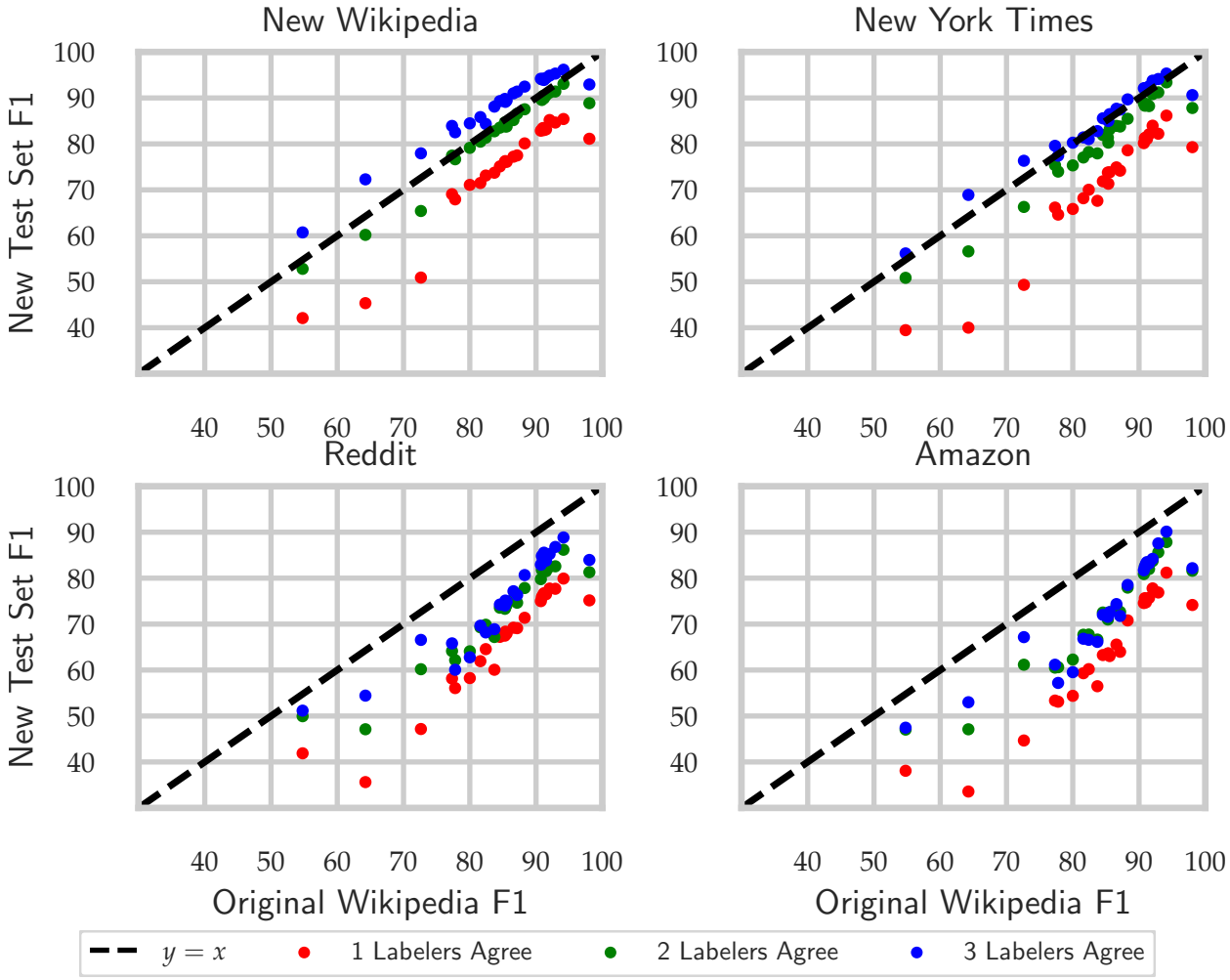


Figure 2.14: Model and human F1 scores on the original SQuAD v1.1 test set compared to our new test sets, stratified by the agreement between the answers given by the labelers, e.g. if three labelers agree, then three labelers provided identical (up to text normalization) answers to the question. Each point corresponds to a model evaluation. Label agreement roughly corresponds to question difficulty (and ambiguity). For clear and simple questions, all of the labelers typically agree. For more subtle or potentially ambiguous questions, the labeler's answers are more varied and tend to disagree more often. Across each dataset, when the questions are easier or less ambiguous (as measured by higher labeler agreement), the models experience proportionally smaller drops on the new dataset.

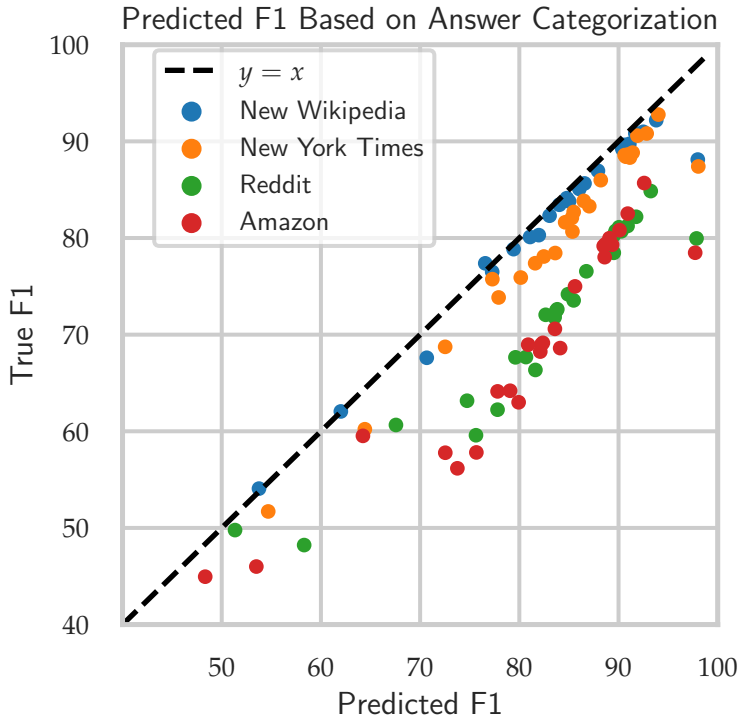


Figure 2.15: Change in answer type distributions introduced in Section 2.6 explain little of the observed performance differences across our new datasets. For each model, we compute the F1 score on each of the answer types on the SQuAD v1.1 dev set, and then we predict the F1 score on the new test set by reweighing these F1 scores based on the frequency of answer types in the new test set. Concretely, if SQuAD v1.1 was 50% `NP` answers and 50% `Places` answers, and a model has average F1 scores of 100 for `NP` and 75 for `Places`, then if a new dataset had 30% `NP` answers and 70% `Places` answers, the predicted F1 score would be 82.5 (versus 87.5 for the original). Then $y = x$ line represents a perfect model. For each of the distribution shift datasets, predictions based on answer category shifts are exceedingly optimistic and explain little of the observed drops. For instance, on the Reddit dataset, answer category shifts suggest models would lose, on average, 2-3 F1 points. However, the observed shift is 12.6 F1 points.

Models Evaluated

We evaluated a representative subset of 25 models submitted to the SQuAD leaderboard since 2016. All of the models were submitted to the CodaLab platform, and thus we evaluate every model in the exact same configuration (weights, hyperparameters, command-line arguments, execution environment, etc.) as the original submission. Below, we list all of the models we evaluated with references, where available, and links to the Codalab submission bundle.

1. XLNET-123 (single model) [252]
   `worksheets.codalab.org/bundles/0x8d330aabc5394f239749ca75a97e9087`
2. Tuned BERT-1seq Large Cased (single model)
   `worksheets.codalab.org/bundles/0xf776d7935d8e4989880ea77cb821291b`
3. MMIPN(Single)
   `worksheets.codalab.org/bundles/0x02ef30c852ac422cbe77590e8533f94c`
4. EAZI (ensemble)
   `worksheets.codalab.org/bundles/0xfaf5f109a572470ba188877fab8fcdc5`
5. BiDAF + Self Attention + ELMo (single model) [175]
   `worksheets.codalab.org/bundles/0x11f631b3e7cb4a0f8acbd60491f729b6`
6. AVIQA (single model)
   `worksheets.codalab.org/bundles/0xe821c3b185d44105b6542a07e0f76dab`
7. DPN (single model)
   `worksheets.codalab.org/bundles/0x37efc3e1594f4262b0384decbd2f964f`
8. UnsupervisedQA V1
   `worksheets.codalab.org/bundles/0xe1c53a62c8644e9b9d9fdfd18feb6a85`
9. M-NET (single)
   `worksheets.codalab.org/bundles/0x08c0461cde4c448e9a2e91a1b734c7aa`
10. SimpleBaseline (single model)
    `worksheets.codalab.org/bundles/0x8d070eba996a4e32b4840c0578544966`
11. Original BERT Large Cased
    `worksheets.codalab.org/bundles/0x91bd55572ea6456a9f6f775393c9e96c`
12. Common-sense Governed BERT-123 (single model)
    `worksheets.codalab.org/bundles/0x69acafc6ee734cdc969607da5059ba37`
13. RQA+IDR (single model)
    `worksheets.codalab.org/bundles/0x54e292cee87d4b1488b9cf0df15aeeec`

14. BERT-Large Baseline (single model) [65]
    worksheets.codalab.org/bundles/0xcd68d4f224b0425ab2b8b34ffb140a75

15. InfoWord-Base (single model)
    worksheets.codalab.org/bundles/0xa41e1de495f84786a1c84d6f6036af0d

16. MARS (single model)
    worksheets.codalab.org/bundles/0x86b64ac049d74c7e9b8af73dfc1ca207

17. BISAN (single model)
    worksheets.codalab.org/bundles/0xfd43e046161f4ba89716d5d48b25ca2f

18. UQA (single model)
    worksheets.codalab.org/bundles/0x64206b3164ea47e7a3d8a2df833c8f9b

19. EAZI (single model)
    worksheets.codalab.org/bundles/0xad2056e99a0a484f8b8e4bcc2b1b0c14

20. MEMEN (single model) [164]
    worksheets.codalab.org/bundles/0xe97e265eff8b437b8ca2b24e460cc066

21. gqa (single model)
    worksheets.codalab.org/bundles/0xc8548cd7df0547dd9003a02e5505dd77

22. BERT+Sparse-Transformer(single model)
    worksheets.codalab.org/bundles/0xd4b07885a4c244dcaf22dac21d283813

23. Jenga (single model)
    worksheets.codalab.org/bundles/0x38bce62d659e43d19f56fc2ba34c3c4d

24. AllenNLP BiDAF (single model) [208]
    worksheets.codalab.org/bundles/0x034e030aa6204c09a91b300508f8d18b

25. DNET (single model)
    worksheets.codalab.org/bundles/0xe50b1f4f7d674f138b6a0a384f33b356

We also evaluated a subset of five models from the Machine Reading for Question Answering (MRQA) Shared Task [78] on our new test sets. As in our primary experiments, all of the models were submitted to the CodaLab platform, and we evaluated every model in the exact same configuration as the original submission. Below, we list all of the models we evaluated with references and links to the submission bundle.

1. `Delphi` [146]
   worksheets.codalab.org/bundles/0x9a53e9c50f1244699c4a24aee483bd4c

2. `HierAtt` [163]
   worksheets.codalab.org/bundles/0x8d851db3255b485c97646c5c0ba812a2

3. `Bert-Large+Adv Train` [132]
   `worksheets.codalab.org/bundles/0xa113983bc3fc42ff89bf3838a6177a0c`

4. `BERT-cased-whole-word`
   `worksheets.codalab.org/bundles/0x456676760aae452cb44ade00bb515b64`

5. `BERT-Multi-Finetune`
   `worksheets.codalab.org/bundles/0x5716df3b477a452a997bcebb9e179c89`

The remaining models submitted to the competition were either not publically accessible or otherwise unable to run on Codalab.

2.11 Appendix: Dataset collection details

In this section, we provide further details regarding our data collection pipeline.

Passage Length Statistics

We report statistics on various text length statistics. We split each paragraph into individual sentences, words, and characters using Spacy [106] and used those components to compute the statistics.

Figures 2.16, 2.17, and 2.18 show the paragraph lengths in sentences, words, and characters across each dataset respectively. Figures 2.19 and 2.20 show the sentence lengths in words and characters across each dataset respectively. Figure 2.21 shows the word length in characters across each dataset.

MTurk Experiment

Worker Details. Crowdworkers were required to have a 97% HIT acceptance rate, a minimum of 1000 HITs, and be located in the United States or Canada. Workers were asked to spend four minutes per paragraph when asking questions and one minute per question when answering questions. We paid workers $9.60 per hour for the amount of time required to complete each task, using an inflation rate of 6.52% between 2016 and 2019.

UI Examples. The task directions and website UI are identical to the original SQuAD data collection setup with the sole exception that the original tasks had workers ask and answer questions for all of the paragraphs for each article, whereas our tasks limit each worker to at most 5 paragraphs. Figures 2.22 and 2.23 show the directions and an example HIT for the Ask task, whereby workers pose questions for the article. Figures 2.24 and 2.25 show the directions and an example HIT for the Answer task, whereby workers answer questions posed during the Ask task.

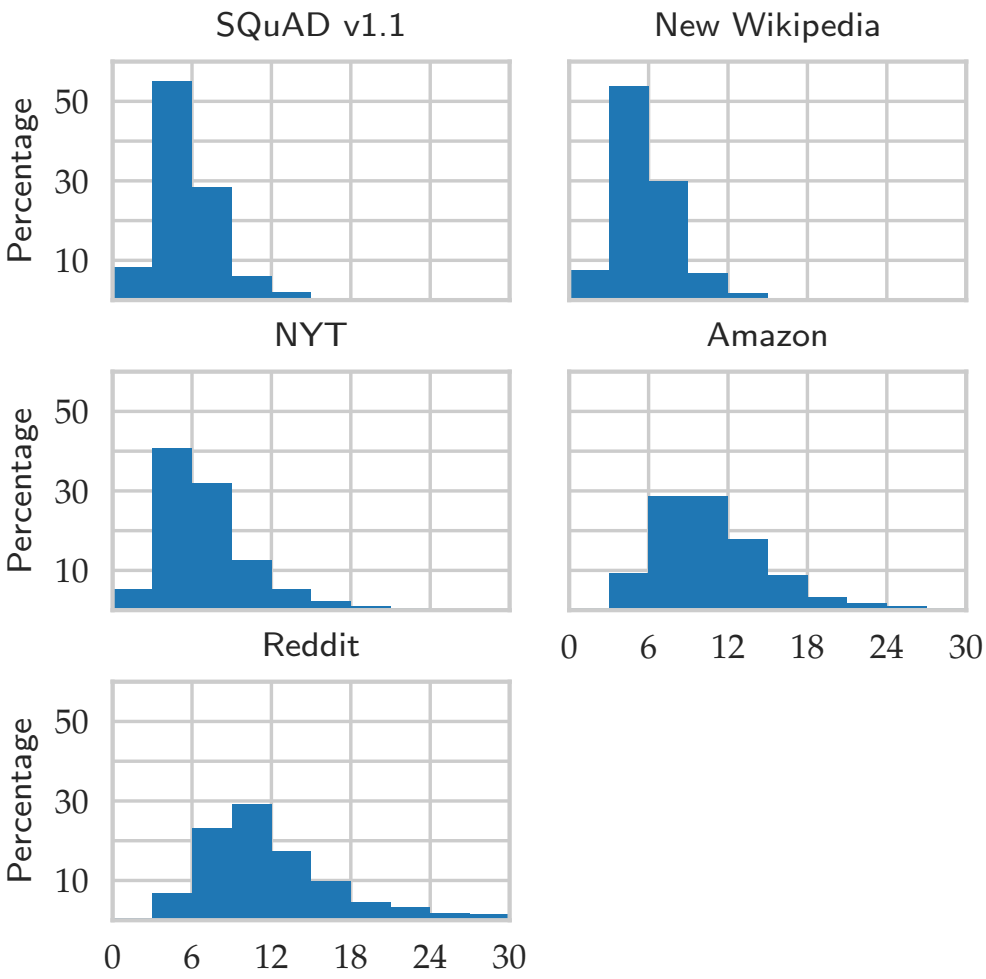


Figure 2.16: Histograms of the number of sentences in each paragraph for both the original and new datasets. The new Wikipedia dataset matches the original dataset, while all other new datasets have different histograms.

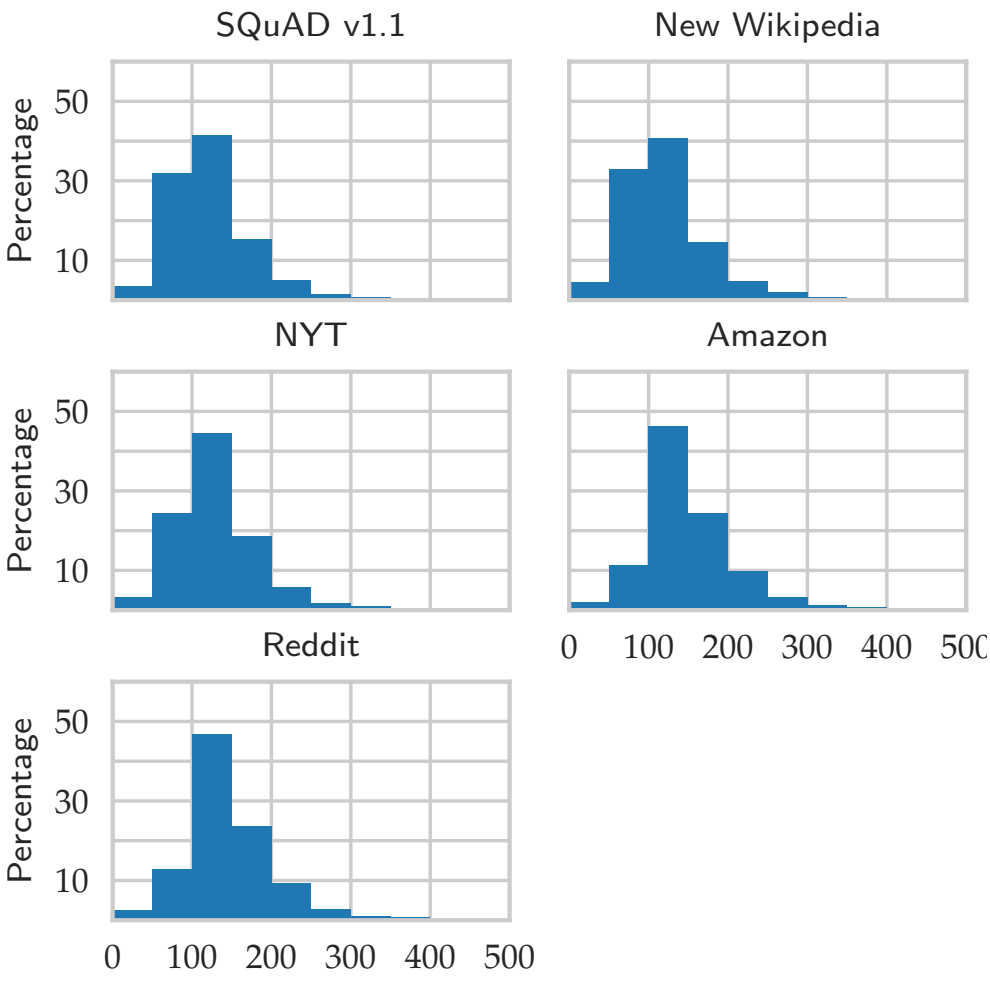


Figure 2.17: Histograms of the number of words in each paragraph for both the original and new datasets. The histograms all match closely, although the Amazon and Reddit datasets' paragraphs have slightly more words.

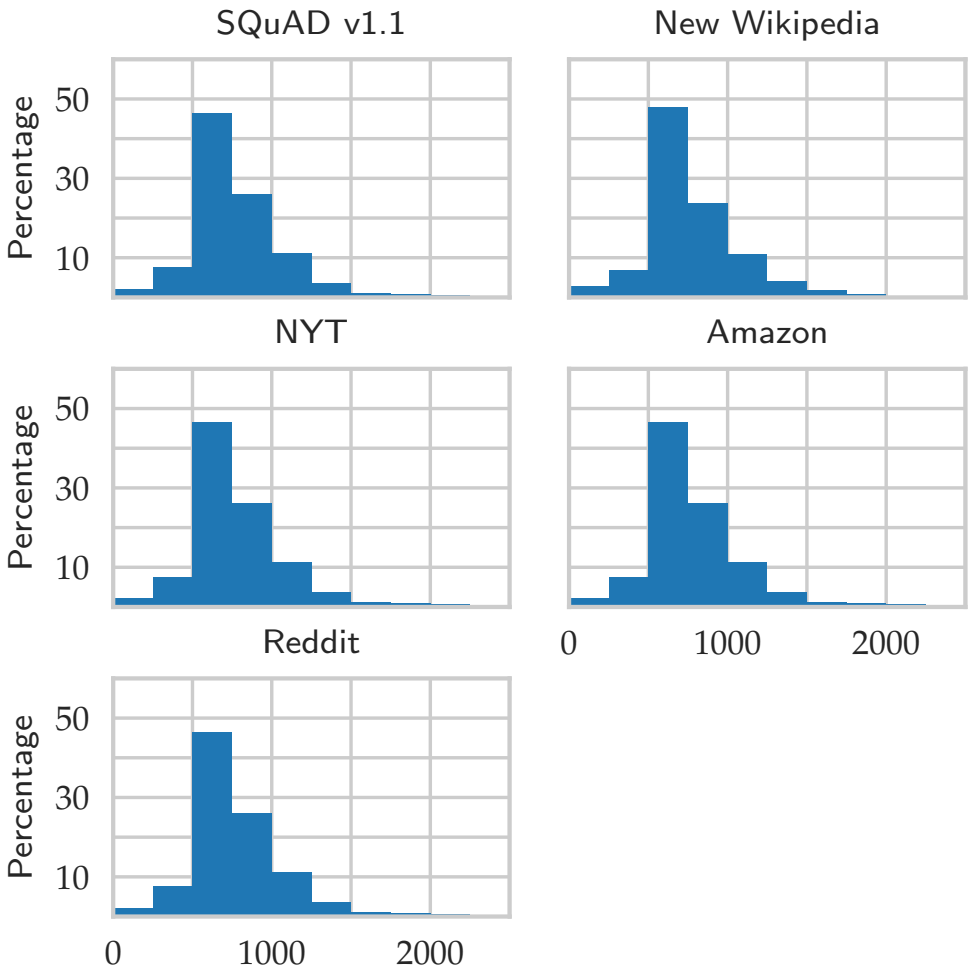


Figure 2.18: Histograms of the number of characters in each paragraph for both the original and new datasets. The histograms lengths match exactly since we sample in a way that ensures the character length will match for each new dataset.

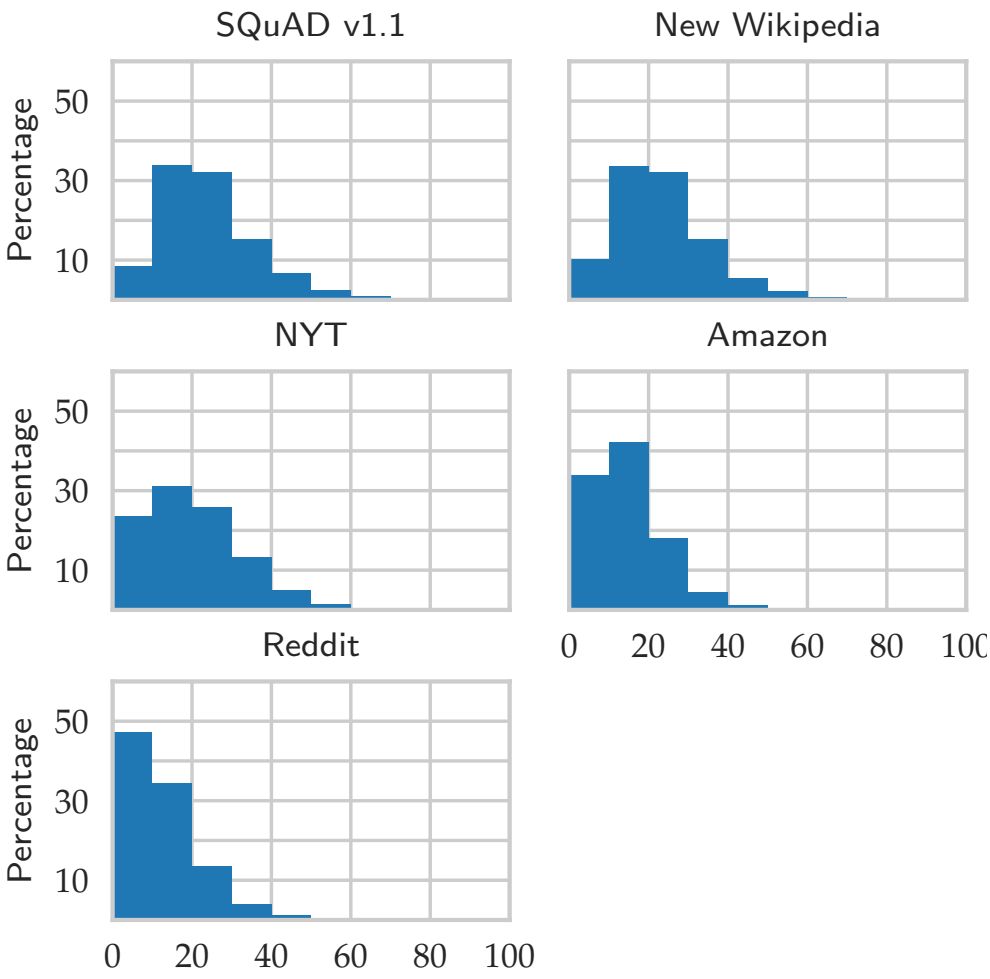


Figure 2.19: Histograms of the number of words in each sentence for both the original and new datasets. The Amazon and Reddit datasets both have significantly shorter sentences than the original dataset, while the NYT dataset has slightly shorter sentences. The new Wikipedia dataset matches the sentence length of the original dataset.

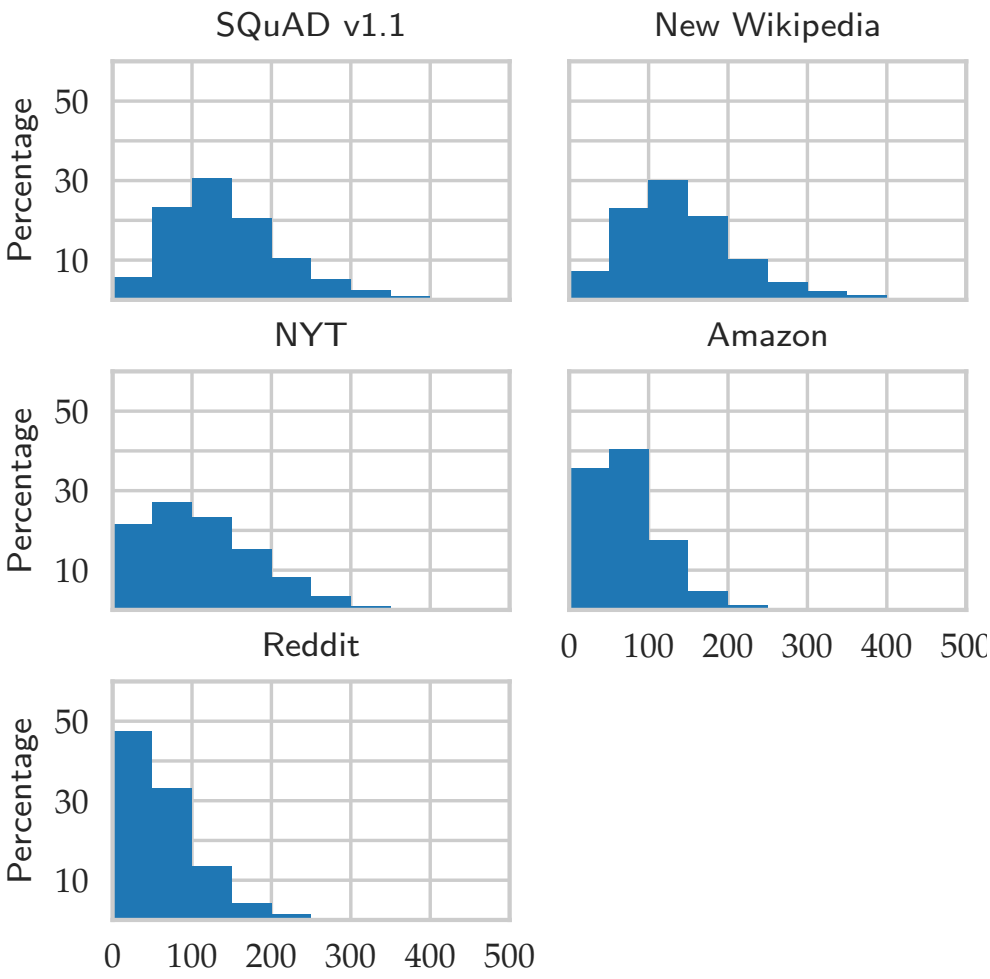


Figure 2.20: Histograms of the number of characters in each sentence for both the original and new datasets. The Amazon and Reddit datasets both have significantly shorter sentences than the original dataset, while the NYT dataset has slightly shorter sentences. The new Wikipedia dataset matches the sentence length of the original dataset.

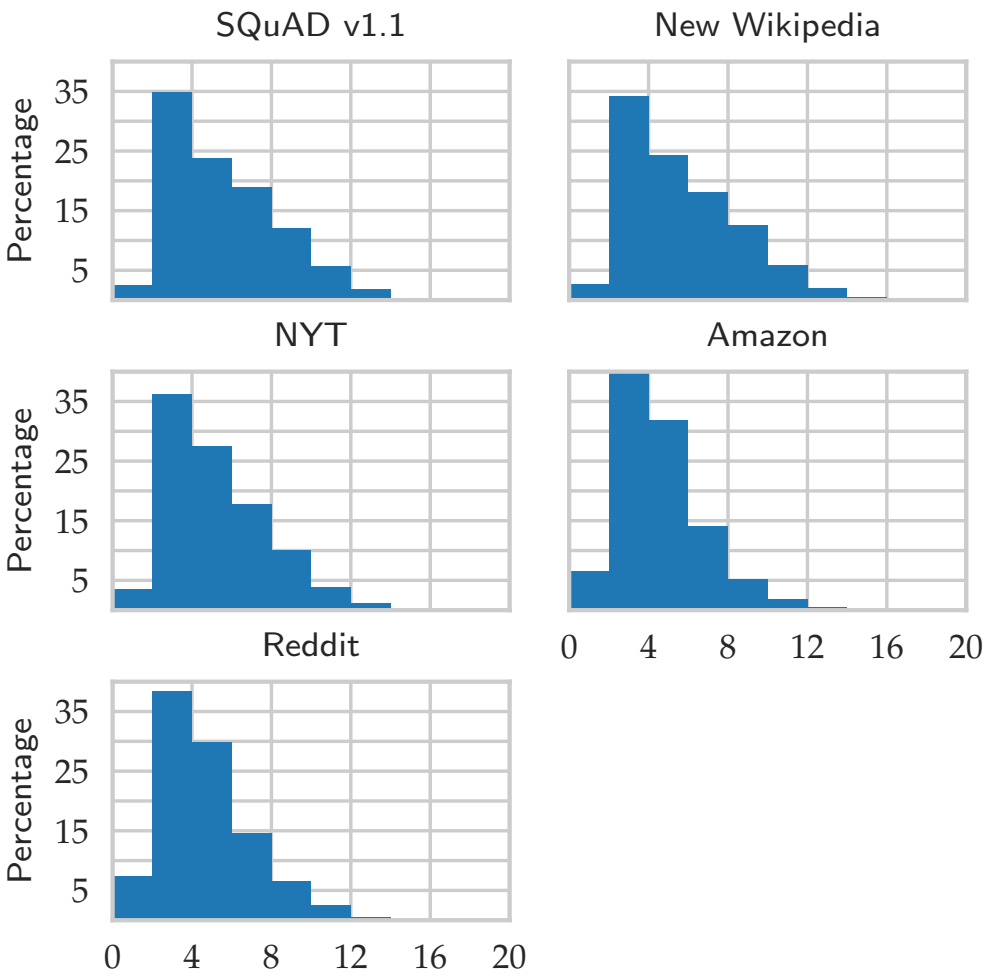


Figure 2.21: Histograms of the number of characters in each word for both the original and new datasets. All datasets have similar histograms when it comes to worth length, although the Amazon and Reddit word lengths decay slightly faster.

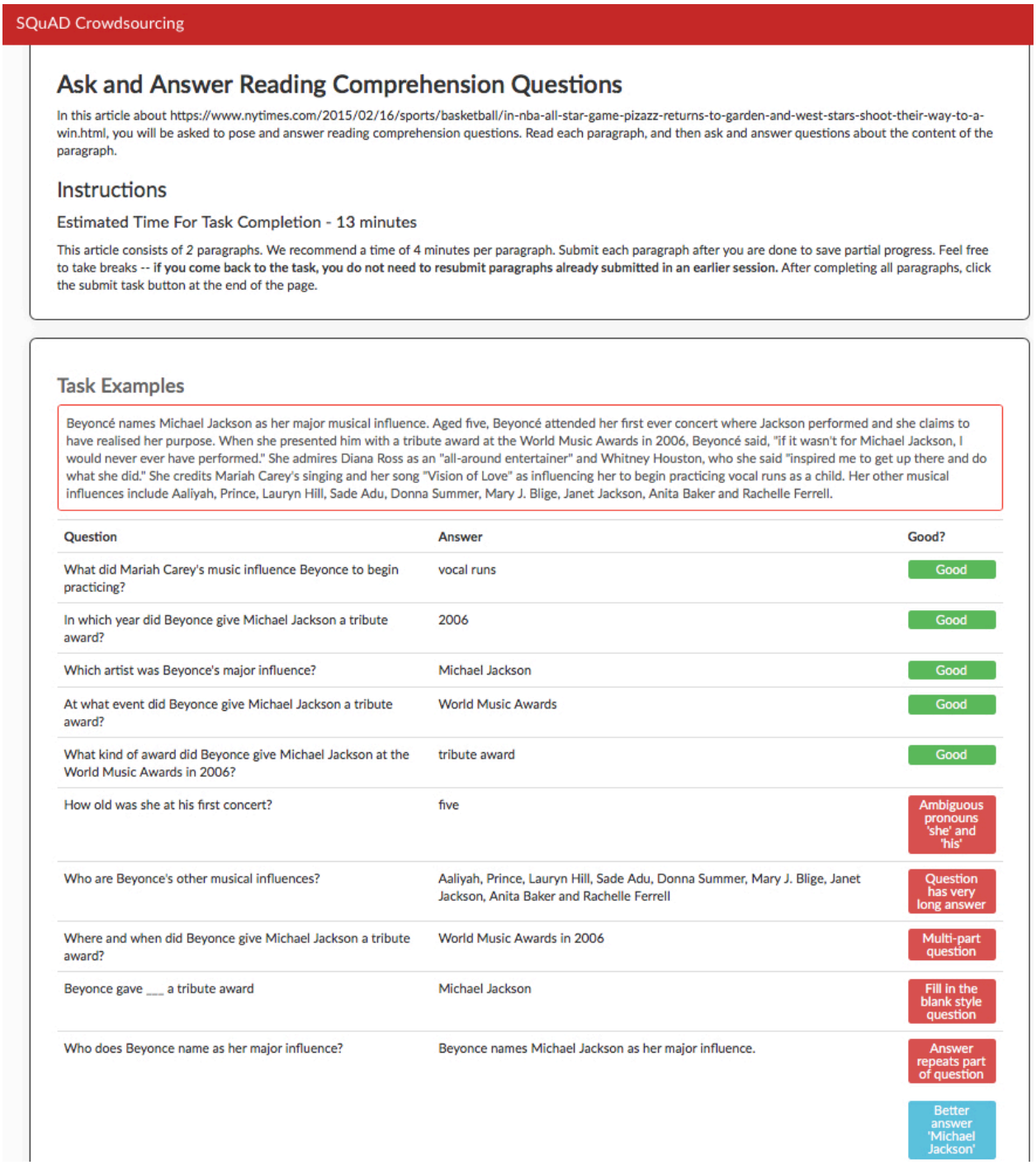


Figure 2.22: Ask task directions.

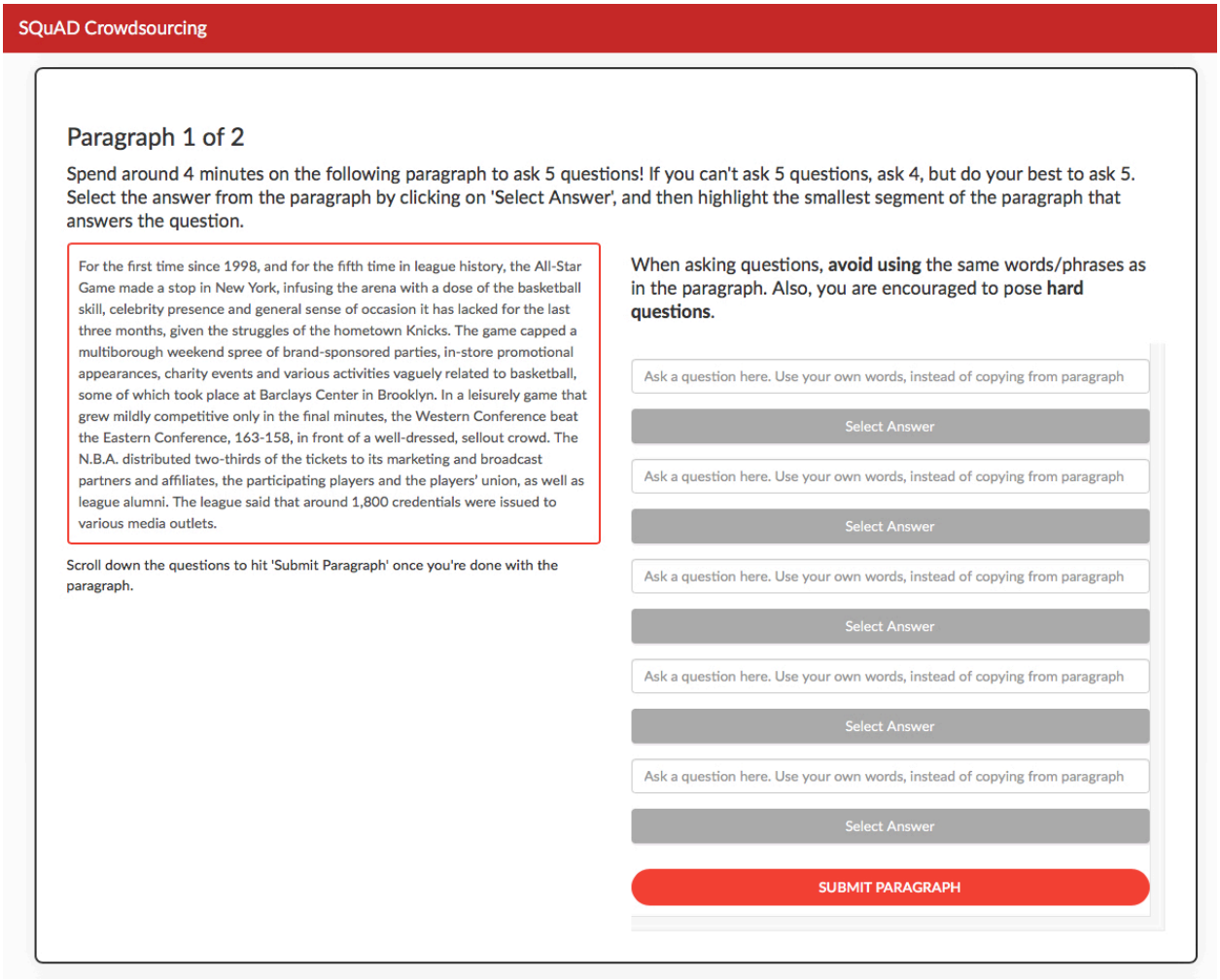


Figure 2.23: Ask task example.

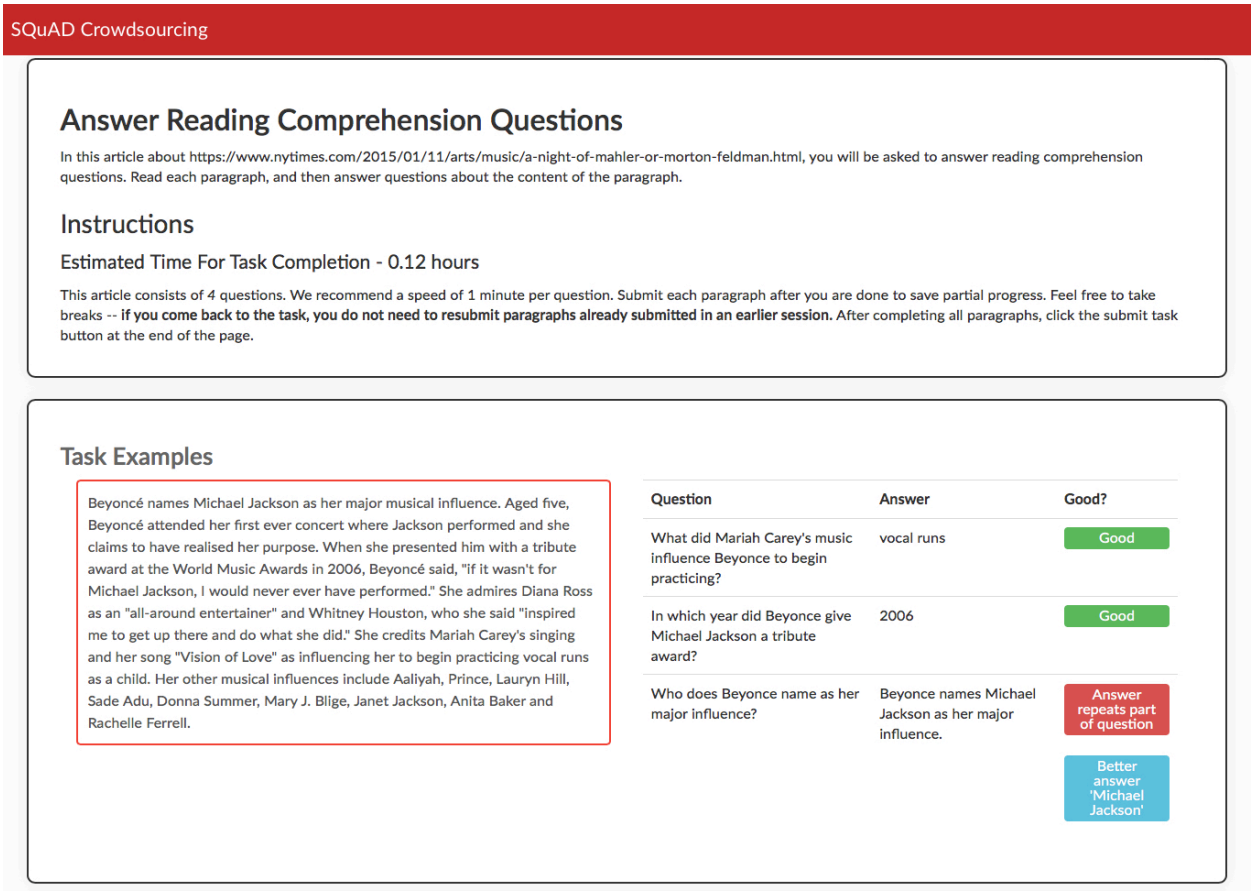


Figure 2.24: Answer task directions.

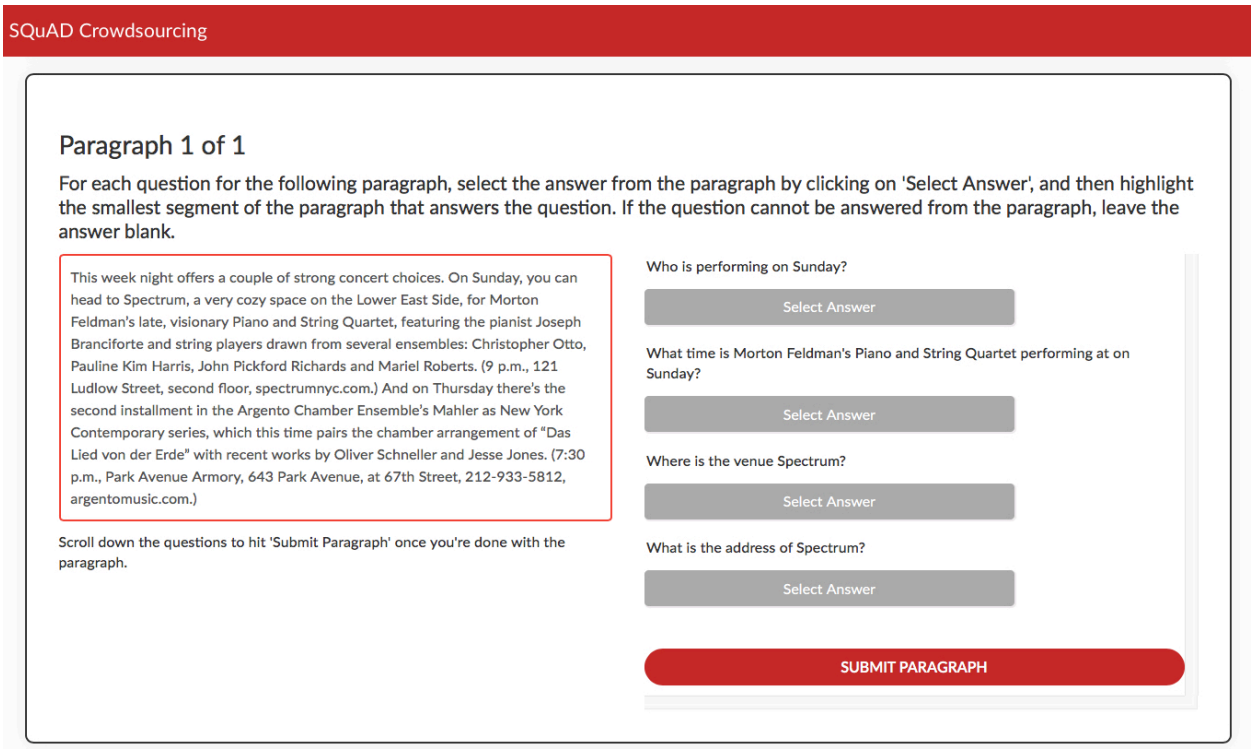


Figure 2.25: Answer task example.

Chapter 3

Model Similarity Mitigates Test-set Reuse

3.1 Introduction

Be it validation sets for model tuning, popular benchmark data, or machine learning competitions, the holdout method is central to the scientific and industrial activities of the machine learning community. As compute resources scale, a growing number of practitioners evaluate an unprecedented number of models against various holdout sets. These practices, collectively, put significant pressure on the statistical guarantees of the holdout method. Theory suggests that for k models chosen independently of n test data points, the holdout method provides valid risk estimates for each of these models up to a deviation on the order of $\sqrt{\log(k)/n}$ [72]. But this bound is the consequence of an unrealistic assumption. In practice, models incorporate prior information about the available test data since human analysts choose models in a manner guided by previous results.

Adaptivity significantly complicates the theoretical guarantees of the holdout method. A simple adaptive strategy, resembling the practice of selectively ensembling k models, can bias the holdout method by as much as $\sqrt{k/n}$ [72]. If this bound were attained in practice, holdout data across the board would rapidly lose its value over time. Nonetheless, in the previous chapter, we observed that despite years of heavy test set reuse, the SQuAD benchmark showed no signs of adaptive overfitting. Recent replication studies give similar evidence that popular machine learning benchmarks continue to support progress despite years of extensive reuse [190, 247].

In this chapter, we contribute a new explanation for why the adaptive bound is not attained in practice and why even the standard non-adaptive bound is more pessimistic than it needs to be. Our explanation centers around the phenomenon of *model similarity*. Practitioners evaluate models that incorporate common priors, past experiences, and standard practices. As we show empirically, this results in models that exhibit significant agreement in their predictions, well beyond what would follow from their accuracy values alone. Comple-

menting our empirical investigation of model similarity, we provide a new theoretical analysis of the holdout method that takes model similarity into account, vastly improving over known bounds in the adaptive and non-adaptive cases when model similarity is high.

Our contributions

Our contributions are two-fold. On the empirical side, we demonstrate that a large number of proposed ImageNet [63, 201] and CIFAR-10 [123] models exhibit a high degree of similarity: Their predictions agree far more than we would be able to deduce from their accuracy levels alone. Complementing our empirical findings, we give new generalization bounds that incorporate a measure of similarity. Our generalization bounds help to explain why holdout data has much greater longevity than prior bounds suggest when models are highly similar, as is the case in practice. Figure 3.1 summarizes these two complementary developments.

Underlying Figure 3.1a is a family of representative ImageNet models whose pairwise similarity we evaluate. The mean level of similarity of these models, together with a refined union bound, offers a $4\times$ improvement over a carefully optimized baseline bound that does not take model similarity into account. In Figure 3.1b we compare our guarantee on the number of holdout reuses with the baseline bound. This illustrates that our bound is not just asymptotic, but concrete—it gives meaningful values in the practical regime. Moreover, in Section 3.5 we discuss how an additional assumption on model predictions can boost the similarity based guarantee by multiple orders of magnitude.

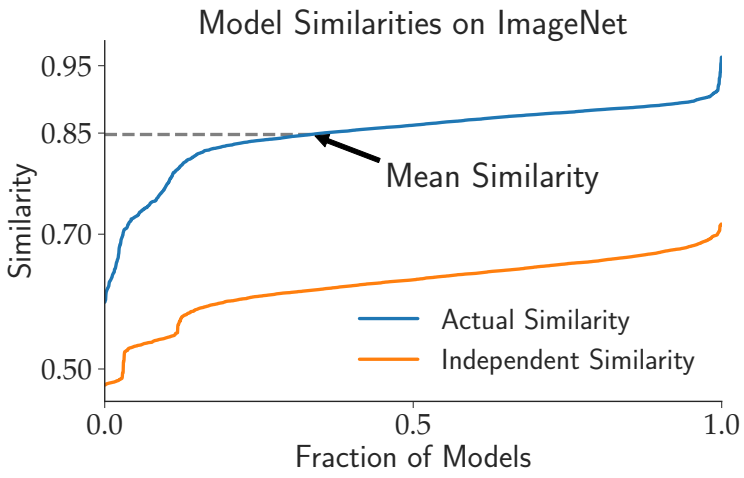
Investigating model similarity in practice further, we evaluate similarity of models encountered during the course of a large random hyperparamter search and a large neural architecture search for the CIFAR-10 dataset. We find that the pairwise model similarities throughout both procedures remain high. The similarity provides a counterweight to the massive number of model evaluations, limiting the amount of overfitting we observe.

Related work

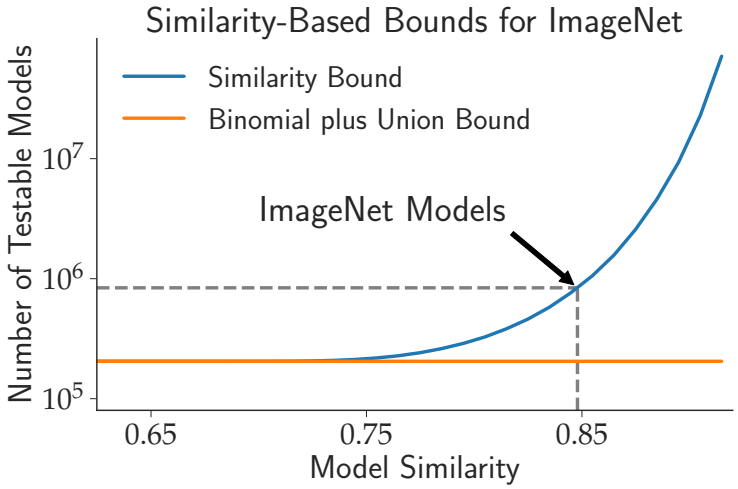
Recht et al. [190] recently created new test sets for ImageNet and CIFAR10, carefully following the original test set creation processes. Reevaluating all proposed models on the new test sets showed that while there was generally an absolute performance drop, the effect of overfitting due to adaptive behavior was limited to non-existent. Indeed, newer and better models on the old test set also performed better on the new test set, even though they had in principle more time to adapt to the test set. Also, Yadav and Bottou [247] recently released a new test set for the seminal MNIST task, on which they observed no overfitting.

Dwork et al. [72] recognized the issue of adaptivity in holdout reuse and provided new holdout mechanisms based on noise addition that support quadratically more queries than the standard method in the worse case. There is a rich line of work on adaptive data analysis; Smith [215] offers a comprehensive survey of the field.

We are not the first to proffer an explanation for the apparent lack of overfitting in machine learning benchmarks. Blum and Hardt [26] argued that if analysts only check if



(a) Pairwise model similarities on ImageNet



(b) Number of models to be tested

Figure 3.1: **(a)** shows the empirical pairwise similarity between Imagenet models and the hypothetical similarity between models if they were making mistakes independently. **(b)** plots the number of testable models on Imagenet such that the population error rates for all models are estimated up to $\pm 1\%$ error with probability 0.95. We compare the guarantee of the standard union bound with that of a union bound which considers model similarities.

they improved on the previous best model, while ignoring models that did not improve, better adaptive generalization bounds are possible. Zrnic and Hardt [264] offered improved guarantees for adaptive analysts that satisfy natural assumptions, e.g. the analyst is unable to arbitrarily use information from queries asked far in the past. More recently, Feldman, Frostig, and Hardt [77] gave evidence that the number of classes in a classification problem helps mitigate overfitting in benchmarks. We see these different explanations as playing

together in what is likely the full explanation of the available empirical evidence. In parallel to our work, Yadav and Bottou [247] discussed the advantages of comparing models on the same test set; pairing tests can provide tighter confidence bounds for model comparisons in this setting than individual confidence intervals for each model.

3.2 Problem setup

Let $f : \mathcal{X} \to \mathcal{Y}$ be a classifier mapping examples from domain $\mathcal{X}$ to a label from the set $\mathcal{Y}$. Moreover, we consider a test set $S = \{(x_1, y_1), \ldots\}$ of n examples sampled i.i.d. from a data distribution $\mathcal{D}$. The main quantity we aim to analyze is the gap between the accuracy of the classifier f on the test set S and the population accuracy of the same classifier under the distribution $\mathcal{D}$. If the gap between the two accuracies is large, we say f overfit to the test set.

As is commonly done in the adaptive data analysis literature [12], we formalize interactions with the test set via *statistical queries* $q : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. In our case, the queries are $\{0, 1\}$-valued; given a classifier f we consider the query q_f defined by $q_f(z) = \mathbb{1}\{f(x) \neq y\}$, where $z = (x, y)$. Then, we denote the empirical mean of query q_f on the test set S (i.e., f's test error) by $\mathbb{E}_S[q_f] = \frac{1}{n}\sum_{i=1}^{n} q_f(z_i)$. The population mean (population error) is accordingly defined as $\mathbb{E}_{\mathcal{D}}[q] = \mathbb{E}_{z\sim\mathcal{D}} q(z)$.

When discussing overfitting, we are usually interested in a set of classifiers, e.g., obtained via a hyperparameter search. Let $f_1, \ldots, f_k$ be such a set of classifiers and $q_1, \ldots, q_k$ be the set of corresponding queries. To quantify the probability that overfitting occurs (i.e., one of the f_i has a large deviation between test and population accuracy), we would like to upper bound the probability

$$\mathbb{P}\left(\max_{1\leq i\leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \geq \varepsilon\right). \tag{3.1}$$

A standard way to bound (3.1) is to invoke the union bound and treat each query separately:

$$\mathbb{P}\left(\max_{1\leq i\leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \geq \varepsilon\right) \leq \sum_{i=1}^{k} \mathbb{P}\left(|\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \geq \varepsilon\right) \tag{3.2}$$

We can then utilize standard concentration results to bound the right hand side. However, such an approach inherently cannot capture dependencies between the queries q_i (or classifiers f_i). In particular, we are interested in the similarity between two queries q and q' measured by $\mathbb{P}\left(q(z) = q'(z)\right)$ (the probability of agreement between the 0-1 losses of the corresponding two classifiers). The main goal of this chapter is to understand how high similarity can lead to better bounds on (3.1), both in theory and in numerical experiments with real data from ImageNet and CIFAR-10.

3.3 Non-adaptive classification

We begin by analyzing the effect of the classifier similarity when the classifiers to be evaluated are chosen *non-adaptively*. For instance, this is the case when the algorithm designer fixes a grid of hyperparameters to be explored before evaluating any of the classifiers on the test set. To draw valid gains from the hyperparameter search, it is important that the resulting test accuracies reflect the true population accuracies, i.e., probability (3.1) is small.

Bound (3.2) is sharp when the events $\{|\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \geq \varepsilon\}$ are almost disjoint, which is not true when the queries are similar to each other. To address this issue, we modify our use of the union bound. We consider the left tails $\mathcal{E}_i = \{\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i] \geq \varepsilon\}$. For any $t \geq 0$, we obtain

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{i=1}^{k} \mathcal{E}_i\right) &\leq \mathbb{P}\left(\{\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] \geq \varepsilon - t\} \bigcup_{i=2}^{k} \mathcal{E}_i\right) \qquad (3.3)\\
&= \mathbb{P}\left(\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] \geq \varepsilon - t\right) + \mathbb{P}\left(\bigcup_{i=2}^{k} \mathcal{E}_i \cap \{\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] < \varepsilon - t\}\right)\\
&\leq \mathbb{P}\left(\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] \geq \varepsilon - t\right) + \sum_{i=2}^{k} \mathbb{P}\left(\mathcal{E}_i \cap \{\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] < \varepsilon - t\}\right).
\end{aligned}$$

Intuitively, the terms $\mathbb{P}\left(\mathcal{E}_i \cap \{\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] < \varepsilon - t\}\right)$ are small when the queries q_1 and q_i are similar: if $\mathbb{P}(q_1(z) = q_i(z))$ is large, we cannot simultaneously have $\mathbb{E}_S[q_1] < \mathbb{E}_{\mathcal{D}}[q_1] + \varepsilon - t$ and $\mathbb{E}_S[q_i] \geq \mathbb{E}_{\mathcal{D}}[q_i] + \varepsilon$ since the deviations go into opposite directions. In the rest of this section, we make this intuition precise in and derive an upper bound on (3.1) in terms of the query similarities. Before we state our main result, we introduce the following notion of a similarity covering.

Definition 1. *Let $\mathcal{F}$ be a set of queries. We say a query set M is a η similarity cover of $\mathcal{F}$ if for any query $q \in \mathcal{F}$ there exist $q', q'' \in M$ such that $\mathbb{E}_{\mathcal{D}}[q'] \leq \mathbb{E}_{\mathcal{D}}[q]$, $\mathbb{E}_{\mathcal{D}}[q''] \geq \mathbb{E}_{\mathcal{D}}[q]$, $\mathbb{P}(q'(z) = q(z)) \geq \eta$, and $\mathbb{P}(q''(z) = q(z)) \geq \eta$ (M does not necessarily have to be a subset of $\mathcal{F}$). Let $N_\eta(\mathcal{F})$ denote the size of a minimal η similarity cover of $\mathcal{F}$ (when the query set $\mathcal{F}$ is clear from context we use the simpler notation N_η).*

Theorem 2. *Let $\mathcal{F} = \{q_1, q_2, \ldots, q_k\}$ be a collection of queries $q_i \colon \mathcal{Z} \to \{0, 1\}$ independent of the test set $\{z_1, z_2, \ldots, z_n\}$. Then, for any $\eta \in [0, 1]$ we have*

$$\mathbb{P}\left(\max_{1 \leq i \leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \geq \varepsilon\right) \;\leq\; 2N_\eta e^{-\frac{n\varepsilon^2}{2}} + 2ke^{-\frac{n\varepsilon}{4}\log\left(1 + \frac{\varepsilon}{4(1-\eta)}\right)}. \qquad (3.4)$$

Then, for all $\eta \leq 1 - \max\left\{\frac{2\log(4k/\delta)}{n}, \sqrt{\frac{\log(4N_\eta/\delta)}{2n}}\right\}$, we have with probability $1 - \delta$

$$\max_{1 \leq i \leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \leq \max\left\{\sqrt{\frac{2\log(4N_\eta/\delta)}{n}}, \sqrt{\frac{32(1-\eta)\log\left(4k/\delta\right)}{n}}\right\}. \qquad (3.5)$$

Moreover, if $\varepsilon = \sqrt{\frac{\log((2N_\eta+1)/\delta)}{n}}$ *and* $\eta \geq 1 - \frac{\varepsilon}{4\left(e^{2\varepsilon}(2k)^{\frac{4}{n\varepsilon}}-1\right)}$*, we have with probability* $1-\delta$

$$\max_{1\leq i\leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \leq \varepsilon. \tag{3.6}$$

To elucidate how model similarity η controls the number of queries k for which Theorem (2) gives a non-trivial bound, consider the case where $N_\eta = 1$, i.e. at least one model is η-similar to all of the others. As the similarity η of the model collection grows, the number of queries k grows as well, as the following simple result shows.

Corollary 3. *Let* $\mathcal{F} = \{q_1, q_2, \ldots, q_k\}$ *be a collection of* k *queries* $q_i \colon \mathcal{Z} \to \{0,1\}$ *fixed independently of the test set. Choose* $\eta_\star$ *so that* $N_{\eta_\star} = 1$*. Suppose* $n \geq c_1 \max\{\varepsilon^{-1}, \varepsilon^{-2}\}$ *and the number of queries* k *satisfies*

$$k \leq \frac{c_2 \varepsilon}{(1-\eta_\star)}$$

for positive constants c_1, c_2*. Then, with probability* $3/4$*,* $\max_{1\leq i\leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \leq \varepsilon$*.*

The proof of Theorem (2) starts with the refined union bound (3.3), or a standard triangle inequality, and then applies the Chernoff concentration bound shown in Lemma 4 for random variables which take values in $\{-1, 0, 1\}$. We defer the proof details of both the lemma and the theorem to Appendix 3.7.

Lemma 4. *Suppose* X_i *are i.i.d. discrete random variables which take values* -1*,* 0*, and* 1 *with probabilities* p_{-1}*,* p_0*, and* p_1 *respectively, and hence* $\mathbb{E}X_i = p_1 - p_{-1}$*. Then, for any* $t \geq 0$ *such that* $p_1 - p_{-1} + t/2 \geq 0$ *we have*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > p_1 - p_{-1} + t\right) \leq e^{-\frac{nt}{2}\log\left(1+\frac{t}{2p_1}\right)}.$$

Discretization arguments based on coverings are standard in statistical learning theory. Covers based on the population Hamming distance $\mathbb{P}(q'(z) \neq q(z))$ have been previously studied [66, 129] (Note that for $\{0,1\}$-valued queries the Hamming distance is equal to the L^2 and L^1 distances). An important distinction between our result and prior work is that prior work requires η to be greater than $1-\varepsilon$. Theorem 2 can offer an improvement over the standard guarantee $\sqrt{\log(k)/n}$ even when η is much smaller than $1-\varepsilon$. First of all note that (3.5) holds for η bounded away from one. Moreover, since $e^{2\varepsilon} \approx 1 + 2\epsilon$, if $(2k)^{\frac{4}{n\varepsilon}} \leq 1 + \sqrt{\varepsilon}$ (the choice of $1+\sqrt{\varepsilon}$ is somewhat arbitrary), we see the requirement on η for (3.6) is satisfied when η is on the order of $1 - \sqrt{\varepsilon}$.

3.4 Adaptive classification

In the previous section, we showed similarity can prevent overfitting when the sequence of queries is chosen *non-adaptively*, i.e. when the queries $\{q_1, q_2, \ldots, q_n\}$ are fixed independently of the test set S. In the *adaptive* setting, we assume the query q_t can be selected as a function of the previous queries $\{q_1, q_2, \ldots, q_{t-1}\}$ and estimates $\{\mathbb{E}_S[q_1], \mathbb{E}_S[q_2], \ldots, \mathbb{E}_S[q_{t-1}]\}$. Even when queries are chosen adaptively, we show leveraging similarity can provide sharper bounds on the probability of overfitting, $\mathbb{P}\left(\max_{1\leq i\leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \geq \varepsilon\right)$.

In the adaptive setting, the field of adaptive data analysis offers a rich technical repertoire to address overfitting [72, 215]. In this framework, analogous to the typical machine learning workflow, an analyst iteratively selects a classifier and then queries a mechanism to provide an estimate of test-set performance. In practice, the mechanism often used is the *Trivial Mechanism* which computes the empirical mean of the query on the test set and returns the exact value to the analyst. For simplicity, we study how similarity improves the performance of the trivial mechanism.

The empirical mean of any query can take at most $n+1$ values, and thus a deterministic analyst might ask at most $(n+1)^{k-1}$ queries in k rounds of interaction with the Trivial Mechanism. Let $\mathcal{F}$ denote the set of $(n+1)^{k-1}$ possible queries. Then, we apply Theorem 2 to $\mathcal{F}$.

Corollary 5. *Let $\mathcal{F}$ be the set of queries that a fixed analyst $\mathcal{A}$ might query the Trivial Mechanism. We assume that the Trivial Mechanism has access to a test set of size n. Let $\alpha \in [0, 1]$,*

$$\varepsilon = \sqrt{\frac{4(k^{1-\alpha}\log(n+1) + \log(2/\delta))}{n}},$$

and $\eta = 1 - \frac{\varepsilon}{4(e^{\varepsilon k^\alpha} - 1)}$. If $N_\eta(\mathcal{F}) \leq (n+1)^{k^{1-\alpha}}$, we have with probability $1-\delta$

$$\max_{1\leq i\leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \leq \varepsilon,$$

for any queries q_1, q_2, $\ldots q_k$ chosen adaptively by $\mathcal{A}$.

Proof. Note that when $\eta = 1 - \frac{\varepsilon}{4(e^{\varepsilon k^\alpha} - 1)}$ we have $\log\left(1 + \frac{\varepsilon}{4(1-\eta)}\right) \geq \varepsilon k^\alpha$. Then, the result follows from the first part of Theorem 2. □

In Corollary 5, the parameter α quantifies the strength of the similarity assumption. For $\alpha = 0$, there is no similarity requirement, and Corollary 5 always applies. In this case, the bound matches standard results for the trivial mechanism with $\varepsilon = \tilde{O}(\sqrt{k/n})$. However, as α grows, the similarity requirement becomes restrictive while the corresponding confidence interval becomes increasingly tight. In particular, for any $\alpha > 0$, if $\mathcal{F}$ permits a similarity cover $N_\eta(\mathcal{F}) \leq (n+1)^{k^{1-\alpha}}$ for $\eta = 1 - (\varepsilon/4)(e^{\varepsilon k^\alpha} - 1)^{-1}$, we obtain a super linear improvement in the dependence on k. For instance, if $\alpha = 1/2$, then $\varepsilon = \tilde{O}(\sqrt{k^{1/2}/n})$, and we obtain a

quadratic improvement in the number of queries for a fixed sample size. This improvement is similar to that achieved by the Gaussian mechanism [12, 72]. Moreover, since our technique is essentially tightening a union bound, this improvement easily extends to other mechanisms that rely on compression-based arguments, for instance, the Ladder Mechanism [26].

3.5 Empirical results

So far, we have established theoretically that similarity between classifiers allows us to evaluate a larger number of classifiers on the test set without overfitting. In this section, we investigate whether these improvements already occur in the regime of contemporary machine learning. We specifically focus on ImageNet and CIFAR-10, two widely used machine learning benchmarks that have recently been shown to exhibit little to no adaptive overfitting in spite of almost a decade of test set re-use [190]. For both datasets, we empirically measure two main quantities: (i) The similarity between a wide range of models, some of them arising from hyperparameter search experiments. (ii) The resulting increase in the number of models we can evaluate in a non-adaptive setting compared to a baseline that does not utilize the model similarities.

Similarities on Imagenet

We utilize the model testbed from Recht et al. [190],[1] who collected a dataset of 66 image classifiers that includes a wide range of standard ImageNet models such as AlexNet [126], ResNets [96], DenseNets [108], VGG [213], Inception [221], and several other models. As a baseline for the observed similarities between these models, we compare them to classifiers with the same accuracy but otherwise random predictions: given two models f_1 and f_2 with population error rates μ_1 and μ_2, we know that the similarity $\mathbb{P}(\mathbb{1}\{f_1(x) \neq y\} = \mathbb{1}\{f_2(x) \neq y\})$ equals $\mu_1\mu_2 + (1-\mu_1)(1-\mu_2)$ if the random variables $\mathbb{1}\{f_1(x) \neq y\}$ and $\mathbb{1}\{f_2(x) \neq y\}$ are independent. Figure 3.1a in the introduction shows these model similarities assuming the models make independent mistakes and also the empirical data for the $\binom{66}{2} = 2{,}145$ pairs of models. We see that the empirical similarities are significantly higher than the random baseline (mean 0.85 vs 0.62).

The corresponding Figure 3.1b shows two lower bounds on the number of models that can be evaluated for the empirical ImageNet data. In particular, we use $n = 50{,}000$ (the size of the ImageNet validation set) and a target probability $\delta = 0.05$ for the overfitting event (3.1) with error $\varepsilon = 0.01$. We compare two methods for computing the number of non-adaptively testable models: a guarantee based on the simple union bound (3.2) and a guarantee based on our more refined union bound derived from our theoretical analysis in Section 3.3. Later in this section, we introduce an even stronger bound that utilizes higher-order interactions between the model similarities and yields significantly larger improvements under an assumption on the structure among the classifiers.

[1] Available at `https://github.com/modestyachts/ImageNetV2`.

To obtain meaningful quantities in the regime of ImageNet, all bounds here require significantly sharper numerical calculations than the standard theoretical tools such as Chernoff bounds. We now describe these calculations at a high level and defer the details to Appendix 3.8. After introducing the three methods, we compare them on the ImageNet data.

Standard union bound. Given n, ε, and the population error rate of all models $\mathbb{E}_{\mathcal{D}}[q_i]$, we can compute the right hand side of (3.2) exactly.[2] It is well known that higher accuracies lead to smaller probability of error and hence allow for a larger number of test set reuses. We assume all models have population accuracy 75.6%, the average top-1 accuracy of the 66 Imagenet models. In this case, the vanilla union bound (3.2) guarantees that $k = 257{,}397$ models can be evaluated on a test set of size 50,000 so that their empirical accuracies would lie in the confidence interval 0.756 ± 0.01 with probability at least 95%.

Similarity Union Bound. While the union bound (3.2) is easy to use, it does not leverage the dependencies between the random variables $\mathbb{1}\{f_i(x) \neq y\}$ for $i \in \{1, 2, \dots k\}$. To exploit this property, we utilize the refined union bound (3.3) which is guaranteed to be an improvement over (3.2) when the parameter t is optimized. In order to use (3.3), we must compute the probabilities

$$\mathbb{P}\left(\{\mathbb{E}_S[q_2] - \mathbb{E}_{\mathcal{D}}[q_2] \leq \alpha_2\} \cap \{\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] \geq \alpha_1\}\right) \tag{3.7}$$

for given α_1, α_2, $\mathbb{E}_{\mathcal{D}}[q_1]$, $\mathbb{E}_{\mathcal{D}}[q_2]$, and similarity $\mathbb{P}(q_1(z) = q_2(z))$. In Appendix 3.8, we show that we can compute these probabilities efficiently by assigning success probabilities to three independent Bernoulli random variables X_1, X_2, and W such that $(X_1 W, X_2 W)$ is equal to $(q_1(z), q_2(z))$ in distribution. Let $p_w := \mathbb{P}(W = 1)$. Then, given i.i.d. draws X_{1i}, X_{2i}, and W_i, we condition on the values of W_i to express probability (3.7) as

$$\begin{aligned} &\mathbb{P}\left(\{\mathbb{E}_S[q_2] - \mathbb{E}_{\mathcal{D}}[q_2] \leq \alpha_2\} \cap \{\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] \geq \alpha_1\}\right) \\ &= \sum_{j=0}^{n} \binom{n}{j} p_w^j (1-p_w)^{n-j} \mathbb{P}\left(\sum_{i=1}^{j} X_{2i} \leq \lfloor n(p_2 + \alpha_2) \rfloor\right) \mathbb{P}\left(\sum_{i=1}^{j} X_{1i} \geq \lceil n(p_1 + \alpha_1) \rceil\right). \end{aligned} \tag{3.8}$$

We refer the reader to Appendix 3.8 for more details. The two tail probabilities for X_{1i} and X_{2i} can be computed efficiently with the use of beta functions. Using (3.8) and (3.3) with a binary search over t, we can compute the probability of making an error ε when estimating the population error rates of k models with given error rates and pairwise similarities. Figure 3.1b shows the maximum number of models k that can be evaluated on the same test set so that the probability of making an $\varepsilon = 0.01$ error in estimating all their error rates is at most 0.05 when the models satisfy $\mathbb{E}_{\mathcal{D}}[q_i] = 0.244$ and $\mathbb{P}(q_i(z) = q_j(z)) \geq 0.85$ for all $1 \leq i, j \leq k$. The figure shows that our new bound offers a significant improvement over the guarantee given by the standard union bound (3.2).

[2] After an additional union bound to decouple the left and right tails.

Similarity union bound with a Naive Bayes assumption. While the previous computation uses the pairwise similarities observed empirically to offer an improved guarantee on the number of allowed test set reuses, it does not take into account higher order dependencies between the models. In particular, Figure 3.4 in Appendix 3.9 shows that 27.8% of test images are correctly classified by all the models, 55.9% of test images are correctly classified by 60 of the 66 models considered, and 4.7% of test images are incorrectly classified by all the models. We now show how this kind of agreement between models enables a larger number of test set reuses. Inspired by the coupling used in (3.8), we make the following assumption.

Assumption A1 (Naive Bayes)**.** *Let q_1, q_2, $\dots q_k$ be a collection of queries such that $\mathbb{E}_{\mathcal{D}}[q_i] = p$ and $\mathbb{P}(q_i(z) = q_j(z)) = \eta$ for some p and η, for all $1 \le i, j \le k$. We say such a collection has a Naive Bayes structure if there exist p_x and p_w in $[0, 1]$ such that $(q_1(z), q_2(z), \dots, q_k(z))$ is equal to $(X_1 W, X_2 W, \dots, X_k W)$ in distribution, where W, X_1, $\dots X_k$ are independent Bernoulli random variables with $\mathbb{P}(W = 1) = p_w$ and $\mathbb{P}(X_i = 1) = p_x$ for all $1 \le i \le k$.*

Intuitively, a collection of queries $\mathbb{1}\{f_i(x) \neq y\}$ has a Naive Bayes structure if the data distribution $\mathcal{D}$ generates easy examples (x, y) with probability $1 - p_w$ such that all the models f_i classify correctly, and if an example is not easy, the models make mistakes independently. As mentioned before, Figure 3.4 supports the existence of such an easy set. When a test point in the ImageNet test set is not an easy example, the models do not make mistakes independently. Therefore, Assumption A1 is not exactly satisfied by existing ImageNet models. However, we know that independent Bernoulli trials saturate the standard union bound (3.2). This effect can also be observed in Figure 3.2. As the similarity between the models decreases, i.e. $1 - p_w$ decreases, the models make mistakes independently and the guarantee with Assumption A1 converges to the standard union bound guarantee. So while Assumption A1 is not exactly satisfied in practice, the violation among the ImageNet classifiers likely implies an even better lower bound on the number of testable models.

Assumption A1 is computationally advantageous. It allows us to compute the overfitting probability (3.1) exactly, as we detail in Appendix 3.8. Figure 3.2 is an extension of Figure 3.1b; it shows the relative improvement of our bounds over the standard union bound in terms of the number of testable models when $\varepsilon = 0.01$ and $\delta = 0.01$. Moreover, Figure 3.2 also shows that the relative improvement of our bounds increases quickly with ε. According to Figure 3.2, Assumption A1 implies that we can evaluate 10^8 models on the test set in the regime of ImageNet without overfitting. While this number of models might seem unnecessarily large, in Section 3.4 we saw that when models are chosen adaptively we must consider a tree of possible models, which can easily contain 10^8 models.

Similarities on CIFAR-10

Practitioners often evaluate many more models than the handful that ultimately appear in publication. The choice of architecture is the result of a long period of iterative refinement,

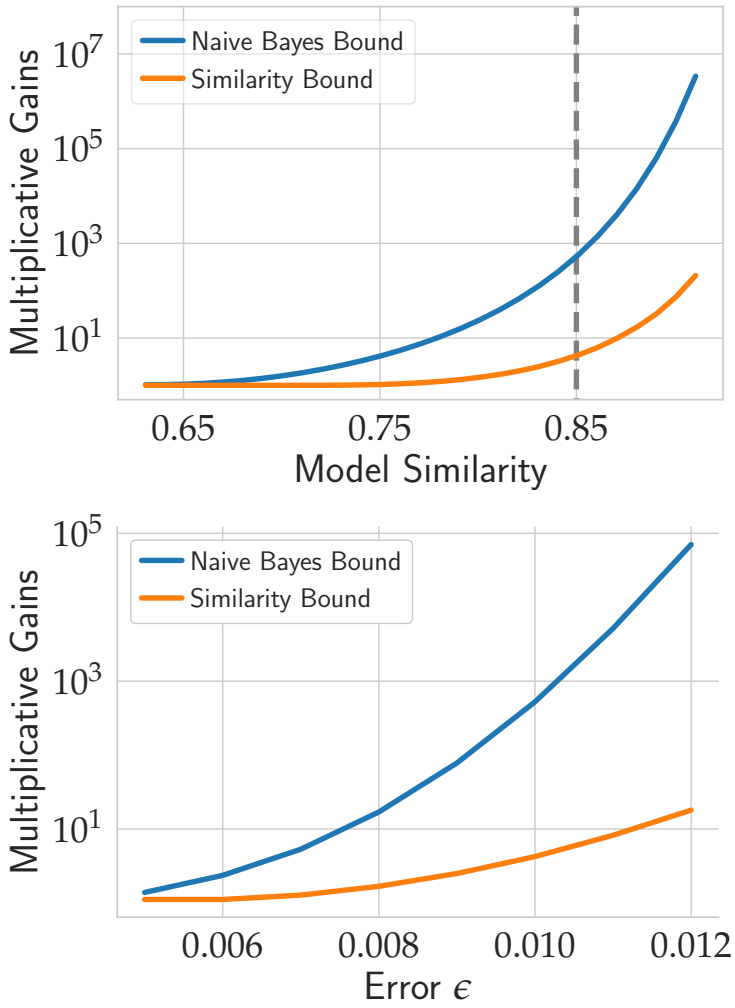


Figure 3.2: Left figure shows the multiplicative gains in the number of testable models, as a function of model similarity, over the guarantee offered by the standard union plus binomial bound, with $\varepsilon = 0.01$ and $\delta = 0.05$. Right figure shows the same multiplicative gains, but as a function of ε, when $\delta = 0.05$ and the pairwise similarity is $\eta = 0.85$.

and the hyperparameters for any fixed architecture are often chosen by evaluating a large grid of plausible models. Using data from CIFAR-10, we demonstrate these common practices both generate large classes of very similar models.

Random hyperparameter search. To understand the similarity between models evaluated in hyperparameter search, we ran our own random search to choose hyperparameters for a ResNet-110. The grid included properties of the architecture (e.g. type of residual block), the optimization algorithm (e.g. choice of optimizer), and the data distribution (e.g. data

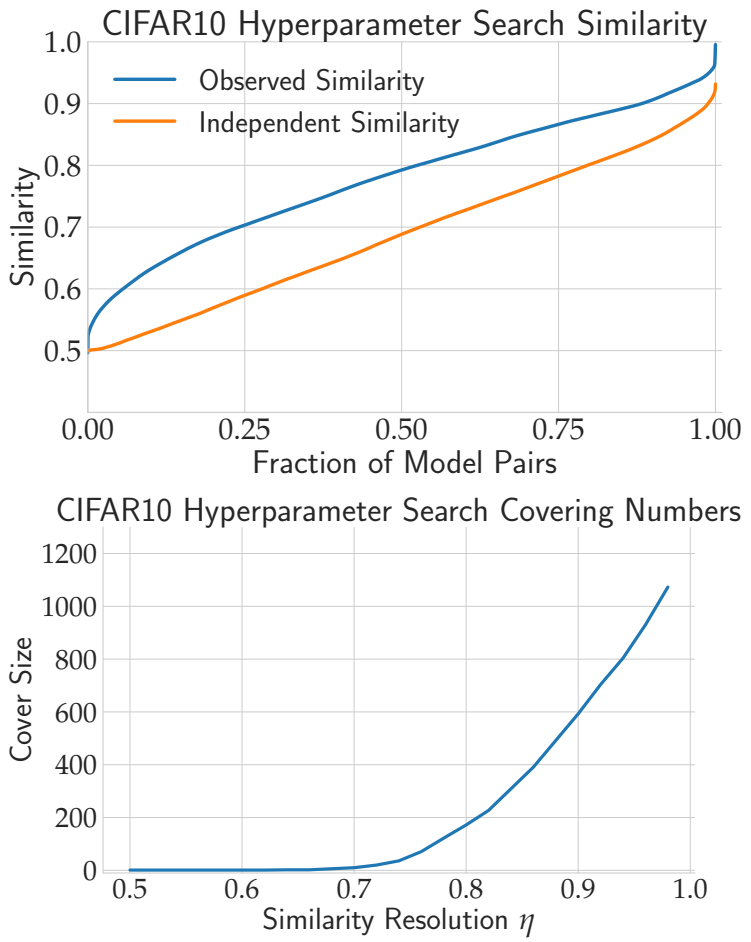


Figure 3.3: Model similarities and covering numbers for random hyperparameter search on CIFAR10.

augmentation strategies). A full specification of the grid is included in Appendix 3.10. We sample and train 320 models, and, for each model, we select 10 checkpoints evenly spaced throughout training. The best model considered achieves accuracy of 96.6%, and, after restricting to models with accuracy at least 50%, we are left with 1,235 model checkpoints. In Figure 3.3, we show the similarity for each pair of checkpoints and compute an upper bound on the corresponding similarity covering number $N_\eta(\mathcal{F})$ for each possible value of η. As in the case of ImageNet, CIFAR10 models found by random search are significantly more similar than random chance would suggest.

Neural architecture search. In the random search experiment, all of the models were chosen non-adaptively—the grid of models is fixed in advance. However, similarity protects against overfitting also in the adaptive setting. To illustrate this, we compute the similarity

Table 3.1: Neural Architecture Search Similarities

Models	Mean Accuracy	Mean Similarity	Increase in Testable Models	
			SB	NBB
20 Random	96.8%	97.5%	$\mathbf{9.9}\times$	$\mathbf{1.6 \cdot 10^4}\times$
20 Highest Scoring	96.9%	97.6%	$\mathbf{12.0}\times$	$\mathbf{3.4 \cdot 10^4}\times$

for models evaluated by automatic neural architecture search. In particular, we ran the DARTS neural architecture search pipeline to adaptively evaluate a large number of plausible models in search of promising configurations [137, 143]. In Table 3.1, we report the mean accuracies and pairwise similarities for 20 randomly selected configurations evaluated by DARTS, as well as the top 20 scoring configurations according to DARTS internal scoring mechanism. Table 3.1 also shows the multiplicative gains in the number of testable models offered by our similarity bound (SB) and our naive Bayes bound (NBB) over the standard union bound are between one and four orders of magnitude. Therefore, even in a high accuracy regime we can guarantee a significantly higher number of test set reuses without overfitting when taking into account model similarities.

3.6 Conclusions and future work

We have shown that contemporary image classification models are highly similar, and that this similarity increases the longevity of the test set both in theory and in experiment. It is worth noting that model similarity does not preclude progress on the test set: two models that are 85% similar can differ by as much as 15% in accuracy (for context: the top-5 accuracy improvement from the seminal AlexNet to the current state of the art on ImageNet is about 17%). In addition, it is well known that higher model accuracy implies a larger number of test set reuses without overfitting. So as the machine learning practitioner explores increasingly better performing models that also become more similar, it can actually become *harder* to overfit.

There are multiple important avenues for future work. First, one natural question is *why* the classification models turn out to be so similar. In addition, it would be insightful to understand whether the similarity phenomenon is specific to image classification or also arises in other classification tasks. There may also be further structural dependencies between models that mitigate the amount of overfitting. Finally, it would be ideal to have a statistical procedure that leverages such model structure to provide reliable and accurate performance bounds for test set re-use.

3.7 Appendix: Proofs for Section 3.3

Lemma 4. *Suppose X_i are i.i.d. discrete random variables which take values -1, 0, and 1 with probabilities p_{-1}, p_0, and p_1 respectively, and hence $\mathbb{E}X_i = p_1 - p_{-1}$. Then, for any $t \geq 0$ such that $p_1 - p_{-1} + t/2 \geq 0$ we have*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > p_1 - p_{-1} + t\right) \leq e^{-\frac{nt}{2}\log\left(1+\frac{t}{2p_1}\right)}.$$

Proof. We assume $p_1 > 0$. The result follows by continuity when $p_1 = 0$. We prove the more general case since the first part of the lemma is a particular case. By standard Chernoff methods we have

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > p_1 - p_{-1} + t\right) \leq e^{-n\lambda(t+p_1-p_{-1})}\left(p_0 + p_1 e^{\lambda} + p_{-1}e^{-\lambda}\right)^n,$$

for any $\lambda \in [0, \infty)$. Let $r > 0$ to be chosen later. Now, we would like to choose λ to be nonnegative and as large as possible so that

$$p_0 + p_1 e^{\lambda} + p_{-1}e^{-\lambda} \leq e^{\lambda r}. \tag{3.9}$$

By changing variables to $e^{\lambda} = z + 1$ for some $z \geq 0$ we want to find z as large as possible so that

$$p_0(z+1) + p_1(z+1)^2 + p_{-1} \leq (z+1)^{1+r}.$$

Then, by Bernoulli's inequality it suffices if z satisfies the inequality

$$p_0(z+1) + p_1(z+1)^2 + p_{-1} \leq 1 + (1+r)\,z,$$

which is equivalent to

$$p_0 + p_1 z + 2p_1 \leq 1 + r.$$

Hence, the desired inequality (3.9) is satisfied if $z \leq \frac{p_{-1}-p_1+r}{p_1}$, which can be satisfied by choosing $z = \frac{p_{-1}-p_1+r}{p_1}$ when $p_{-1} - p_1 + r \geq 0$. In this case, we would be able to set $\lambda = \log\left(1 + \frac{p_{-1}-p_1+r}{p_1}\right)$ and obtain

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > p_1 - p_{-1} + t\right) \leq e^{-n\log\left(1+\frac{p_{-1}-p_1+r}{p_1}\right)(t+p_1-p_{-1}-r)}.$$

Set $r = p_1 - p_{-1} + t/2$ and by the assumption on t we are guaranteed that $r \geq 0$ and $p_{-1} - p_1 + r \geq 0$. The conclusion follows. □

Theorem 2. *Let $\mathcal{F} = \{q_1, q_2, \ldots, q_k\}$ be a collection of queries $q_i \colon \mathcal{Z} \to \{0,1\}$ independent of the test set $\{z_1, z_2, \ldots, z_n\}$. Then, for any $\eta \in [0,1]$ we have*

$$\mathbb{P}\left(\max_{1\le i\le k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \ge \varepsilon\right) \;\le\; 2N_\eta e^{-\frac{n\varepsilon^2}{2}} + 2ke^{-\frac{n\varepsilon}{4}\log\left(1+\frac{\varepsilon}{4(1-\eta)}\right)}. \tag{3.4}$$

Then, for all $\eta \le 1 - \max\left\{\frac{2\log(4k/\delta)}{n}, \sqrt{\frac{\log(4N_\eta/\delta)}{2n}}\right\}$, we have with probability $1-\delta$

$$\max_{1\le i\le k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \le \max\left\{\sqrt{\frac{2\log(4N_\eta/\delta)}{n}}, \sqrt{\frac{32(1-\eta)\log(4k/\delta)}{n}}\right\}. \tag{3.5}$$

Moreover, if $\varepsilon = \sqrt{\frac{\log((2N_\eta+1)/\delta)}{n}}$ and $\eta \ge 1 - \frac{\varepsilon}{4\left(e^{2\varepsilon}(2k)^{\frac{4}{n\varepsilon}}-1\right)}$, we have with probability $1-\delta$

$$\max_{1\le i\le k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \le \varepsilon. \tag{3.6}$$

Proof. First we prove (3.4) and we start with the right tails. We have

$$\mathbb{P}\left(\bigcup_{i=1}^{k}\{\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i] \ge \varepsilon\}\right) \le \mathbb{P}\left(\bigcup_{i=1}^{k}\{\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i] \ge \varepsilon\} \bigcup_{\widetilde{q}\in M}\{\mathbb{E}_S[\widetilde{q}] - \mathbb{E}_{\mathcal{D}}[\widetilde{q}] \ge \varepsilon\}\right),$$

where M is a minimal η similarity cover of $\mathcal{F}$. Then, there exists a partition of $\mathcal{F}$ into subsets $R_{\widetilde{q}}$, with $\widetilde{q} \in M$, such that for any $q \in \mathcal{F}$ there exists $\widetilde{q}$ such that $q \in R_{\widetilde{q}}$, $\mathbb{E}_{\mathcal{D}}[q] \ge \mathbb{E}_{\mathcal{D}}[\widetilde{q}]$, and $\mathbb{P}(q(z) = \widetilde{q}(z)) \ge \eta$. Since $R_{\widetilde{q}}$ is a partition of $\mathcal{F}$, we have $\sum_{\widetilde{q}\in M} |R_{\widetilde{q}}| = k$. Therefore, following the same argument as in (3.3), we have

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{i=1}^{k}\{\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i] \ge \varepsilon\}\right) &\le \sum_{\widetilde{q}\in M}\mathbb{P}\left(\mathbb{E}_S[\widetilde{q}] - \mathbb{E}_{\mathcal{D}}[\widetilde{q}] \ge \frac{\varepsilon}{2}\right) \\
&\quad + \sum_{\widetilde{q}\in M}\sum_{q\in R_{\widetilde{q}}}\mathbb{P}\left(\{\mathbb{E}_S[\widetilde{q}] - \mathbb{E}_{\mathcal{D}}[\widetilde{q}] \le \varepsilon/2\} \cap \{\mathbb{E}_S[q] - \mathbb{E}_{\mathcal{D}}[q] \ge \varepsilon\}\right) \\
&\le \sum_{\widetilde{q}\in M}\mathbb{P}\left(\mathbb{E}_S[\widetilde{q}] - \mathbb{E}_{\mathcal{D}}[\widetilde{q}] \ge \frac{\varepsilon}{2}\right) \\
&\quad + \sum_{\widetilde{q}\in M}\sum_{q\in R_{\widetilde{q}}}\mathbb{P}\left(\mathbb{E}_S[\widetilde{q}] - \mathbb{E}_{\mathcal{D}}[\widetilde{q}] + \varepsilon/2 \le \mathbb{E}_S[q] - \mathbb{E}_{\mathcal{D}}[q]\right).
\end{aligned}$$

Now, for every $\widetilde{q} \in M$ and any $q \in R_{\widetilde{q}}$ we use a standard Chernoff bound and Lemma 4 to show

$$\mathbb{P}\left(\mathbb{E}_S[\widetilde{q}] - \mathbb{E}_{\mathcal{D}}[\widetilde{q}] \ge \frac{\varepsilon}{2}\right) \le e^{-\frac{n\varepsilon^2}{2}} \;\text{ and }\; \mathbb{P}\left(\mathbb{E}_S[\widetilde{q}] - \mathbb{E}_{\mathcal{D}}[\widetilde{q}] + \varepsilon/2 \le \mathbb{E}_S[q] - \mathbb{E}_{\mathcal{D}}[q]\right) \le e^{-\frac{n\varepsilon}{4}\log\left(1+\frac{\varepsilon}{4(1-\eta)}\right)}.$$

To see why we can apply Lemma 4 note that $q(z) - \widetilde{q}(z)$ takes values in $\{-1, 0, 1\}$ with the probability of 0 being at least η, and $\mathbb{E}_{\mathcal{D}}[q - \widetilde{q}] \geq 0$ by the choice of the covering set. Since $|M| = N_\eta$ and since $\sum_{\widetilde{q} \in M} |R_{\widetilde{q}}| = k$, we find

$$\mathbb{P}\left(\bigcup_{i=1}^{k} \{\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i] \geq \varepsilon\}\right) \leq N_\eta e^{-\frac{n\varepsilon^2}{2}} + k e^{-\frac{n\varepsilon}{4}\log\left(1+\frac{\varepsilon}{4(1-\eta)}\right)}.$$

An analogous argument for the left tails yields (3.4). Now, we turn to showing (3.5). The goal is to find ε such that

$$2N_\eta e^{-\frac{n\varepsilon^2}{2}} \leq \frac{\delta}{2} \quad \text{and} \quad 2k e^{-\frac{n\varepsilon}{4}\log\left(1+\frac{\varepsilon}{4(1-\eta)}\right)} \leq \frac{\delta}{2}. \tag{3.10}$$

The first inequality is satisfied if $\varepsilon \geq \sqrt{\frac{2\log(4N_\eta/\delta)}{n}}$. To find ε that satisfies the second condition we make use of the inequality $\log(1+t) \geq \frac{t}{t+1}$ for all $t \geq 0$. We search for ε that also satisfies $\varepsilon \leq 4(1-\eta)$. Then,

$$\frac{n\varepsilon}{4}\log\left(1 + \frac{\varepsilon}{4(1-\eta)}\right) \geq \frac{n\varepsilon^2}{32(1-\eta)},$$

and we would like the right hand side to be at least $\log(4k/\delta)$. If we choose

$$\varepsilon = \max\left\{\sqrt{\frac{2\log(4N_\eta/\delta)}{n}}, \sqrt{\frac{32(1-\eta)\log(4k/\delta)}{n}}\right\},$$

the condition $\varepsilon \leq 4(1-\eta)$ is satisfied because of the assumption on η. In this case, both conditions (3.10) are satisfied and (3.5) is proven. Finally, note that when $\eta \geq 1 - \frac{\varepsilon}{4(e^{2\varepsilon}(2k)^{\frac{4}{n\varepsilon}} - 1)}$ we have

$$2k e^{-\frac{n\varepsilon}{4}\log\left(1+\frac{\varepsilon}{4(1-\eta)}\right)} \leq e^{-\frac{n\varepsilon^2}{2}}.$$

Then, (3.6) follows by choosing $\varepsilon = \sqrt{\frac{\log((2N_\eta+1)/\delta)}{n}}$. This completes the proof.

□

Corollary 3. *Let $\mathcal{F} = \{q_1, q_2, \ldots, q_k\}$ be a collection of k queries $q_i \colon \mathcal{Z} \to \{0, 1\}$ fixed independently of the test set. Choose $\eta_\star$ so that $N_{\eta_\star} = 1$. Suppose $n \geq c_1 \max\{\varepsilon^{-1}, \varepsilon^{-2}\}$ and the number of queries k satisfies*

$$k \leq \frac{c_2 \varepsilon}{(1 - \eta_\star)}$$

for positive constants c_1, c_2. Then, with probability $3/4$, $\max_{1 \leq i \leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \leq \varepsilon$.

Proof. Choose $\eta_\star$ so that $N_{\eta_\star} = 1$. Using equation (3.4) from Theorem (2),

$$\mathbb{P}\left(\max_{1\le i\le k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \ge \varepsilon\right) \;\le\; 2N_\eta e^{-\frac{n\varepsilon^2}{2}} + 2ke^{-\frac{n\varepsilon}{4}\log\left(1+\frac{\varepsilon}{4(1-\eta)}\right)}.$$

Consider each term separately. If $n \ge \frac{\log(4/\delta)}{\varepsilon^2}$, then the first term $2N_{\eta_\star}e^{-n\varepsilon^2/2} \le \delta/2$. Now, we choose k so that the second term is at most $\delta/2$, i.e.

$$2ke^{-\frac{n\varepsilon}{4}\log\left(1+\frac{\varepsilon}{4(1-\eta_\star)}\right)} \le \frac{\delta}{2},$$

which is equivalent to requiring

$$k \le \frac{\delta}{4}\left(1 + \frac{\varepsilon}{4(1-\eta_\star)}\right)^{n\varepsilon/4}.$$

If $n \ge \frac{4}{\epsilon}$, then the right hand side can be lower bounded by

$$\frac{\delta}{4}\left(1 + \frac{\varepsilon}{4(1-\eta)}\right)^{n\varepsilon/4} \ge \frac{\delta}{4}\left(1 + \frac{\varepsilon}{4(1-\eta_\star)}\right) \ge \frac{\delta}{4}\left(\frac{\varepsilon}{4(1-\eta_\star)}\right),$$

and the conclusion follows from plugging in $\delta = 1/4$. □

3.8 Appendix: Tail probability of two dependent binomials

In this section we detail the computations of the two similarity union bounds (with and without the Naive Bayes assumption).

Similarity Union Bound. We wish to compute the probability

$$\mathbb{P}\left(\{\mathbb{E}_S[q_2] - \mathbb{E}_{\mathcal{D}}[q_2] \le \alpha_2\} \cap \{\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] \ge \alpha_1\}\right),$$

where $q_1(z)$ and $q_2(z)$ have some joint distribution over $\{0,1\}^2$. Let use denote $p_1 = \mathbb{E}_{\mathcal{D}}[q_1]$, $p_2 = \mathbb{E}_{\mathcal{D}}[q_2]$, and $\eta = \mathbb{P}(q_1(z) = q_2(z))$ respectively. These three quantities fully determine the joint probability distribution of $q_1(z)$ and $q_2(z)$. Specifically, we have

$$\mathbb{P}(q_1(z)=1, q_q(z)=1) = \frac{p_1+p_2+\eta-1}{2}, \qquad \mathbb{P}(q_1(z)=1, q_q(z)=0) = \frac{1+p_1-p_2-\eta}{2}$$
$$\mathbb{P}(q_1(z)=0, q_q(z)=1) = \frac{1+p_2-p_1-\eta}{2}, \qquad \mathbb{P}(q_1(z)=0, q_q(z)=0) = \frac{1+\eta-p_1-p_2}{2}.$$

We denote these four probabilities by p_{11}, p_{10}, p_{01}, and p_{00} respectively. We aim to find three independent Bernoulli random variables X_1, X_2, and W such that (X_1W, X_2W) equals

$(q_1(z), q_2(z))$ in distribution. It turns out we can achieve this whenever $p_{11} \geq (p_{10}+p_{11})(p_{01}+p_{11})$, a condition that is always satisfied in the settings we consider, by setting

$$\mathbb{P}(X_1 = 1) = \frac{p_{11}}{p_{01}+p_{11}}\,, \quad \mathbb{P}(X_2 = 1) = \frac{p_{11}}{p_{10}+p_{11}}\,, \quad \mathbb{P}(W = 1) = \frac{(p_{10}+p_{11})(p_{01}+p_{11})}{p_{11}}.$$

Then, given i.i.d. draws X_{1i}, X_{2i}, and W_i, probability (3.7) equals

$$\mathbb{P}\left(\left\{\sum_{i=1}^{n} X_{2i}W_i \leq \lfloor n(p_2+\alpha_2)\rfloor\right\} \bigcap \left\{\sum_{i=1}^{n} X_{1i}W_i \geq \lceil n(p_1+\alpha_1)\rceil\right\}\right).$$

Denote $p_w = \mathbb{P}(W = 1)$. Then, we condition on the possible values of W_i to obtain

$$\begin{aligned}
&\mathbb{P}\left(\{\mathbb{E}_S[q_2] - \mathbb{E}_{\mathcal{D}}[q_2] \leq \alpha_2\} \cap \{\mathbb{E}_S[q_1] - \mathbb{E}_{\mathcal{D}}[q_1] \geq \alpha_1\}\right) \\
&\quad = \sum_{j=0}^{n} \binom{n}{j} p_w^j (1-p_w)^{n-j} \mathbb{P}\left(\sum_{i=1}^{j} X_{2i} \leq \lfloor n(p_2+\alpha_2)\rfloor\right) \mathbb{P}\left(\sum_{i=1}^{j} X_{1i} \geq \lceil n(p_1+\alpha_1)\rceil\right).
\end{aligned}$$

The two tail probabilities for X_{1i} and X_{2i} can be computed efficiently with the use of beta functions.

Similarity union bound with a Naive Bayes assumption. In this section we wish to compute directly the overfitting probability

$$\mathbb{P}\left(\max_{1\leq i\leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \geq \varepsilon\right)$$

when the query vector $(q_1(z), q_2(z), \ldots, q_k(z))$ is equal in distribution to $(X_1W, X_2W, \ldots, X_kW)$ for some independent Bernoulli random variables W, $X_1, \ldots X_k$. Recall that we assume that all queries q_i have equal error rates $\mathbb{E}_{\mathcal{D}}[q_i]$; let us denote it $\mu = \mathbb{E}_{\mathcal{D}}[q_i]$. Moreover, for any two queries q_i and q_j we have $\mathbb{P}(q_i(z) = q_j(z)) = \eta$.

Suppose we are given i.i.d. draws W_i and i.i.d. draws $X_{\ell i}$ for $1 \leq i \leq n$ and $1 \leq \ell \leq k$. Then, if $p_w := \mathbb{P}(W = 1)$, by conditioning on the values of the random variables W_i we obtain

$$\mathbb{P}\left(\max_{1\leq i\leq k} |\mathbb{E}_S[q_i] - \mathbb{E}_{\mathcal{D}}[q_i]| \geq \varepsilon\right) = \sum_{j=1}^{n} \binom{n}{j} p_w^j (1-p_w)^{n-j} \mathbb{P}\left(\bigcup_{\ell=1}^{k} \left|\frac{1}{n}\sum_{i=1}^{j} X_{\ell i} - \mu\right| \geq \varepsilon\right).$$

The random variables $\sum_{i=1}^{j} X_{\ell i}$ have the same distribution for all ℓ and are independent. Then,

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{\ell=1}^{k} \left|\frac{1}{n}\sum_{i=1}^{j} X_{\ell i} - \mu\right| \geq \varepsilon\right) &= 1 - \mathbb{P}\left(\bigcap_{\ell=1}^{k} \left|\frac{1}{n}\sum_{i=1}^{j} X_{\ell i} - \mu\right| < \varepsilon\right) \\
&= 1 - \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{j} X_{1i} - \mu\right| < \varepsilon\right)^k.
\end{aligned}$$

Therefore, we have

$$\mathbb{P}\left(\max_{1\le i\le k}|\mathbb{E}_S[q_i]-\mathbb{E}_{\mathcal{D}}[q_i]|\ge\varepsilon\right)=\sum_{j=1}^{n}\binom{n}{j}p_w^j(1-p_w)^{n-j}\left[1-\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{j}X_{1i}-\mu\right|<\varepsilon\right)^k\right].$$

3.9 Appendix: Empirical distribution of image difficulty in ImageNet

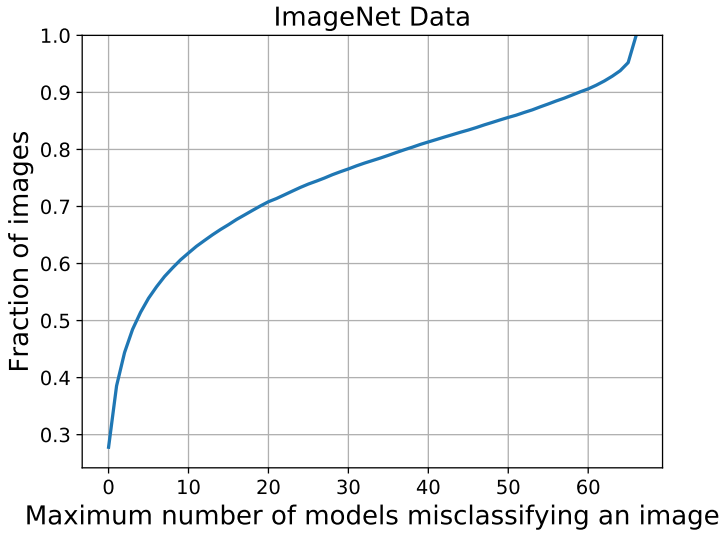


Figure 3.4: The empirical "difficulty" distribution of the 50,000 images in the ImageNet validation set as measured by the classifiers in the testbed from Recht et al. [190]. The plot shows how many of the images are misclassified by at most a certain number of the models. For instance, about 27.8% of the images are correctly classified by all models, and 55.9% of the images are correctly classified by 60 of the 66 models. 4.7% of the images are misclassified by all models. The plot shows that a significant fraction of images is classified correctly by all or almost all of the models. These empirical findings support the Naive Bayes assumption in Section 3.5.

3.10 Appendix: CIFAR-10 random hyperparameter grid search

We conducted a large hyperparameter search on a ResNet110. All of our experiments build on the ResNet implementation and training code provided by `https://github.com/hysts/`

`pytorch\_image\_classification`. In Table 3.2, we specify the grid used in the experiments. If not explicitly stated, all other hyperparameters are set to their default settings.

Table 3.2: Random grid search hyperparameters.

Parameter	Sampling Distribution
Number of base channels	Uniform$\{4, 8, 16, 32\}$
Residual block type	Uniform{"Basic", "Bottleneck"}
Remove ReLu before residual units	Uniform{True, False}
Add BatchNorm after last convolutions	Uniform{True, False}
Preactivation of shortcuts after downsampling	Uniform{True, False}
Batch size	Uniform{32, 64, 128, 256}
Base learning rate	Uniform[1e-4, 0.5]
Weight decay	$10^{\text{Uniform}[-5,-1]}$
Use weight decay with batch norm	Uniform{True, False}
Optimizer	Uniform{SGD, SGD with Momentum, Nesterov GD, Adam}
Momentum (SGD with momentum)	Uniform{0.6, 0.99}
β_1 (Adam)	Uniform[0.8, 0.95]
β_2 (Adam)	Uniform[0.9, 0.999]
Learning rate schedule	Uniform{Cosine, Fixed Decay}
Learning rate decay point 1 (Fixed Decay)	Uniform{40, 60, 80, 100}
Learning rate decay point 2 (Fixed Decay)	Uniform{120, 140, 160, 180}
Use random crops	Uniform{True, False}
Random crop padding	Uniform{2, 4, 8}
Use horizontal flips	Uniform{True, False}
Use cutout	Uniform{True, False}
Cutout size	Uniform{8, 12, 16}
Use dual cutout augmentation	Uniform{True, False}
Dual cutout α	Uniform[0.05, 0.3]
Use random erasing	Uniform{True, False}
Random erasing probability	Uniform[0.2, 0.8]
Use mixup data augmentation	Uniform{True, False}
Mixup α	Uniform[0.6, 1.4]
Use label smoothing	Uniform{True, False}
Label smoothing ϵ	Uniform[0.01, 0.2]

Chapter 4

The Strong Correlation Between Out-of-Distribution and In-Distribution Generalization

4.1 Introduction

The standard benchmark paradigm to measure generalization is to evaluate a model on a single test set drawn from the same distribution as the training set. But this paradigm provides only a narrow *in-distribution* performance guarantee: a small test error certifies future performance on new samples from exactly the same distribution as the training set. In many scenarios, it is hard or impossible to train a model on precisely the distribution it will be applied to. Hence a model will inevitably encounter *out-of-distribution* data on which its performance could vary widely compared to in-distribution performance. Understanding the performance of models beyond the benchmark training distribution raises the following fundamental question of external validity: how does out-of-distribution performance relate to in-distribution performance?

Classical theory for generalization across different distributions provides a partial answer [155, 16]. For a model f trained on a distribution D, known guarantees typically relate the in-distribution test accuracy on D to the out-of-distribution test accuracy on a new distribution D' via inequalities of the form

$$|\mathrm{acc}_D(f) - \mathrm{acc}_{D'}(f)| \;\leq\; d(D, D')$$

where d is a distance between the distributions D and D' such as the total variation distance. Qualitatively, these bounds suggest that out-of-distribution accuracy may vary widely as a function of in-distribution accuracy unless the distribution distance d is small and the accuracies are therefore close (see Figure 1 (top-left) for an illustration). More recently, empirical studies have shown that in some settings, models with similar in-distribution performance can indeed have different out-of-distribution performance [158, 261, 59].

In Chapter 2, we observed models from the SQuAD benchmark showed a much more regular pattern. The new out-of-distribution accuracies were almost perfectly linearly correlated with the original in-distribution accuracies for a range of deep neural networks. This trend was also observed in other dataset reproduction experiments involving CIFAR-10, MNIST, and ImageNet [190, 248, 147]. Importantly, this correlation holds *despite the substantial gap between in-distribution and out-of-distribution accuracies* (see Figure 1 (top-middle) for an example). However, it is currently unclear how widely these linear trends apply since they have been mainly observed for dataset reproductions and common variations of convolutional neural networks.

In this chapter, we conduct a broad empirical investigation to characterize when precise linear trends such as in Figure 1 (top-middle) may be expected, and when out-of-distribution performance is less predictable as in Figure 1 (top-left). Concretely, we make the following contributions:

- We show that precise linear trends occur on several datasets and associated distribution shifts (see Figure 1). Going beyond the dataset reproductions in earlier work, we find linear trends on
  - popular image classification benchmarks (CIFAR-10 [124], CIFAR-10.1 [190], CIFAR-10.2 [147], CIFAR-10-C [98], CINIC-10 [61], STL-10 [52], ImageNet [62], ImageNet-V2 [190]),
  - a pose estimation testbed based on YCB-Objects [39],
  - and two distribution shifts derived from concrete applications of image classification: satellite imagery and wildlife photos via the FMoW-WILDS and iWildCam-WILDS variants from WILDS [50, 14, 120].

- We show that the linear trends hold for many models ranging from state-of-the-art methods such as convolutional neural networks, visual transformers, and self-supervised models, to classical methods like logistic regression, nearest neighbors, and kernel machines. Importantly, we find that classical methods follow the same linear trend as more recent deep learning architectures. Moreover, we demonstrate that varying model or training hyperparameters, training set size, and training duration all result in models that follow the same linear trend.

- We also identify three settings in which the linear trends do *not* occur or are less regular: some of the synthetic distribution shifts in CIFAR-10-C (e.g., Gaussian noise), the Camelyon17-WILDS shift of tissue slides from different hospitals, and a version of the aforementioned iWildCam-WILDS wildlife classification problem with a different in-distribution train-test split [14]. We analyze these cases in detail via additional experiments to pinpoint possible causes of the linear trends.

- Pre-training a model on a larger and more diverse dataset offers a possibility to increase robustness. Hence we evaluate a range of models pre-trained on other datasets to study

the impact of pre-training on the linear trends. Interestingly, even pre-trained models sometimes follow the same linear trends as models trained only on the in-distribution training set. Two examples are ImageNet pre-trained models evaluated on CIFAR-10 and FMoW-WILDS. In other cases (e.g., iWildCam-WILDS), pre-training yields clearly different relationships between in-distribution and out-of-distribution accuracies.

- As a starting point for theory development, we provide a candidate theory based on a simple Gaussian data model. Despite its simplicity, this data model correctly identifies the covariance structure of the distribution shift as one property affecting the performance correlation on the Gaussian noise corruption from CIFAR-10-C.

Overall, our results show a striking linear correlation between the in-distribution and out-of-distribution performance of many models on multiple distribution shifts. This raises the intriguing possibility that, despite their different creation mechanisms, a diverse range of distribution shifts may share common phenomena. In particular, improving in-distribution performance reliably improves out-of-distribution performance as well. However, it is currently unclear whether improving in-distribution performance is the only way, or even the best way, to improve out-of-distribution performance. More research is needed to understand the extent of the linear trends observed in this work and whether robustness interventions can improve over the baseline given by empirical risk minimization. We hope that our work serves as a step towards a better understanding of distribution shift and how we can train models that perform robustly out-of-distribution.

Chapter outline. Next, we introduce our main experimental framework that forms the backbone of our investigation. The following sections instantiate this framework for multiple distribution shifts.

Section 4.3 shows results on a wide range of distribution shifts where precise linear trends do occur. Section 4.4 then turns to distribution shifts where the linear trends are less regular or do not exist. Section 4.5 investigates the role of pretraining in more detail since models pre-trained on a different dataset sometimes – but not always – deviate from the linear trends.

After our experiments, we briefly summarize the empirical phenomena in Section 4.6 and then present our theoretical model in Section 4.7. Section 4.8 describes related work and Section 4.9 concludes with a discussion of our results, possible implications for research on reliable machine learning, and directions for future work.

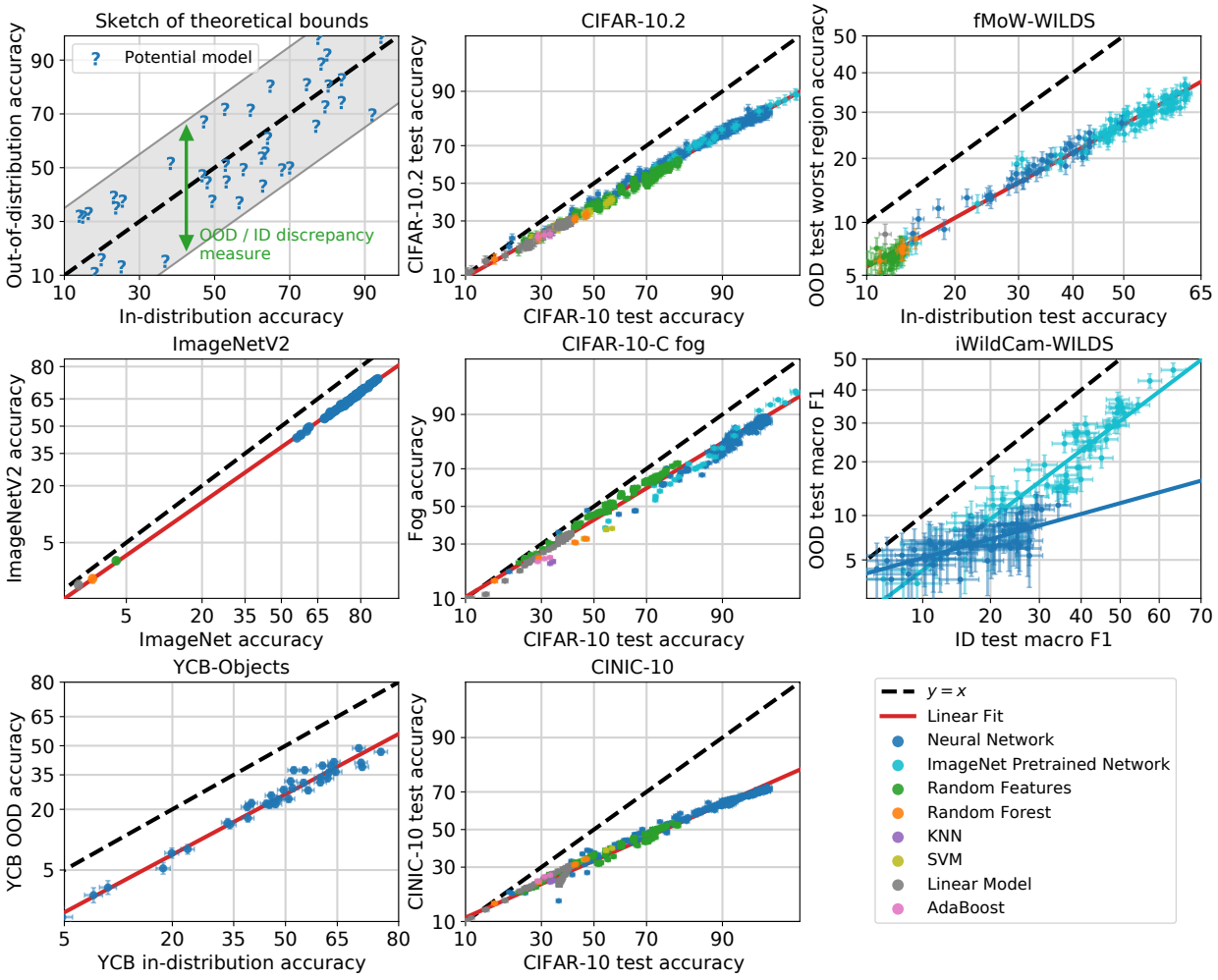


Figure 4.1: Out-of-distribution accuracies vs. in-distribution accuracies for a wide range of models, datasets, and distribution shifts. **Top left:** A sketch of the current bounds from domain adaptation theory. These bounds depend on distributional distances between in-distribution and out-of-distribution data, and they are loose in that they limit the deviation away from the y = x diagonal but do not prescribe a specific trend within these wide bounds (see Section 4.8). **Remaining panels:** In contrast, we show that for a wide range of models and datasets, there is a precise linear trend between out-of-distribution accuracy and in-distribution accuracy. Unlike what we might expect from theory, the linear trend does not follow the $y = x$ diagonal. The different panels represent different pairs of in-distribution and out-of-distribution datasets. Within each panel, we plot the performances of many different models, with different model architectures and hyperparameters. These datasets capture a variety of distribution shifts from dataset reproduction (CIFAR-10.2, ImageNet-V2); a real-world spatiotemporal distribution shift on satellite imagery (FMoW-WILDS); using a different benchmark test dataset (CINIC-10); synthetic perturbations (CIFAR-10-C and YCB-Objects); and a real-world geographic shift in wildlife monitoring (iWildCam-WILDS). Interestingly, for iWildCam-WILDS, models pretrained on ImageNet follow a different linear trend than models trained from scratch in-distribution, and we plot a separate trend line for ImageNet pretrained models in the iWildCam-WILDS panel.

4.2 Experimental setup

In each of our main experiments, we compare performance on two data distributions. The first is the training distribution D, which we refer to as "in-distribution" (ID). Unless noted otherwise, all models are trained only on samples from D (the main exception is pre-training on a different distribution). We compute ID performance via a held-out test set sampled from D. The second distribution is the "out-of-distribution" (OOD) distribution D' that we also evaluate the models on. For a loss function ℓ (e.g., error or accuracy), we denote the loss of model f on distribution D with $\ell_D(f) = \mathbb{E}_{x,y\sim D}\left[\ell(f(x), y)\right]$.

Experimental procedure. The goal of this chapter is to understand the relationship between $\ell_D(f)$ and $\ell_{D'}(f)$ for a wide range of models f (convolutional neural networks, kernel machines, etc.) and pairs of distributions D, D' (e.g., CIFAR-10 and the CIFAR-10.2 reproduction). Hence for each pair D, D', our core experiment follows three steps:

1. Train a set of models $\{f_1, f_2, \ldots\}$ on samples drawn from D. Apart from the shared training distribution, the models are trained independently with different training set sizes, model architectures, random seeds, optimization algorithms, etc.

2. Evaluate the trained models f_i on two test sets drawn from D and D', respectively.

3. Display the models f_i in a scatter plot with each model's two test accuracies on the two axes to inspect the resulting correlation.

An important aspect of our scatter plots is that we apply a non-linear transformation to each axis. Since we work with loss functions bounded in $[0, 1]$, we apply an axis scaling that maps $[0, 1]$ to $[-\infty, +\infty]$ via the probit transform. The probit transform is the inverse of the cumulative density function (CDF) of the standard Gaussian distribution, i.e., $l_{\text{transformed}} = \Phi^{-1}(l)$. Transformations like the probit or closely related logit transform are often used in statistics since a quantity bounded in $[0, 1]$ can only show linear trends for a bounded range. The linear trends we observe in our correlation plots are substantially more precise with the probit (or logit) axis scaling. Unless noted otherwise, each point in a scatter plot is a single model (not averaged over random seeds) and we show each point with 95% Clopper-Pearson confidence intervals for the accuracies.

We assembled a unified testbed that is shared across experiments and includes a multitude of models ranging from classical methods like nearest neighbors, kernel machines, and random forests to a variety of high-performance convolutional neural networks. Our experiments involved more than 3,000 trained models and 100,000 test set evaluations of these models and their training checkpoints. Due to the size of these experiments, we defer a detailed description of the testbed used to Appendix 4.10.

ID Dataset	OOD Dataset	R^2 of linear fit (probit domain)	Number of models evaluated
CIFAR-10	CIFAR-10.1	0.995	1,060
	CIFAR-10.2	0.997	1,060
	CINIC-10	0.991	949
	STL-10	0.995	456
	CIFAR-10-C `Fog`	0.990	790
	CIFAR-10-C `Brightness`	0.940	519
ImageNet	ImageNet-V2	0.996	219
YCB-Objects	YCB-Objects OOD	0.975	39
iWildCam-WILDS ID	iWildCam-WILDS OOD	0.881 (0.536)	66 (63)
FMoW-WILDS ID	FMoW-WILDS OOD	0.984	162

Table 4.1: Summary of ID and OOD pairs where we observe precise linear trends in our experiments. Number of models evaluated is the number of models trained on the ID training set, evaluated on the ID and OOD test sets, and used to compute the linear fits. For brevity, we list only two CIFAR-10-C shifts; see Appendix 4.12 for a complete list. As discussed in Section 4.5, on iWildCam-WILDS, ImageNet pretrained models exhibit a different linear trend than models trained from scratch. Therefore, for iWildCam-WILDS, we report the number of models and R^2 for pretrained models and models trained from scratch (given in parentheses) separately.

4.3 The linear trend phenomenon

In this section, we show precise linear trends between in-distribution and out-of-distribution performance occur across a diverse set of models, data domains, and distribution shifts. Moreover, the linear trends holds not just across variations in models and model architectures, but also across variation in model or training hyperparameters, training dataset size, and training duration.

Distribution shifts with linear trends

We find linear trends for models in our testbed trained on five different datasets—CIFAR-10, ImageNet, FMoW-WILDS, iWildCam-WILDS, and YCB-Objects—and evaluated on distribution shifts that fall into four broad categories. These trends are summarized in Table 4.1.

Dataset reproduction shifts. Dataset reproductions involve collecting a new test

set by closely matching the creation process of the original. Distribution shift arises as a result of subtle differences in the dataset construction pipelines. Recent examples of dataset reproductions are the CIFAR-10.1 and ImageNet-V2 test sets from Recht et al. [190], who observed linear trends for deep models on these shifts. In Figure 4.1, we extend this result and show both deep *and classical* models trained on CIFAR-10 and evaluated on CIFAR-10.2 [147] follow a linear trend. In Appendix 4.11, we further show linear trends occur for deep and classical CIFAR-10 models evaluated on CIFAR-10.1 and for ImageNet models evaluated on ImageNet-V2.

Distribution shifts between machine learning benchmarks. We also consider distribution shifts between distinct benchmarks which are drawn from different data sources, but which use a compatible set of labels. For instance, both CIFAR-10 and CINIC-10 [61] use the same set of labels, but CIFAR-10 is drawn from TinyImages [234] and CINIC-10 is drawn from ImageNet [62] images. We show CIFAR-10 models exhibit linear trends when evaluated on CINIC-10 (Figure 4.1) or on STL-10 [52] (Appendix 4.11).

Synthetic perturbations. Synthetic distribution shifts arise from applying a perturbation, such as adding Gaussian noise, to existing test examples. CIFAR-10-C [98] applies 19 different synthetic perturbations to the CIFAR-10 test set. For many of these perturbations, we observe linear trends for CIFAR-10 trained models, e.g. the `Fog` shift in Figure 4.1. However, there are several exceptions, most notably adding isotropic Gaussian noise. We give further examples of linear trends on synthetic CIFAR-10-C shifts in Appendix 4.11, and we more thoroughly discuss non-examples of linear trends in Section 4.4. In Figure 4.1, we also show that pose-estimation models trained on rendered images of YCB-Objects [39] follow a linear trend when evaluated on a images rendered with perturbed lighting and texture conditions.

Distribution shifts in the wild. We also find linear trends on two of the real-world distribution shifts from the WILDS benchmark [120]: FMoW-WILDS and iWildCam-WILDS. FMoW-WILDS is a satellite image classification task derived from Christie et al. [50] where in-distribution data is taken from regions (e.g., the Americas, Africa, Europe) across the Earth between 2002 and 2013, the out-of-distribution test-set is sampled from each region during 2016 to 2018, and models are evaluated by their accuracy on the worst-performing region. In Figure 4.1, we show models trained on FMoW-WILDS exhibit linear trends when evaluated out-of-distribution under both of these temporal and subpopulation distribution shifts.

iWildCam-WILDS is an image dataset of animal photos taken by camera traps deployed in multiple locations around the world [120, 14]. It is a multi-class classification task, where the goal is to identify the animal species (if any) within each photo. The held-out test set comprises photos taken by camera traps that are not seen in the training set, and the distribution shift arises because different camera traps vary markedly in terms of angle, lighting, and background. In Figure 4.1, we show models trained on iWildCam-WILDS also exhibit linear trends when evaluated OOD across different camera traps.

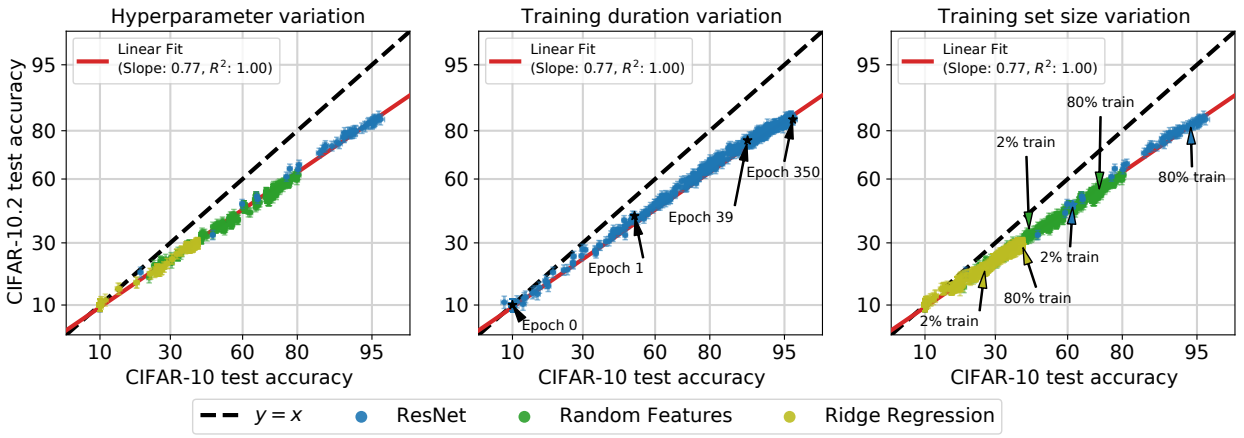


Figure 4.2: The linear trend between ID and OOD accuracy is invariant to changes in model hyperparameters, the number of training steps, and training set size. In each panel, we compare models with the linear fit from Figure 4.1. **Left:** For each model family, we vary model-size, regularization, and optimization hyperparameters. **Middle:** We evaluate each network after every epoch of training. **Right:** We train models on randomly sampled subsets of the training data, ranging from 1% to 80% of the CIFAR-10 training set size. In each setting, variation in hyperparameters, training duration, or training set size moves models along the trend line, but does not affect the linear fit.

Variations in model hyperparameters, training duration, and training set size

The linear trends we observe hold not just across different models, but also across variation in model and optimization hyperparameters, training dataset size, and training duration.

In Figure 4.2, we train and evaluate both classical and neural models on CIFAR-10 and CIFAR-10.2 while systematically varying (1) model hyperparameters, (2) training duration, and (3) training dataset size. When varying hyperparameters controlling the model size, regularization, and the optimization algorithm, the model families continue to follow the same trend line ($R^2 = 0.99$). We also find models lie on the same linear trend line *throughout training* ($R^2 = 0.99$). Finally, we observe models on trained on random subsets of CIFAR-10 lie on the same linear trend line as models trained on the full CIFAR-10 training set, despite their corresponding drop in in-distribution accuracy ($R^2 = 0.99$). In each case, hyperparameter tuning, early stopping, or changing the amount of i.i.d. training data moves models along the trend line, but does not alter the linear fit.

While we focus here on CIFAR-10 models evaluated on CIFAR-10.2, in Appendix 4.11, we conduct an identical set of experiments for CINIC-10, CIFAR-10-C `Fog`, YCB-Objects, and FMoW-WILDS. We find the same invariance to hyperparameter, dataset size, and training duration shown in Figure 4.2 also holds for these diverse collection of datasets.

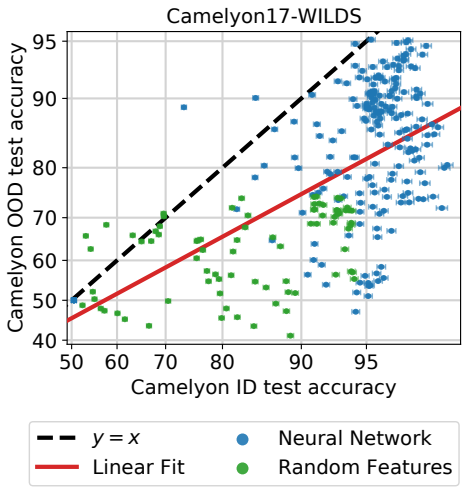


Figure 4.3: A range of neural network and random feature models trained on Camelyon17-WILDS and evaluated on the ID and OOD test sets. OOD accuracy is highly variable across the spectrum of ID accuracies, and there is no precise linear trend.

4.4 Distribution shifts with weaker correlations

We now investigate distribution shifts with a weaker correlation between ID and OOD performance than the examples presented in the previous section. We will discuss the Camelyon17-WILDS tissue classification dataset and specific image corruptions from CIFAR-10-C. Further discussion of a version of the iWildCam-WILDS wildlife classification dataset with a different in-distribution train-test split can be found in Appendix 4.12.

Camelyon17-WILDS

Camelyon17-WILDS [7, 120] is an image dataset of metastasized breast cancer tissue samples collected from different hospitals. It is a binary image classification task where each example is a tissue patch. The corresponding label is whether the patch contains any tumor tissue. The held-out OOD test set contains tissue samples from a hospital not seen in the training set. The distribution shift largely arises from differences in staining and imaging protocols across hospitals.

In Figure 4.3, we plot the results of training different ImageNet models and random features models from scratch across a variety of random seeds. There is significant variation in OOD performance. For example, the models with 95% ID accuracy have OOD accuracies that range from about 50% (random chance) to 95%. This high degree of variability holds even after averaging each model over ten independent training runs (see Appendix 4.12).

Appendix 4.12 also contains additional analyses exploring the potential sources of OOD performance variation, including ImageNet pretraining, data augmentation, and similarity

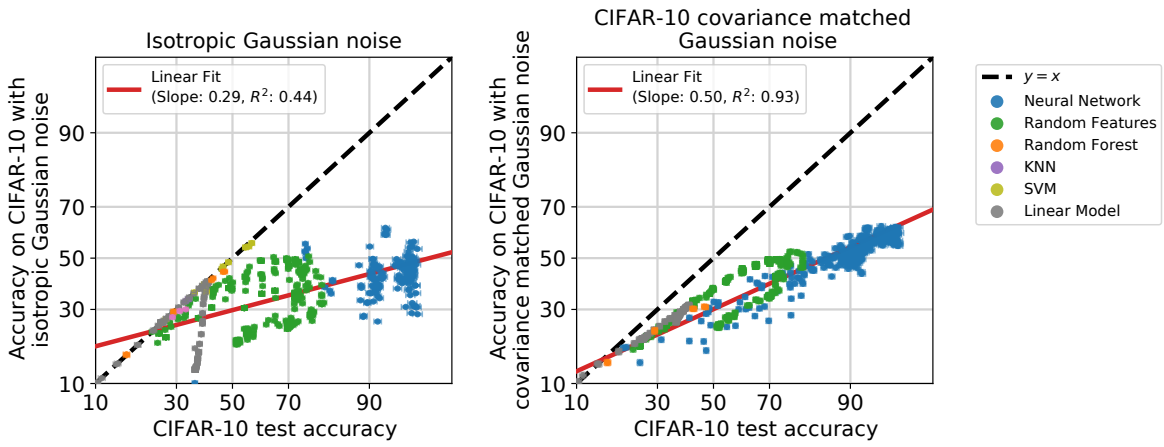


Figure 4.4: When the out-of-distribution data covariance matches the in-distribution data covariance, the linear fit is significantly better. **Left:** A collection of models trained on CIFAR-10 and evaluated in-distribution on CIFAR-10 and out-of-distribution on CIFAR-10 images corrupted with *isotropic* Gaussian noise. **Right:** The same collection of models evaluated out-of-distribution on CIFAR-10 images corrupted with Gaussian noise with the *same covariance as CIFAR-10*.

between test examples. Specifically, we observe that ImageNet pretraining does not increase the ID-OOD correlation, while strong data augmentation significantly reduces, but does not eliminate, the OOD variation. Another potential reason for the variation is the similarity between images from the same slide / hospital, as similar examples have been shown to result in analogous phenomena in natural language processing [261]. We explore this hypothesis in a synthetic CIFAR-10 setting, where we simulate increasing the similarity between examples by taking a small seed set of examples and then using data augmentations to create multiple similar versions. We find that in this CIFAR-10 setting, shrinking the effective test set size in this way increases OOD variation to a substantially greater extent than shrinking the effective training set size.

CIFAR-10-Corrupted

CIFAR-10-C [98] corrupts the CIFAR-10 test set with various image perturbations. The choice of corruption can have a significant impact on the correlation between ID and OOD accuracy. Appendix 4.12 provides plots and R^2 values for each corruption. We already showed an example of one of the more precise fits, fog corruption, in Figure 4.1 (bottom middle). Interestingly, the mathematically easy to describe corruption with Gaussian noise is one of the corruptions with worst ID-OOD correlation (see Figure 4.4 left).

We also investigate how the relationship between the ID and OOD data covariances impacts the linear trend. The theoretical model discussed in Section 4.7 predicts linear fits

occur if the data covariances between ID and OOD are the same up to a constant scaling factor. Thus, in Figure 4.4, we compare adding *isotropic* Gaussian noise to the CIFAR-10 test set versus adding Gaussian noise with the *same covariance as data examples from CIFAR-10*. We find that when the OOD covariance matches the ID covariance the linear fit is substantially better ($R^2 = 0.93$ vs. $R^2 = 0.44$). This finding is consistent with the theoretical model we propose and discuss in Section 4.7.

In Appendix 4.12, we also compare CIFAR-10-C to ImageNet-C and notice that each corruption displays a more linear trend on ImageNet-C compared to CIFAR-10-C. Investigating this discrepancy further is an interesting direction for future work.

4.5 The effect of pretrained models

In this section, we expand our scope to methods that leverage models pretrained on a third auxiliary distribution different from the ones we refer to in-distribution (ID) and out-of-distribution (OOD). Fine-tuning pretrained models on the task-specific (ID) training set is a central technique in modern machine learning [68, 188, 122, 176, 64], and zero-shot prediction (using the pretrained model directly without any task-specific training) is showing increasing promise as well [34, 182]. Therefore, it is important to understand how the use of pretrained models affects the robustness of models to OOD data, and whether fine-tuning and zero-shot inference differ in that respect.

The dependence of the pretrained model on auxiliary data makes the ID/OOD distinction more subtle. Previously, "ID" simply referred to the distribution of the training set, while OOD referred to an alternative distribution not seen in training. In this section, the training set includes the auxiliary data as well, but we still refer to the *task-specific* training set distributions as ID. This means, for example, that when fine-tuning an ImageNet model on the CIFAR-10 training set, we still refer to accuracy on the CIFAR-10 test set as ID accuracy. In other words, the "ID" distributions we refer to in this section are precisely the "ID" distributions of the previous sections (displayed on the x-axes in our scatter plots), but the presence of auxiliary training data alters the meaning of the term.

With the effect of auxiliary data on the meaning of "ID" in mind, it is reasonable to expect that ID/OOD linear trends observed when training purely on ID data will change or break down when pretrained models are used. In this section we test this hypothesis empirically and reveal a more nuanced reality: the task and the use of the pretrained model matter, and sometimes models pre-trained on seemingly broader distributions still follow the same linear trend as the models trained purely on in-distribution data. We first present our findings for fine-tuning pretrained ImageNet models and subsequently discuss results for zero-shot prediction. See Appendix 4.13 for more experimental details.

Fine-tuning pretrained models on ID data. Figure 4.5 plots OOD performance vs. ID performance for models trained from-scratch (purely on ID data) and fine-tuned models whose initialization was pretrained on ImageNet. Across the board, pretrained models attain

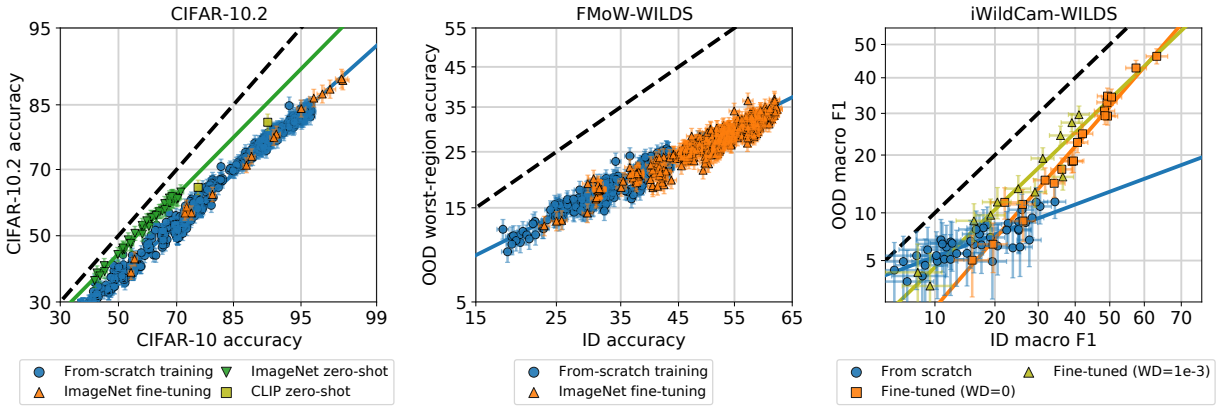


Figure 4.5: The effect of pre-training with additional data on CIFAR-10.2 (left), FMoW-WILDS (middle), and iWildCam-WILDS (right). On CIFAR-10.2 and FMoW-WILDS, fine-tuning pretrained models moves the models along the predicted ID-OOD line. However, on CIFAR-10.2, zero-shot prediction using pretrained models deviates from this line. On iWildCam-WILDS, fine-tuning pretrained models changes the ID-OOD relationship observed for models trained from scratch. Moreover, the weight decay hyperparameter affects the ID-OOD relationship in fine-tuned models.

better performance on both the ID and OOD test sets. However, fine-tuning affects ID-OOD correlations differently across tasks. In particular, for CIFAR-10 reproductions and for FMoW-WILDS, fine-tuning produces results that lie on the same ID-OOD trend as purely ID-trained models (Figure 4.5 left and center). On the other hand, a similar fine-tuning procedure yields models with a different ID-OOD relationship on iWildCam-WILDS than models trained from scratch on this dataset. Moreover, the weight decay used for fine-tuning seems to also affect the linear trend (Figure 4.5 right).

One conjecture is that the qualitatively different behavior of fine-tuning on iWildCam-WILDS is related to the fact that ImageNet is a more diverse dataset that may encode robustness-inducing invariances that are not represented in the iWildCam-WILDS ID training set. For instance, both ImageNet and iWildCam-WILDS contain high-resolution images of natural scenes, but the camera perspectives in iWildCam-WILDS may be more limited compared to ImageNet. Hence ImageNet classifiers may be more invariant to viewpoint, which may aid generalization to previously unseen camera viewpoints in the OOD test set of iWildCam-WILDS. On the other hand, the satellite images in FMoW-WILDS are all taken from an overhead viewpoint, so learning invariance to camera viewpoints from ImageNet might not be as beneficial. Investigating this and related conjectures (e.g., invariances such as lighting, object pose, and background) is an interesting direction for future work.

Zero-shot prediction on pretrained models. A common explanation for OOD performance drop is that training on the ID training set biases the model toward patterns that

are more predictive on the ID test set than on its OOD counterpart. With that explanation in mind, the fact that fine-tuned models maintain the same ID/OOD linear trend as from-scratch models is surprising: once could reasonably expect that an initialization determined independently of either ID or OOD data would produce models that are less biased toward the former. Indeed, in the extreme scenario that no fine-tuning takes place, the model should have no bias toward either distribution, and we therefore expect to see a different ID/OOD trend.

The CIFAR-10 allows us directly test this expectation directly by performing zero-shot inference on models pretrained on ImageNet: since the CIFAR-10 classes form a subset of the ImageNet classes, we simply feed (resized) CIFAR-10 images to these models, and limit the prediction to the relevant class subset. The resulting classifiers have no preference for either the ID or OOD test set because they depend on neither distribution. We plot the zero-shot prediction results in Figure 4.5 (left) and observe that, as expected, they deviate from the basic linear trend. Moreover, they form a different linear trend closer—but not identical—to $x = y$. The fact that the zero-shot linear trend is closer to $x = y$ supports the hypothesis that the performance drop partially stems from bias in ID training. However, the fact that this trend is still below $x = y$ suggests that the drop is also partially due to CIFAR-10 reproductions being harder than CIFAR-10 for current methods (interestingly, humans show similar performance on both test sets [190, 161, 210]). These finding agree with prior work [147].

As another test of zero-shot inference, we apply two publically-available CLIP models on CIFAR-10 by creating last-layer weights out of natural language descriptions of the classes [182]. As Figure 4.5 (left) shows, these models are slightly above the basic ID/OOD linear trend, but below the trend of zero-shot inference with ImageNet models.

Additional experiments. In Appendix 4.13 we describe additional experiments with pretrained models. To explore a middle ground between zero-shot prediction and full-model fine-tuning, we consider a linear probe on CLIP for both CIFAR-10 and FMoW-WILDS. For CIFAR-10, we also consider models trained on a task-relevant subset of ImageNet classes [61] and models trained in a semi-supervised fashion using unlabeled data from 80 Million Tiny Images [234, 41, 5]. Generally, we find that, compared to zero-shot prediction, these techniques deviate less from the basic linear trend. We also report results on additional OOD settings, namely CIFAR-10.1 and different region subsets for FMoW-WILDS, and reach similar conclusions.

4.6 Summary of empirical phenomena

The previous sections have presented a variety of empirical phenomena concerning the relationship between in-distribution and out-of-distribution performance. To summarize these phenomena, we now briefly highlight the key observations. These observations will also guide the development of our theoretical model of distribution shift in the next section.

The key observations are:

1. The linear trend between in-distribution and out-of-distribution performance applies to a wide range of model families and holds under variation in architecture, hyperparameters, and training duration (Section 4.3).

2. The linear trends are more precise after applying a probit or logit scaling on both axes of the scatter plotes.

3. Changing the amount of in-distribution training data does not affet the linear trend (Section 4.3). On the other hand, pre-training on a different data distribution can – but does not always – yield a different trend (Section 4.5).

4. Some distribution shifts show precise linear trends while others do so only for subsets of models or not at all (Section 4.4).

4.7 Theoretical models for linear fits

In this section we propose and analyze a simple theoretical model that distills several of the empirical phenomena from the previous sections. Our goal here is *not* to obtain a general model that encompasses complicated real distributions such as the images in CIFAR-10. Instead, our focus is on finding a simple model that is still rich enough to exhibit some of the same phenomena as real data distributions.

A simple Gaussian distribution shift setting

We consider a simple binary classification problem where the label y is distributed uniformly on $\{-1, 1\}$ both in the original distribution D and shifted distribution D'. Conditional on y, we consider D such that $\boldsymbol{x} \in \mathbb{R}^d$ is an isotropic Gaussian, i.e.,

$$\boldsymbol{x} \,|\, y \;\sim\; \mathcal{N}(\boldsymbol{\mu} \cdot y;\, \sigma^2 I_{d\times d}),$$

for mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and variance $\sigma^2 > 0$.

We model the distribution shift as a change in σ and $\boldsymbol{\mu}$. Specifically, we assume that the shifted distribution D' corresponds to shifted parameters

$$\boldsymbol{\mu}' = \alpha \cdot \boldsymbol{\mu} + \beta \cdot \boldsymbol{\Delta} \qquad \text{and} \qquad \sigma' = \gamma \cdot \sigma \tag{4.1}$$

where $\alpha, \beta, \gamma > 0$ are fixed scalars and $\boldsymbol{\Delta}$ is *uniformly distributed* on the sphere in $\mathbb{R}^d$. Note that in our setting D' is a random object determined by the draw of $\boldsymbol{\Delta}$.

Within the setup describe above, we focus on linear classifiers of the form $\boldsymbol{x} \mapsto \operatorname{sign}(\boldsymbol{\theta}^\top \boldsymbol{x})$. The following theorem states that, as long as $\boldsymbol{\theta}$ depends only on the training data and is *thereby independent of the random shift direction* $\boldsymbol{\Delta}$, the probit-transformed accuracies on D and D' have a near-linear relationship with slope α/γ. (Recall that the probit transfrom

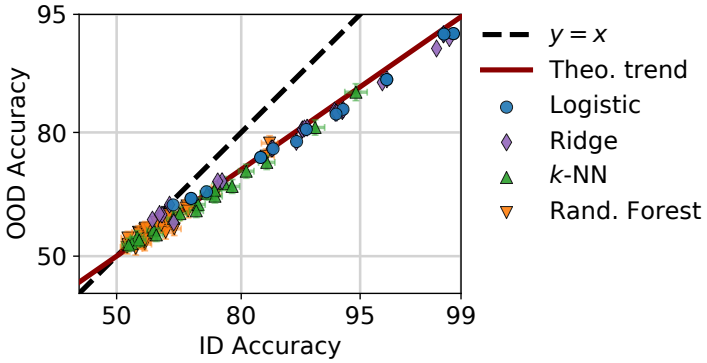


Figure 4.6: Illustration of the theoretical distribution shift model in Section 4.7 with $d = 10^5$, $\alpha = 0.7$, $\beta = 0.5$ and $\gamma = 1$ (see Appendix 4.14 for details). The accuracies for linear models (logistic and ridge regression) agree with the prediction of Theorem 6. Moreover, nonlinear models (nearest neighbors and random features) exhibit the same probit trend we prove for linear classifiers.

is the inverse of the standard Normal cdf $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \mathrm{d}t$). The deviation from linearity is of order $d^{-1/2}$ and vanishes in high dimension.

Theorem 6. *In the setting described above where $\boldsymbol{\Delta}$ is independent of $\boldsymbol{\theta}$, let $\delta \in (0, 1)$. With probability at least $1 - \delta$, we have*

$$\left| \Phi^{-1}(\mathrm{acc}_{D'}(\boldsymbol{\theta})) - \frac{\alpha}{\gamma} \Phi^{-1}(\mathrm{acc}_{D}(\boldsymbol{\theta})) \right| \;\leq\; \frac{\beta}{\gamma\sigma} \sqrt{\frac{2 \log 2/\delta}{d}} \,.$$

The theorem is a direct consequence of the concentration of measure; see proof in Appendix 4.14.

We illustrate Theorem 6 by simulating its setup and training different linear classifiers by varying the loss function and regularization. Figure 4.6 shows good agreement between the performance of linear classifiers and the theoretically-predicted linear trend. Furthermore, conventional nonlinear classifiers (nearest neighbors and random forests) also satisfy the same linear relationship, which does not directly follow from our theory. Nevertheless, if the decision boundary of the nonlinear becomes nearly linear in our setting a similar theoretical analysis might be applicable. Our simple Gaussian setup thus illustrates how linear trends can arise across a wide range of models.

Modeling departures from the linear trend

In the previous section, we identified a simple Gaussian setting that showed linear fits across a large range of models. Now we discuss small changes to the setting that break linear trends and draw parallels to the empirical observations on complex datasets presented in this chapter. In Appendix 4.14, we discuss each of these modifications in further detail.

Adversarial distribution shifts. Previously, the direction $\boldsymbol{\Delta}$ which determines the distribution shift as defined above in eq. (4.1), was chosen independent of the tested models $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k$. However, when $\boldsymbol{\Delta}$ is instead chosen by an adversary with knowledge of the tested models, the ID-OOD relationship can be highly non-linear. This is reminiscent of adversarial robustness notions where models with comparable in-distribution accuracies can have widely differing adversarial accuracies depending on the training method.

Pretraining data. Additional training data from a *different* distribution available for pretraining could contain information about the shift $\boldsymbol{\Delta}$. In this case, the pretrained models are not necessarily independent of $\boldsymbol{\Delta}$ and these models could lie above the linear fit of classifiers without pretraining. See Section 4.5 for a discussion of when such behavior arises in practice.

Shift in covariance. Previously, we assumed that $\boldsymbol{x} \mid y$ is always an isotropic Gaussian. Instead consider a setting where the original distribution is of the form $\boldsymbol{x}|y \sim \mathcal{N}(\boldsymbol{\mu}y; \Sigma)$ where Σ is not scalar (i.e., has distinct eigenvalues). Then, the linear trend breaks down even when the distribution shift is simple additive white Gaussian noise corresponding to $\boldsymbol{x}|y \sim \mathcal{N}(\boldsymbol{\mu}y; \Sigma + (\sigma')^2 I_{d\times d})$. For example, ridge regularization turns out to be an effective robustness intervention in this setting. However, if the shifted distribution is of the form $\boldsymbol{x}|y \sim \mathcal{N}(\boldsymbol{\mu}y; \gamma\Sigma)$ for some scalar $\gamma > 0$, it is straightforward to see that a linear trend holds.

These theoretical observations suggest that a covariance change in ID/OOD the distribution shift could be a possible explanation for some departures from the linear trends such as additive Gaussian noise corruptions in CIFAR-10-C. To test this hypothesis, we created a new distribution shift by corrupting CIFAR-10 with noise sampled from the same covariance as the original CIFAR-10 distribution. As discussed in Section 4.4, we find that the correlation between ID and OOD accuracy is substantially higher with the covariance-matched noise than with isotropic Gaussian noise with similar magnitude.

While the theoretical setting we study in this work is much simpler than real-world distributions, the analysis sheds some light on when to expect linear trends and what leads to departures. Ideally, a theory would precisely explain what differentiates CIFAR-10.2, CINIC-10, and the CIFAR-10-C-Fog shift (see Figure 4.1) where we see linear trends from simply adding Gaussian noise to the images as in CIFAR-10-C-Gaussian where we do not observe linear trends. A possible direction may be to characterize shifts by their generation process, and we leave this to future work.

4.8 Related work

Due to the large body of research on distribution shifts, domain adaptation, and reliable machine learning, we only summarize the most directly related work here. Appendix 4.15 contains a more detailed discussion of related work.

Domain generalization theory. Prior work has theoretically characterized the performance of classifiers under distribution shift. Ben-David et al. [17] provided the first VC-dimension-based generalization bound. They bound the difference between a classifier's error

on the source distribution (D) and target distribution (D') via a classifier-induced divergence measure. Mansour, Mohri, and Rostamizadeh [155] extended this work to more general loss functions and provided sharper generalization bounds via Rademacher complexity. These results have been generalized to include multiple sources [25, 105, 154]. See the survey of Redko et al. [191] and the references therein for further discussion of these results. The philosophy underlying these works is that robust models should aim to minimize the induced divergence measure and thus guarantee similar OOD and ID performance.

The linear trends we observe in this chapter are not captured by such analyses. As illustrated in Figure 1 (top-left), the bounds described above can only state that OOD performance is highly predictable from ID performance if they are equal (i.e., when the gray region is tight around the $x = y$ line). In contrast, we observe that OOD performance is *both* highly predictable from ID performance and significantly different from it. Our Gaussian model in Section 4.7 demonstrates how the linear trend phenomenon can come about in a simple setting. However, unlike the above-mentioned domain generalization bounds, it is limited to particular distributions and the hypothesis class of linear classifiers.

Mania and Sra [152] proposed a condition that implies an approximately linear trend between ID and OOD accuracy, and empirically checked their condition in dataset reproduction settings. The condition is related to model similarity, and requires the probability of certain multiple-model error events to not change much under distribution shift. An interesting question for future work is whether their condition can shed light on what distribution shifts show linear trends, and what axes transformations lead to the most precise trends.

Empirical observations of linear trends. Precise linear trends between in-distribution and out-of-distribution generalization were first discovered in the context of dataset reproduction experiments. Recht et al. [189, 190], Yadav and Bottou [248], and Miller et al. [161] constructed new test sets for CIFAR-10 [124], ImageNet [62, 200], MNIST [131], and SQuAD [187] and found linear trends similar to those in Figure 1.

However, these studies were limited in their scope, as they only focused on dataset reproductions. Taori et al. [228] later showed that linear trends still occur for ImageNet models on datasets like ObjectNet, Vid-Robust, and YTBB-Robust [9, 209]. On ImageNet, Shankar et al. [211] showed that linear trends also occur between the original top-k accuracy metrics and a multi-label accuracy metric based on a new set of multi-label annotations. All of these experiments, however, were limited to ImageNet or ImageNet-like tasks. We significantly broaden the scope of the linear trend phenomenon by including a range of additional distribution shifts such as CINIC-10, STL-10, FMoW-WILDS, and iWildCam-WILDS, as well as identifying negative examples like Camelyon17-WILDS and some CIFAR-10-C shifts. In addition, we also include a pose estimation task with YCB-Objects. The results show that linear trends not only occur in academic benchmarks but also in distribution shifts coming from applications "in the wild." Furthermore, we show that linear trends hold across different learning approaches, training durations, and hyperparameters.

Kornblith, Shlens, and Le [122] study linear fits in the context of transfer learning and train or fine-tune models on the distribution corresponding to the y-axis in our setting. On a variety of image classification tasks, they show that a model's ImageNet test accuracy

linearly correlates with the model's accuracy on the new task after fine-tuning. The similarity between their results and ours suggest that they may both be part of a broader phenomenon of predictable generalization in machine learning.

In concurrent work, Andreassen et al. [3] study the impact of fine-tuning on the effective robustness of pre-trained models, i.e., how much a pre-trained model is above the linear trend given by models trained only on in-distribution data [228]. At a high level, this investigation is similar to Section 4.5 in this chapter, but the datasets are complementary: Andreassen et al. [3] focus on distribution shifts in the context of CIFAR-10 and ImageNet while we also study the effect of pre-training on the three distribution shifts from WILDS (FMoW-WILDS, iWildCam-WILDS, and Camelyon17-WILDS). Andreassen et al. [3] measure how the effective robustness of a model evolves during fine-tuning and find that models gain accuracy but lose effective robustness over the course of fine-tuning. In addition, Andreassen et al. [3] investigate how data diversity in the pre-training data along with other factors like model size impact the effective robustness achieved by fine-tuning.

4.9 Discussion

Initial research on dataset reproductions found that many neural networks follow a linear trend in scatter plots relating in-distribution to out-of-distribution performance. Our work and concurrent work on object detection, natural language processing, and magnetic resonance imaging [36, 144, 60] show that such linear trends are not a peculiarity of dataset reproductions but occur for many types of models and distribution shifts. The striking regularity of these trends raises the possibility that for certain classes of distribution shifts and models, out-of-distribution performance is solely a function of in-distribution performance.

While our work and related work give many examples of model types and distribution shifts with such a universal trend, our experiments have also demonstrated datasets where linear trends do *not* occur. This naturally raises the question for what model types and distribution shifts out-of-distribution performance is a function of in-distribution performance. As a starting point for future work in this direction, we now formalize the precise relationship between in-distribution and out-of-distribution performance as *correlation property*.

Definition 7 (Correlation property)**.** *A pair of distributions D, D' and a family of models $\mathcal{M}$ have the α-approximate correlation property under loss function ℓ (e.g., accuracy) and monotone transform function $\gamma : \mathbb{R} \to \mathbb{R}$ (e.g., a linear function in probit domain) if for all models $f \in \mathcal{M}$ we have*

$$\left| \gamma\left(\mathop{\mathbb{E}}_{x,y\sim D} [\ell(f(x), y)] \right) \; - \; \mathop{\mathbb{E}}_{x,y\sim D'} [\ell(f(x), y)] \right| \; \leq \; \alpha \; .$$

With the correlation property in place, we can now state a candidate hypothesis for specific distribution shifts such as the shift from CIFAR-10 to CIFAR-10.1.

Conjecture 1. *The distribution shift from CIFAR-10 to CIFAR-10.1 (or ImageNet to ImageNet-V2, or FMoW-WILDS, etc.) has the 1%-approximate correlation property with loss function accuracy and transform*

$$\gamma(l) \;=\; \Phi\left(\mathit{slope} \cdot \Phi^{-1}(l) + \mathit{offset}\right)$$

for models that are trained with empirical risk minimization (ERM) on the respective training distribution.

The restriction to models trained on in-distribution data is important since for instance ImageNet models applied to CIFAR-10 in a zero-shot manner follow a different trend (recall Section 4.5). Our experiments with a wide range of ERM models give evidence for these conjectures, but are not ultimate proof. An interesting direction for future work is understanding which distributions shifts have the correlation property for ERM models.

Question 1. *What pairs of distributions D, D' have the correlation property for models trained via empirical risk minimization only on D?*

Beyond exploring the data distribution question, another dimension of the correlation property is what models it applies to. There is evidence that statements similar to Conjecture 1 hold for a wider range of models than ERM. For instance, nearest neighbor models follow the same trend as ERM models on several distribution shifts studied in this chapter (see Figure 4.1). Moreover, earlier work found that a wide range of robustness interventions (e.g., adversarial training, data augmentation, filtering layers, distributionally robust optimization) do not improve over ERM baselines on the ImageNet-V2, ObjectNet, and FMoW-WILDS distribution shifts [228, 120, 89]. On the other hand, it is also easy to construct models that do *not* follow the same trend as the ERM baseline. Interpolating between a CIFAR-10 model on the linear trend and a random classifier yields models above the linear trend.[1] This leads to the following question:

Question 2. *For the CIFAR-10 → CIFAR-10.1 distribution shift (or ImageNet to ImageNet-V2, or FMoW-WILDS, etc.), what models satisfy the correlation property?*

Beyond interpolating with a random classifier, we are currently not aware of models that violate the correlation property on the aforementioned distribution shifts by a substantial amount. Interpolating with a random classifier is a peculiar intervention since it decreases *both* in-distribution and out-of-distribution performance. Hence a stronger version of Conjecture 1 that applies to a wider range or even all "useful" models trained only on in-distribution data is plausible. However, we currently do not have a satisfying way to make this stronger conjecture precise. We hope that future work further investigates Questions 1 and 2 (and their combination) both empirically and theoretically to shed light on the relationship between in-distribution and out-of-distribution generalization.

[1] To interpolate between the two classifiers, draw a sample from a Bernoulli with success probability p. If the sample is 0, classify with the original classifier. Otherwise classify with the random classifier. Varying p in $[0, 1]$ interpolates between the two classifiers.

Possible implications

If the correlation property holds for relevant distribution shifts and models, it can be a valuable guide for building reliable machine learning systems. An important point here is that – at least empiricially – the correlation property often holds not only for a single pair of distributions, but for an entire range of distribution shifts (e.g., from CIFAR-10 to CIFAR-10.1, CIFAR-10.2, CINIC-10, and STL-10 or for temporal and spatial distribution shifts in FMoW-WILDS). So when a practitioner encounters a linear trend between in-distribution and out-of-distribution performance, it is reasonable to expect that similar distribution shifts will also exhibit a linear trend.

We now briefly describe three implications that arise when the correlation property holds for a range of distribution shifts. We note that these implications are conditional and more research is needed to understand their extent.

- **Model selection.** Practitioners are often faced with the challenge of selecting a model that performs well not only on a specific test set but also on future unseen data that may come from different distributions. If the shifted distributions have the correlation property with respect to the training distribution, selecting the best model under these distribution shifts reduces to selecting the best model on the in-distribution test set.

- **Baseline for measuring out-of-distribution robustness.** A central goal of research on reliable machine learning is to develop models that perform well on out-of-distribution data. There are two natural ways to quantify this goal: (i) performance on out-of-distribution data, and (ii) the gap between in-distribution and out-of-distribution performance.

  The correlation property for empirical risk minimization implies that optimizing for in-distribution performance also provides corresponding gains in out-of-distribution performance. Hence existing work on improving in-distribution performance already improves the robustness of a model according to criterion (i) without explicitly targeting robustness. So if a proposed training technique claims to improve the robustness of a model as a quantity distinct from in-distribution performance, the proposed technique should not only improve out-of-distribution performance, but also reduce the gap between in-distribution and out-of-distribution performance – criterion (ii) – beyond what current methods optimizing for in-distribution performance achieve. Graphically in terms of our scatter plots, the proposed technique promoting robustness should produce a model that lies *above* the linear trend given by empirical risk minimization.[2]

  To better compare new training techniques to prior work in terms of robustness, we recommend that papers illustrate the effect of their technique with a scatter plot of

[2]Improving the out-of-distribution performance of a model by improving its in-distribution performance is also clearly a valid way to improve robustness according to criterion (i). But the proposed technique should then be compared to existing methods for improving in-distribution performance (architecture variations, training schedules, etc.) and ideally improve over the out-of-distribution performance achieved by state-of-the-art methods for in-distribution performance.

relevant models (e.g., the evaluations in Taori et al. [228], this chapter, and Section 3.3 of Radford et al. [182]). In addition, papers should report *both* in-distribution and out-of-distribution performance of their technique so that the effect on both quantities is clear. We also refer the reader to Taori et al. [228], who formalize the concept of "robustness beyond a baseline" as "effective robustness".

- **Guide for algorithmic interventions to improve robustness.** If we can characterize for what set of training approaches the correlation property holds, research aiming to decrease the gap between in-distribution and out-of-distribution performance can focus on other approaches. For instance, our experiments suggest that architecture variations in neural network may not affect the gap between in-distribution and out-of-distribution performance, but better pre-training datasets can at least sometimes reduce this gap.

Frequently asked questions

In conversations with colleagues and reviewers, certain questions about our work appeared repeatedly. To clarify our perspective on these issues, we answer the three most common questions below:

Q: Do all distribution shifts have linear trends?

No. Section 4.4 gives concrete examples of distribution shifts that do not follow a linear trend. Before our work, it was already clear that adversarial distribution shifts such as ℓ_∞-robustness do not follow a linear trend because models trained with ERM usually have little to no robustness to adversarial examples while multiple approaches give non-trivial robustness, e.g., [150, 55, 184, 244]. Since multiple non-adversarial and natural (i.e., not synthetically constructed) distribution shifts *do* show linear trends for a wide range of models, the main question is *what* distribution shifts have linear trends.

Q: Should we only work on improving in-distribution performance?

No. As mentioned above, not all distribution shifts have linear trends. Moreover, more work is needed to understand whether new robustness interventions can improve over the linear trends observed for empirical risk minimization and existing robustness interventions [228, 120]. In addition, Section 4.13 suggests pre-training as a promising direction for improving out-of-distribution performance, as also shown by the recent CLIP model [182].

Q: Is it is possible to construct models that violate the linear trend?

Yes, when the linear trend is on a non-linear transformation of the accuracy such as the probit transform we use in this chapter. This is due to the fact that we can always construct a family of models with a *linear* ID/OOD performance relationship (without the probit transform) by randomly switching between two base models.

Concretely, let f be a model with non-trivial performance and consider the following *interpolations* between f and a trivial random classifier: given input x, output $f(x)$ with probability p and output a random class label with probability $1-p$. Let C be the number of classes and let $\text{acc}_D(f)$ and $\text{acc}_{D'}(f)$ be the in-distribution and out-of-distribution accuracies of f, respectively. Then, as we vary p from 0 to 1, the in- and out-of-distributions accuracies of the interpolating model trace a line from $(1/C, 1/C)$ to $(\text{acc}_D(f), \text{acc}_{D'}(f))$. Therefore, if we apply a non-linear transformation to these accuracies (such as a probit transform), the in- and out-of-distribution performance of these models no longer follows a linear trend. Characterizing for which models the linear trend (approximately) holds is an important direction for future work.

4.10 Appendix: Omitted details about experimental testbed

A rigorous empirical investgation of the correlation between in-distribution and out-of-distribution performance requires a broad set of experiments. To measure the behavior of many models on a variety of datasets, we utilized three different experimental "testbeds." A testbed consists of a collection of one or more "dataset universes" and a compatible set of models that can be trained and evaluated on these "universes." Each dataset universe itself consists of a training set (e.g. CIFAR-10 train), an in-distribution test-set (CIFAR-10 test), and a collection of out-of-distribution test-sets (e.g. CIFAR-10.2, CIFAR-10-C, etc). Within a universe, models trained on one dataset can be tested on all other datasets, with each test set representing a different distribution. The three testbeds we use are:

1. A new custom-built test for experiments with CIFAR-10 and WILDS (FMoW-WILDS, Camelyon17-WILDS, and iWildCam-WILDS)
2. An ImageNet testbed based on Taori et al. [228], and
3. A testbed for pose estimation in the context of the YCB-Objects dataset [39].

In the rest of this section, we first detail the custom-built CIFAR-10 and WILDS testbed since it forms the basis for most experiments in this chapter. We then describe our modifications to the ImageNet testbed of Taori et al. [228] in Section 4.10, and finally we describe our testbed for YCB-Objects in Section 4.10.

CIFAR-10 and WILDS testbed

We now describe the datasets in our main testbed and summarize the models it contains. Our main testbed contains four distinct "universes." Each universe consists of at least three datasets that we use for training and testing models both in-distribution and out-of-distribution.

The four universes are CIFAR-10, FMoW-WILDS, Camelyon17-WILDS, and iWildCam-WILDS, which we now describe in more detail. The latter three datasets are taken from the WILDS benchmark [120], and we use the train/test splits and evaluation procedures therein.

CIFAR-10 and related datasets

The CIFAR-10 universe comprises 32×32 pixel color images used in an image classification task. The ten classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The CIFAR-10 universe contains the following datasets:

- **CIFAR-10** is the main dataset in the CIFAR-10 universe and was introduced by Krizhevsky [124]. CIFAR-10 is derived from the larger Tiny Images dataset [234]. Since its introduction, CIFAR-10 has become one of the most widely used image classification benchmarks.

- **CIFAR-10.1** is a reproduction of the CIFAR-10 dataset. Recht et al. [190] closely followed the dataset creation process of CIFAR-10 and assembled a new dataset also using Tiny Images as a source. CIFAR-10.1 contains only about 2,000 images and is therefore usually used only as a test set. The distribution shift from CIFAR-10 to CIFAR-10.1 poses an interesting challenge since many parameters of the data generation process are held constant but a standard ResNet model still sees an 8 to 9 percentage points accuracy drop.

- **CIFAR-10.2** is a second reproduction of the CIFAR-10 dataset. Lu et al. [147] again closely followed the dataset creation process of CIFAR-10 to assemble a new dataset from Tiny Images, this time with different annotators compared to CIFAR-10.1. CIFAR-10.2 contains 12,000 images with a suggested split into 10,000 training images and 2,000 test images. We conduct all of our experiments using the 2,000 image test set. Similar to CIFAR-10.1, CIFAR-10.2 is a distribution shift arising from changes in the filtering process conducted by the human annotators.

- **CINIC-10** [61] is a dataset in CIFAR-10 format that supplements CIFAR-10 with additional images from the full ImageNet dataset (not only the 2012 competition set). In total, CINIC-10 contains 270,000 images. Here, we limit CINIC-10 to the images coming from ImageNet in order to keep the distribution more clearly separate from CIFAR-10. The resulting test set has size 70,000. CINIC-10 represents a distribution shift because the source of the images changed from Tiny Images to ImageNet.

- **STL-10** [52] is another CIFAR-10-inspired dataset derived from ImageNet. Since the focus of STL-10 is unsupervised learning, the dataset contains 100,000 unlabeled and 13,000 labeled images. We only use the labeled subset because we are mainly interested in STL-10 as a test set with distribution shift (as in CINIC-10, the data source changed from Tiny Images to ImageNet). The class structure of STL-10 is slightly different from the CIFAR-10 classes: instead of the frog class, STL-10 contains a monkey class.

When experimenting with STL-10, we therefore limit the dataset to the remaining nine classes. This yields an overall test set size of 11,700.

- **CIFAR-10-C** contains a range of synthetic distribution shifts derived from CIFAR-10. Hendrycks and Dietterich [98] created CIFAR-10-C by applying perturbations such as Gaussian noise, motion blur, or synthetic weather patterns (fog, snow, etc.) to the CIFAR-10 test set. In total, CIFAR-10-C contains 19 different perturbations, each with five different severity levels.

FMoW-WILDS

In FMoW-WILDS, which is adapted from the Functional Map of the World dataset [50], the task is to classify land or building use from satellite images taken in different geographical regions (Africa, Americas, Oceania, Asia, and Europe) and in different years. Specifically, the input is an RGB satellite image and the label is one of 62 different land or building use categories (e.g., 'shopping mall' or 'road bridge').

The training set comprises 76,863 images taken around the world between 2002 and 2013. The in-distribution test set comprises 11,327 images from the same distribution, i.e., also taken around the world between 2002 and 2013, and we evaluate models by their average accuracy. The out-of-distribution validation set comprises 19,915 images taken around the word between 2013 and 2016, and the out-of-distribution test set consists of 22,108 images taken around the world between 2016 and 2018.

We evaluate models out-of-distribution by either their average accuracy or their worst accuracy over all five geographical regions. When evaluating models using their worst-region accuracy, the out-of-distribution test set reflects both a distribution shift across time (from 2002–2013 to 2016–2018) and across regions (from images that are distributed across the world to images that are only from a given region). In our experiments, the worst-performing region is generally Africa, which has the second smallest number of training examples, ahead of Oceania.

Camelyon17-WILDS

In Camelyon17-WILDS, which is a patch-based variant of the CAMELYON17 dataset [7], the task is to classify whether a given patch of tissue contains any tumor tissue. Specifically, the input is a 96 $\times$ 96 patch of tissue extracted from a whole-slide image (WSI) of a breast cancer metastasis in a lymph node section, and the label is whether any pixel in the central 32 $\times$ 32 region of the patch has been labeled as part of a tumor in the ground-truth pathologist annotations.

The training set comprises 302,436 patches taken from 30 WSIs across 3 hospitals (10 WSIs per hospital). The in-distribution test set comprises 33,560 patches taken from the same set of 30 WSIs; this corresponds to the "in-distribution validation set" in the WILDS benchmark [120]. The out-of-distribution test set comprises 85,054 patches taken from 10 WSIs from a different hospital. All of the above sets are class-balanced. We evaluate models

by their average accuracy; performance on the out-of-distribution test set reflects a model's ability to generalize to different hospitals from the ones it was trained on.

iWildCam-WILDS

In iWildCam-WILDS, which is adapted from the iWildCam 2020 Competition Dataset [14], the task is to classify which animal species (if any) is present in a camera trap photo. Specifically, the input is a (resized) 448×448 pixel color image from a camera trap, and the label is one of 182 animal species (including "no animal").

The training set comprises 129,809 images taken by 243 camera traps; the in-distribution test set comprises 8,154 images taken by those same 243 camera traps; and the out-of-distribution training set comprises 42,791 images taken by 48 different camera traps. As images taken by different camera traps can vary greatly in terms of camera angle, illumination, background, and animal distribution, the performance on the out-of-distribution test set reflects a model's ability to generalize to different camera traps from the ones it was trained on.

While we study the current version of iWildCam-WILDS unless noted otherwise, we also study an earlier version of iWildCam-WILDS with a different in-distribution train-test split in 4.12.

Evaluation metric and confidence interval calculation. Following Koh et al. [120], we evaluate models by their macro F1 score, as this better captures model performance on rare species. Macro F1 is the average of the per class F1 scores for all classes appearing in the test data. We obtain confidence interval for this metric using the following heuristic. Suppose class i has empirical F1 score f_i and n_i examples in the test set. As an approximate confidence interval for the F1 score of this class, we consider $[f_i - \delta_i, f_i + \delta_i]$ where δ_i is such that $[0.5 - \delta_i, 0.5 + \delta_i]$ is a 95% Clopper-Pearson confidence interval for a Bernoulli success probability given $n_i/2$ positive observation out of n_i total observations. The size of this confidence interval is guaranteed to be larger than the size of the confidence intervals for both recall and precision for this class. Since the F1 score is the harmonic mean of recall and precision, the interval should provide adequate coverage for the F1 score as well. Finally, we combine the per class intervals to obtain a macro F1 confidence interval of the form $[\bar{f} - C^{-1/2}\bar{\delta}, \bar{f} + C^{-1/2}\bar{\delta}]$, where C is the number of classes in the test data and $\bar{f}, \bar{\delta}$ are the averages of $f_1, \ldots, f_C$ and $\delta_1, \ldots, \delta_C$, respectively. This expression makes the approximation that individual F1 estimates are independent, which is not entirely accurate because the per-class precision estimates rely on overlapping samples.

As Figures 4.1 and 4.11 show, the confidence intervals computed with the above heuristic are fairly large. This may be in part due to a somewhat pessimistic approximation of the individual F1 confidence intervals, but it also reflects the fact that many of the iWildCam-WILDS classes are very rare (with 10 or fewer examples in the test set), and we simply do not have a good estimate for how models perform on them. The high level of class rarity was

also the reason we chose not to use a bootstrap confidence interval: re-sampling the test sets with replacement leads to entire classes being dropped and biases the macro F1 estimates.

Label noise reduction. The iWildCam-WILDS labels contain errors that stem primarily due to the fact they are derived from video-level annotation indicating whether a motion-activation event contains a particular animal. As consequence, many video frames that are in fact empty (showing no animal) are mislabeled as containing an animal that appeared in a temporally adjacent frame. To reduce this noise, we used auxiliary the auxiliary MegaDetector data provided as part of iWildCam 2020 Competition Dataset. More specifically, we performed our evaluation only on frames that were either labeled as empty or contained a MegaDetector detection with confidence at least 0.95.[3] This filtering step provided a modest improvement to strength of the observed correlations (with R^2 increasing by one or two points).

Models for CIFAR-10 and WILDS Experiments

To probe how widely the linear trend phenomena apply, we integrated a large number of classification models into our testbed. At a high level, we divide these models into two types: deep neural networks (predominantly convolutional neural networks) and classical approaches. Due to the wide range of neural network architectures and training approaches emerging over the past decade, we further subdivide the neural network models based on their training set.

Convolutional neural networks for CIFAR-10. We integrated the following model architectures into our testbed. Unless noted otherwise, we used the implementations from `https://github.com/kuangliu/pytorch-cifar`. The models span a range of manually designed architectures and the results of automated architecture searches. We refer the reader to the respective references for details about the individual architectures.

- **DenseNet**, with depths 121 and 169 [108].
- **Dual Path Networks (DPN)**, with depths 26 and 92 [47].
- **EfficientNet**, specifically the B0 variant [227].
- **GoogLeNet**, a member of the Inception family [222].
- **MobileNet**, both the original and the MobileNetV2 variant [202].
- **MyrtleNet**, which are optimized for particularly fast training times. The code for these networks is from `https://github.com/davidcpage/cifar10-fast`.

[3] We still performed the model *training* using precisely the same data, splits and labels as Koh et al. [120]; the filtering step was done at the evaluation stage only.

- **PNASNet**, both A and B variants [142].
- **RegNet**, configurations X_200, X_400, and Y_400 [183].
- **ResNet**, varying the number of layers (18, 34, 50, and 101), and including the PreAct variant for each depth [96, 97].
- **ResNeXT** models with various widths and depths (2x64d, 32x4d, 4x64d) [246].
- **Squeeze-and-Excitation Networks** with 18 layers [107].
- **ShuffleNet**, specifically the G2, G3, and V2 variants, with network scale factors 0.5, 1, 1.5, and 4 for the ShuffleNetV2 architecture [258, 148].
- **VGG** with 11, 13, 16, and 19 layers [213].

Convolutional neural networks pre-trained on ImageNet. We explored the use of models pre-trained on ImageNet both in the CIFAR-10-universe and in the WILDS datasets. In some experiments, we also trained the following architectures from scratch to quantify the effect of pre-training in detail (see Section 4.5). The code for the following models is from `https://github.com/creafz/pytorch-cnn-finetune`.

- **AlexNet** [125].
- **DenseNet** with 121, 161, 169, and 201 layers [108]
- **Dual Path Networks (DPN)**, variants 68, 68b, and 92 [47].
- **GoogLeNet**, a member of the Inception family [222].
- **MobileNetV2** [202].
- **Neural Architecture Search Networks (NASNets)**, specifically NASNet-A-Large and PNASNet-5-Large [263, 142]
- **ResNet** with 18, 34, 50, 101, and 152 layers [96, 97].
- **ResNeXT**, configurations 50_32x4d and 101_32x4d [246].
- **Squeeze-and-Excitation Networks**, specifically se_resnext50_32x4d and se_resnext101_32x4d [107]
- **ShuffleNetV2**, scale factors 0.5 and 1 [258, 148].
- **SqueezeNet**, version 1.0 and 1.1 [109].
- **VGG** with 11, 13, and 16 layers, including variants with batch normalization for 13 and 16 layers [213].

Convolutional neural networks only trained on ImageNet. For the zero-shot generalization experiments in Section 4.5, we also utilized a set of models trained on ImageNet without any further fine-tuning to CIFAR-10. As above, the models are from `https://github.com/creafz/pytorch-cnn-finetune`.

- **AlexNet** [125]
- **DenseNet** with 121, 161, 169, and 201 layers [108].
- **Dual Path Networks (DPN)**, variants 68, 68b, 92, 98, 107, and 131 [47].
- **Inception** models: GoogleNet, InceptionV3, and InceptionResNetV2 [222, 225, 223].
- **MobileNetV2** [202].
- **PolyNet** [259].
- **ResNet** with 18, 34, 50, 101, and 152 layers [96, 97].
- **Squeeze-and-Excitation Networks** specifically senet154, se_resnet50, se_resnet101, se_resnet152, se_resnext50_32x4d, and se_resnext101_32x4d [107].
- **ShuffleNetV2**, scale factors 0.5 and 1 [258, 148].
- **SqueezeNet**, version 1.0 and 1.1 [109].
- **ResNeXT**, configurations 50_32x4d, 101_32x4d, 101_32x8d, and 101_64x4d [246].
- **VGG** with 11, 13, 16, and 19 layers, all with and without batch normalization [213].
- **Xception** [48].

Further models trained on extra data. To measure the effect of extra training data more broadly than only relying on ImageNet for pre-training, we also included the following three models utilizing data from different sources:

- **CLIP**: We evaluate the two publicly released CLIP models [182]. These models were trained with 400 million image-caption pairs scraped from the web. We evaluate the two ResNet50 and VisionTransformer variants released by the CLIP team. CLIP models are particularly interesting since they can be evaluated zero-shot: image classification labels can be turned into textual prompts so that the model can be evaluated on downstream tasks without needing to look at the training data.
- **Self-training on 80 Million Tiny Images:** Carmon et al. [41] introduced robust self-training (RST) and showed that unlabeled data can improve adversarial robustness. In the context of their work, they also trained baseline CIFAR-10 models that used data from 80 Million Tiny Images [234] in addition to the standard CIFAR-10 training set.

This baseline model is an interesting addition to our testbed since the extra training data from a potentially more diverse source may move the model away from the linear trend given by models only trained on CIFAR-10.

- **Out-distribution aware self-training (ODST):** Augustin and Hein [5] develop an iterative self-training approach to leverage unlabeled data when some of the unlabeled data is not relevant to the classification task of interest. They also instantiate their approach on CIFAR-10, using 80 Million Tiny Images as an unlabeled data source. As before, the ODST models are relevant for our experiments because they use extra training data beyond the standard CIFAR-10 training set.

Classical methods. In addition to the neural network methods discussed previously, we also integrated several classical, non-neural network methods into our testbed. Unless noted otherwise, we used the implementations from scikit-learn [172]. Each of these methods works directly on the image pixels, which are each scaled to have zero-mean and unit variance on the training set. We included the following methods into our testbed:

- Random features [53], using the implementation from `github.com/modestyachts/nondeep`.
- AdaBoost from Hastie et al. [94], using an scikit-learn decision tree classifier to build the boosted ensemble.
- Ridge regression classifiers with varying ℓ_2 regularization parameter
- Support vector machines with linear, gaussian, and polynomial kernels and varying regularization penalty term.
- Logistic regression with varying regularization parameter and using both ℓ_1 and ℓ_2 regularization.
- Quadratic discriminant analysis.
- Random forests [32] with varying maximum tree depth, number of trees in the forest, and using both entropy and gini impurity as the splitting criterion.
- Nearest neighbor with varying number of k nearest-neighbors and using ℓ_2 distance between points.

Summary statistics

The following two tables give a brief overview of the number of experiments we ran with our testbed. Table 4.2 shows how many distinct models we trained for each of our training sets (a total of about 3,000). Each of these models was then evaluated on a range of test sets to generate the scatter plots in this chapter.

Dataset	Number of trained models
CIFAR-10	1,895
iWildCam-WILDS	197
FMoW-WILDS	592
Camelyon17-WILDS	461

Table 4.2: Number of trained models (of all types) by training set. The model counts include only fully trained models, not intermediate checkpoints.

Dataset	Number of model evaluations
CIFAR-10	6,814
CIFAR-10.1	5,315
CIFAR-10.2	11,212
CIFAR-10-C	39,677
CINIC-10	4,259
STL-10	507
iWildCam-WILDS	15,147
FMoW-WILDS	12,127
Camelyon17-WILDS	7,056

Table 4.3: Number of model evaluations by test set type. Some of the rows, e.g., CIFAR-10-C, correspond to multiple individual test sets. We count evaluations of a model and its training checkpoints separately here.

Table 4.3 shows the total number of evaluations for each family of datasets. Besides being tested on multiple datasets, one trained model can also have led to several evaluations since we sometimes evaluated all training checkpoints of a model on multiple datasets as well to study whether the linear trends are reliable when varying training duration (see Section 4.3). This lead to an overall total of about 100,000 model evaluations, each of which corresponds to one point in a scatter plot in this chapter.

ImageNet Testbed

Datasets

We include all of the natural distribution shifts from the testbed of Taori et al. [228], excluding the consistency shifts since those are somewhat adversarial in nature.

ImageNet-V2. ImageNet-V2 is a reproduction of the ImageNet test set collected by Recht et al. [190].

ObjectNet. ObjectNet is a test set of objects in a variety of scenes, poses, and lighting conditions with 113 classes that overlap with ImageNet [9].

ImageNet-Vid-Anchors and YTBB-Anchors. These are two datasets introduced by Shankar et al. [209] that measure accuracy on frames taken from video clips. They contain 30 and 24 super-classes of the ImageNet class hierarchy, respectively. For evaluation, we measure accuracy using the pm-0 metric as defined in Shankar et al. [209], which measures accuracy over the anchor frames of the video clips.

ImageNet-A. ImageNet-A [99] is an adversarially collected dataset, constructed by downloading labeled images from a variety of online sources and then selecting the subset that was misclassified by a ResNet-50 model.

ImageNet Testbed Models

We include all of the existing models in the testbed from Taori et al. [228], and add a few others:

1. **CLIP:** We add the two CLIP models released by Radford et al. [182] and evaluate them zero-shot using the publicly released textual prompts.
2. **Self-supervised models:** We add models trained using a few different self-supervised methods: SimSiam [46], SimCLRv2 [45], and SwAV [42]. For SimSiam and SwAV, we use the ResNet50 variants pretrained on ImageNet without labels and then final-layer finetuned on ImageNet. For SimCLRv2, we use a ResNet50 and a ResNet152 variant, and for each use a model final-layer finetuned and whole-network finetuned on ImageNet.
3. **Classical models:** We add four low-accuracy classical models: random features [53], random forests [32], one-nearest neighbors, and a linear model trained with least squares. The random forests model, the linear model, and the one-nearest neighbors model were trained directly on pixels of images downsampled to 32x32.

4. **Low accuracy ConvNets:** We add a multitude of low-accuracy ResNet models trained for various epochs and on various subsets of the training set.

YCB-Objects Testbed

We describe the 6D pose estimation task, our synthetic dataset, and the models in our testbed below.

6D Pose Estimation

In 6D pose estimation, the task is to determine the three-dimensional position and orientation of an object in a scene. Concretely, for our purposes, models are given as input a single 128×128 RGB image of an object and must determine the object's 6 degree-of-freedom pose (rotation and translation) relative to the scene. For more background on pose estimation, see Lepetit and Fua [133] or Xiang et al. [245] and the references therein.

We evaluate each model using the accuracy metric from Hinterstoisser et al. [104]. Specifically, given a ground truth rotation R and translation t, estimated rotation $\tilde{R}$ and translation $\tilde{t}$, and a 3D model $\mathcal{M}$ consisting of m points $x \in \mathcal{M}$, then average distance (ADD) metric of Hinterstoisser et al. [104] is the mean of the distances between 3D model points transformed under the ground-truth and estimated poses

$$\text{ADD} = \frac{1}{m} \sum_{x \in \mathcal{M}} \|(Rx + t) - (\tilde{R}x + \tilde{t})\|_2.$$

An estimated 6D pose is consider to be *correct* if the ADD is less than 10% of the diameter of the 3D model $\mathcal{M}$.

YCB-Objects Datasets.

Similar to Xiang et al. [245] and Tremblay et al. [235], we construct a synthetic datasets for 6D pose estimation by rendering images of known object models from Calli et al. [39] and Hashimoto et al. [93] using Blender [24]. We use the subset of 16 non-symmetric YCB objects from Xiang et al. [245], as well as the two non-symmetric objects from Hashimoto et al. [93] in our experiments.

In our datasets, each object is placed on a plane with one of 60 textures from `texturehaven.com` and rendered with lighting from one of 60 HDRIs from `hdrihaven.com`. To generate distribution shift, we separate the textures into two, non-overlapping subsets based on their material properties. The in-distribution test set uses one subset of textures, and the out-of-distribution test set uses the other. See Figure 4.13 for example images corresponding to the in and out-of-distribution textures and corresponding datasets.

We generate datasets by uniformly sampling an object, a background lighting environment, a background texture (from the in or out-of-distribution subset), an object pose, and a camera pose. We generate in-distribution training sets of 50,000 and 100,000 images and

both in and out-of-distribution test sets of 10,000 images. In this section, we use the 50,000 example training set for our experiments. We use the 100,000 example training set to explore the effect of adding more i.i.d. training set in Appendix 4.11.

In simulation, the object model, the object pose, and the camera pose are all known in advance, so we can easily compute a ground truth pose for each object relative to the scene. We additionally annotate each image in our dataset with 9 2D keypoints corresponding to projection of the 3D bounding box and the 3D center of the object onto the 2D image. Figure 4.13 visualizes these annotations for a random sample of images from the training set.

YCB-Objects Models

The neural pose estimation models in our testbed are all based on semantic segmentation networks for predicting 2D keypoints. In essence, each network takes as input the entire image of the scene and predicts the nine keypoints previously described and shown in Figure 4.13. Some implementations in the literature first estimate bounding boxes of the objects in the scene before passing the images to the keypoint prediction network. Since our scenes only contain a single object, the networks in our testbed do not perform this step.

Given 2D keypoints predictions, each model then uses the PnP algorithm [134] to recover the 3D object pose. This approach, developed by Rad and Lepetit [181] and Pavlakos et al. [168] is also used in high-performing implementations like Tremblay et al. [235]. Our testbed contains several models for the semantic segmentation backbone. Unless otherwise noted, the implementation is taken from `github.com/qubvel/segmentation_models.pytorch`.

1. UNet [196] with ResNet [96], MobileNet [202], and EfficientNet-b7 [227] as the encoder.
2. UNet++ [262].
3. FCN_ResNet with varing depths 18, 34, 50, and 101 [145], using the implementation from `github.com/pytorch/vision/tree/master/torchvision/models/segmentation`.
4. LinkNet [43].
5. PSPNet [260].
6. PoseNet [116].
7. 2-layer CNN

Each of these models outputs a set of nine heatmaps, one corresponding to each keypoint prediction. For each model, we use the PnP implementation from Bradski [30].

4.11 Appendix: Further experiments on the linear trend phenomenon

In this section, we present additional examples of linear trends between in-distribution and out-of-distribution performance across each of the testbeds discussed in Appendix 4.10. In Appendix 4.11, we first highlight examples of linear trends across a variety of distribution shifts for models in each of the CIFAR-10, FMoW-WILDS, ImageNet, and the YCB-Objects "universes" discussed previously. Then, in Appendix 4.11, we show these linear trends are invariant to changes in model hyperparameters, training duration, and training set size.

Further examples of linear trends

CIFAR-10

Dataset reproduction shifts. In Figure 4.7, we plot out-of-distribution test accuracy vs. in-distribution CIFAR-10 test accuracy for each of the CIFAR-10 testbed models described in Appendix 4.10 on two different dataset reproduction shifts: CIFAR-10.1, CIFAR-10.2. For each shift, the relationship between in-distribution and out-of-distribution test accuracy for both classical and neural models is well captured by a linear fit, and the corresponding R^2 statistic is greater than 0.99 for each example.

Distribution shifts between machine learning benchmarks. In Figure 4.7, we also plot out-of-distribution test accuracy vs. in-distribution CIFAR-10 test accuracy for each of the CIFAR-10 testbed models described in Appendix 4.10 on two different machine learning benchmark shifts: CINIC-10, and STL-10. The class structure of STL-10 differs slightly from CIFAR-10 and includes a monkey class instead of a frog class. For the STL-10 experiment we therefore consider nine-class variants of STL-10 and CIFAR-10, omitting instances with monkey or frog labels, and, for each model, we mask the frog class (or logit) and predict only among the remaining nine classes. The relationship between ID and OOD accuracy is well-captured by a linear fit and the R^2 statistic is greater than 0.99 in each case.

Synthetic perturbations. In Figure 4.8, we plot out-of-distribution test accuracy vs. in-distribution CIFAR-10 test accuracy for the same collection of CIFAR-10 testbed models on a subset of eight different synthetic dataset shifts from CIFAR-10-C [98] where very clean linear trends occur— fog, brightness, snow, defocus blur, spatter, elastic transform, frost, and saturate. For each shift, the linear fit well-approximates the relationship between in-distribution and out-of-distribution accuracy, and the R^2 statistic is greater than 0.94 for each example. However, the fits are not as clean as the machine learning benchmark shifts discussed previously, and, moreover, for several of the synthetic perturbations in CIFAR-10-C, there is no linear trend at all. We discuss examples from CIFAR-10-C where linear trends fail to hold further in Section 4.4 and Appendix 4.12.

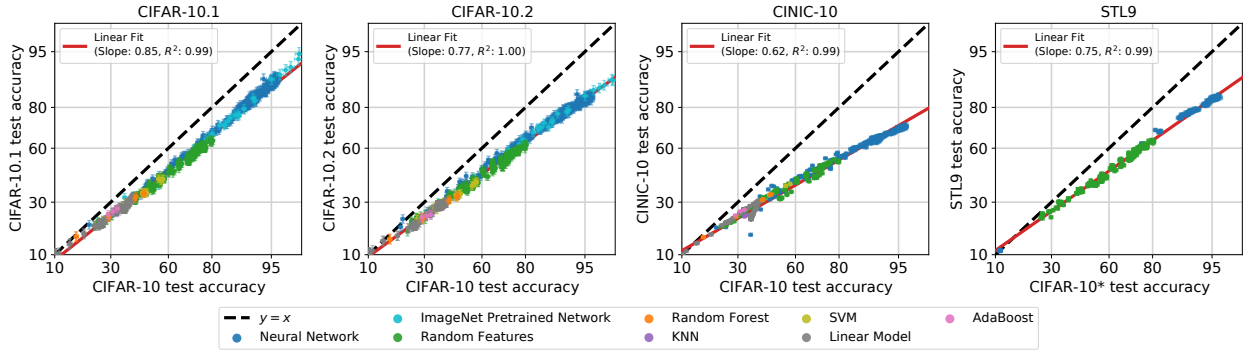


Figure 4.7: Out-of-distribution accuracies vs. in-distribution CIFAR-10 test accuracies for a wide range of models across two different dataset reproduction shifts, CIFAR-10.1 and CIFAR-10.2, as well as two different shifts between machine learning benchmarks, CINIC-10, and STL-10. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals (mostly covered by the point markers). For the STL-10 experiment, we consider nine-class subsets of both STL-10 and CIFAR-10, omitting the monkey and frog class, respectively, and restrict each model to predict only from the remaining nine-classes.

FMoW-WILDS

In Figure 4.9, we plot out-of-distribution test accuracy vs. in-distribution FMoW-WILDS test accuracy for both the classical methods and the ImageNet networks from the main testbed described in Appendix 4.10. We evaluate each model on both the out-of-distribution validation and the out-of-distribution test set from FMoW-WILDS using two metrics: average accuracy and worst accuracy over five geographical regions (for more details on FMoW-WILDS, see Appendix 4.10). To remove noise from very low accuracy models, we restrict our attention to models with FMoW-WILDS test set accuracy at least 10%. Across both out-of-distribution datasets and both metrics, the linear fit well-captures the relationship between in and out-of-distribution performance with an R^2 statistic of a least 0.98.

Experimental details. Below, we provide additional technical details about our FMoW-WILDS experiments.

- **Datasets.** We train each model on the training split of the FMoW-WILDS dataset [50] defined by Koh et al. [120], and perform testing on the in-distribution (ID) and out-of-distribution (OOD) validation and test splits defined by Koh et al. [120].
- **Worst-region accuracy confidence intervals.** We heuristically obtain confidence intervals for the worst-region accuracy by computing standard 95% Clopper-Pearson confidence intervals for accuracy in the region with the lowest accuracy on the test set for each model.

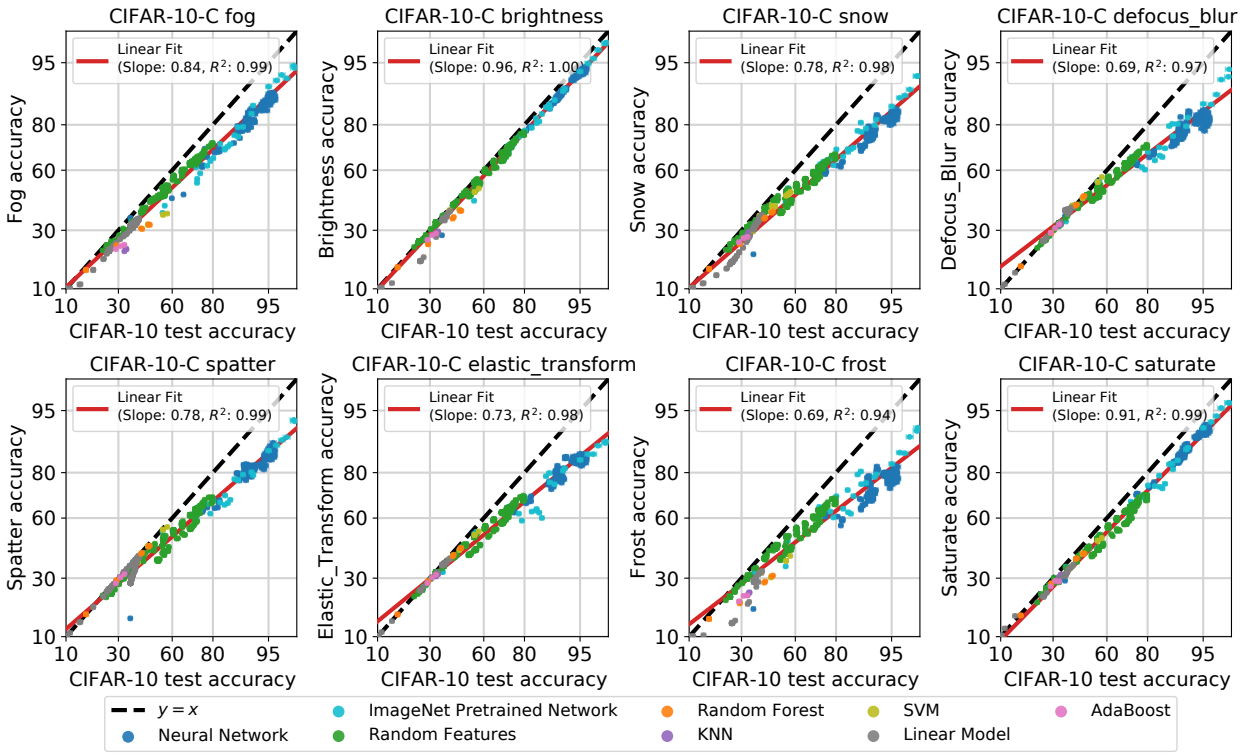


Figure 4.8: Out-of-distribution accuracies vs. in-distribution CIFAR-10 test accuracies for a wide range of models from our CIFAR-10 testbed across eight *synthetic perturbation* shifts from CIFAR-10-C. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals (mostly covered by the point markers).

- **Training hyperparameters.** Unless otherwise noted, we train all of the neural models using learning rate 10^{-4} and weight decay 0 for 50 epochs. We use Adam throughout with all other parameters set to their default PyTorch values.

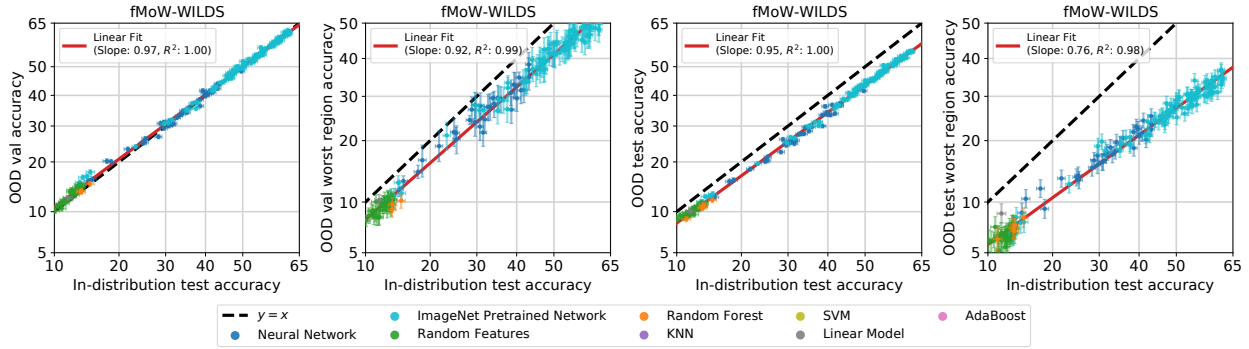


Figure 4.9: Out-of-distribution accuracies vs. in-distribution FMoW-WILDS test accuracies for a wide range of classical methods and ImageNet networks from our main testbed. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals (mostly covered by the point markers). **Left**: In the left two plots, we evaluate each model on the FMoW-WILDS OOD *validation set* using both average and worst-region accuracy. **Right**: In the right two plots, we evaluate each model on the FMoW-WILDS OOD *test set* using both average and worst-region accuracy. In all four cases, a linear fit well captures the relationship between in-distribution and out-of-distribution performance with R^2 statistics greater than 0.98 in each setting.

ImageNet

In Figure 4.10, we plot the existing models from Taori et al. [228] alongside the new models in our testbed (CLIP, self-supervised models, and classical methods like random features, random forests, nearest neighbors, and linear regression) on the ImageNet natural distribution shifts. First, we observe that the two CLIP models are significantly robust on all shifts (these models are the two green points above the line at around 60% ImageNet accuracy). This is interesting and is in line with our conclusions that pretraining on extra data can increase model robustness to distribution shift. Second, we observe that all three low-accuracy models lie relatively near the predicted fit line for ImageNetV2, ImageNet-Vid-Anchors, ImageNet-Sketch, and ImageNet-R. Note that this line is fit only to the standard neural networks (blue points). Understanding why the fit isn't as predictive in the low-accuracy regime for ObjectNet, YTBB-Anchors, and ImageNet-A is an interesting direction for future work. Note that the fit to ImageNet-A is performed piecewise around the ResNet-50 model accuracy following the procedure in Taori et al. [228].

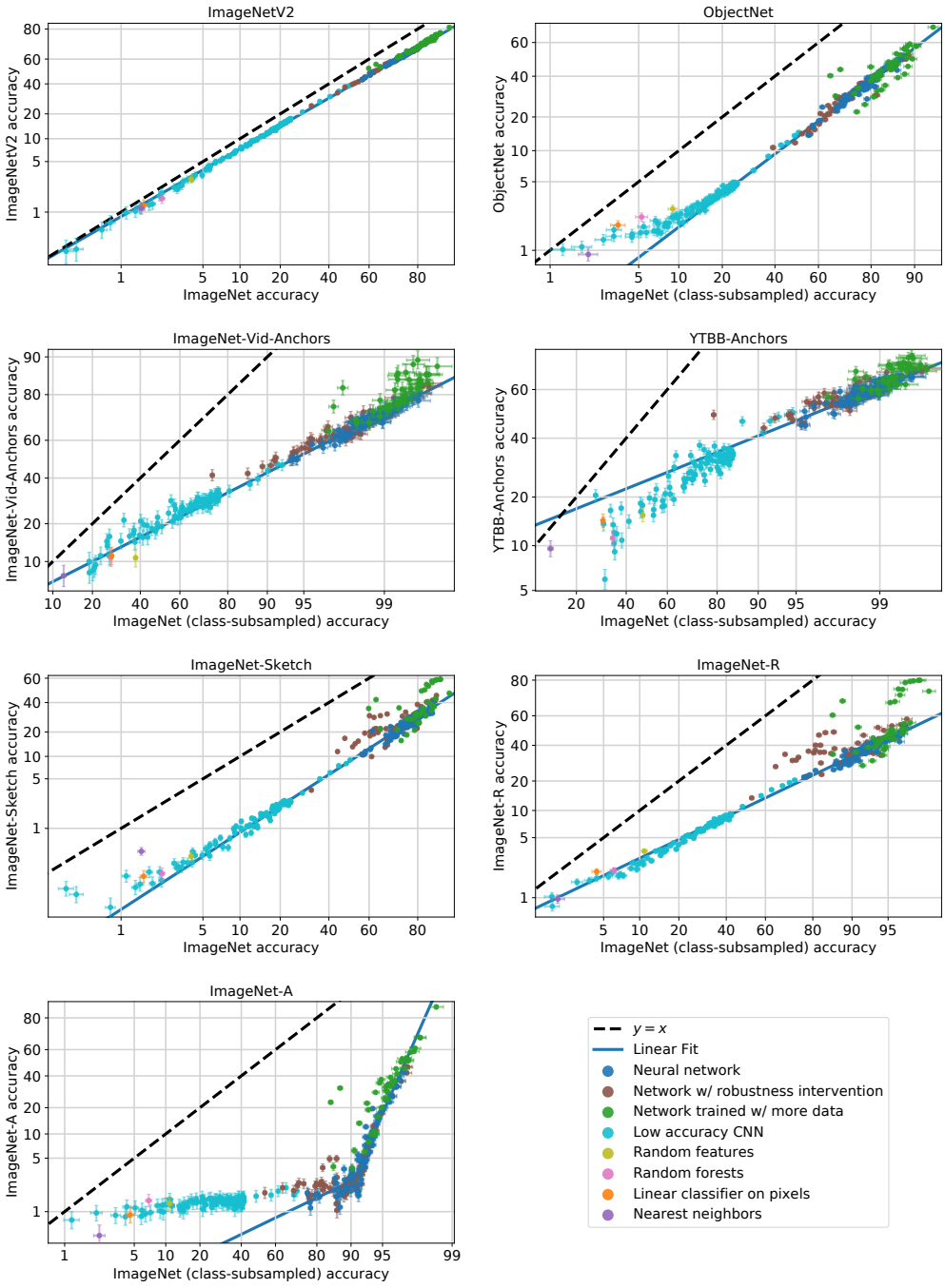


Figure 4.10: Model accuracies on the ImageNet natural distribution shifts [228]: ImageNetV2, ObjectNet, ImageNet-Vid-Anchors, YTBB-Anchors, ImageNet-Sketch, ImageNet-R, and ImageNet-A. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals. The axes are scaled via logit scaling, and the linear fit is fit only to standard networks following [228].

iWildCam-WILDS

Experiment details. The models reported in the iWildCam-WILDS panel of Figure 4.1 were obtained using the following parameters. We trained 10 neural network architectures on the iWildCam-WILDS training set (see legend of Figure 4.11 below). For each architecture, we perform both training from scratch and fine-tuning from a model pretrained on ImageNet. The fine-tuning configurations are similar to the setting of Koh et al. [120]: we train for 12 epochs with batch size 16 using Adam and sweep over learning rate and weight decay values in the grid $\{3\cdot 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\} \times \{0, 10^{-3}, 10^{-2}\}$. The other Adam parameters were set to the Pytorch defaults. For models trained from scratch we use the same hyperparameter grid except we train for 15 epochs, which seems to suffice for convergence for each model with at least some of the learning rates. For details on error bar calculation and label noise reduction, see Appendix 4.10.

Architecture variation with fixed weight decay. Figure 4.11 provides a more detailed view of the iWildCam-WILDS experiments, wherein we plot the final epoch performance of the models we train, where the weight decay is set to zero. As the figure shows, the error bars for all model intersect the fitted linear trend line (in probit domain), with the exception of AlexNet when training from scratch, which is slightly below the linear trend. Varying the weight decay parameter appears to affect the ID/OOD trend of fine-tuned model; see Section 4.5 and Appendix 4.13 for additional discussion and plots.

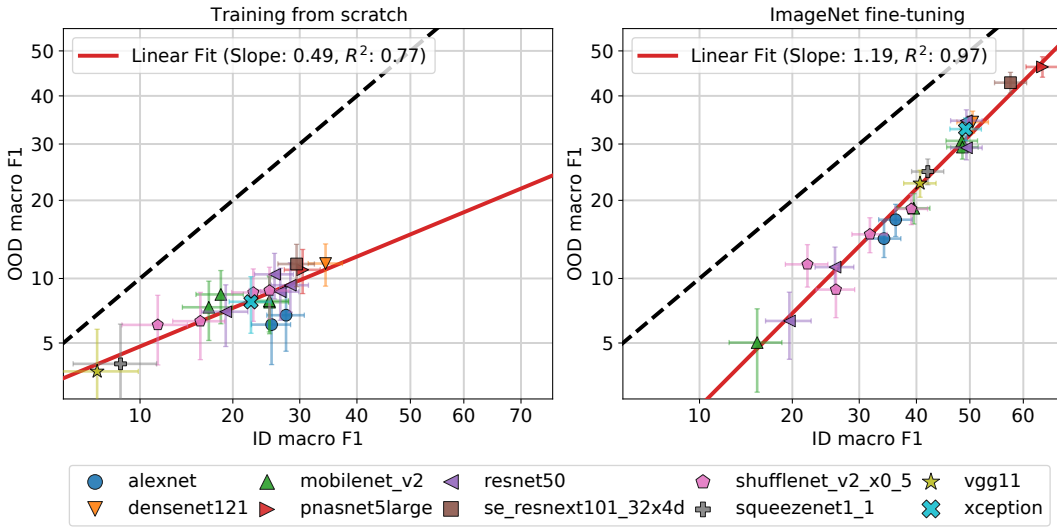


Figure 4.11: OOD vs. ID macro F1 scores for iWildCam-WILDS models trained from scratch (left) or fine-tuned from pretrained ImageNet models (right), with varying model architecture and learning rate, but weight decay fixed to zero. Contrast with Figure 4.32 for results when varying the weight decay parameter.

YCB-Objects

In Figure 4.12, we plot out-of-distribution accuracy versus in-distribution accuracy for a synthetic 6D pose estimation task using the YCB object models from Calli et al. [39] and a testbed of neural models for 6D pose estimation. As in the previous examples, a linear fit well-approximates the relationship between in and out-of-distribution accuracy with an R^2 statistic of 0.99.

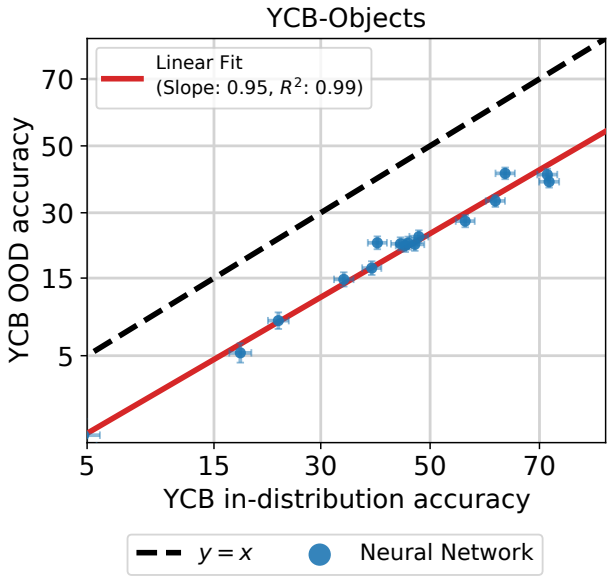


Figure 4.12: Out-of-distribution accuracy vs. in-distribution accuracy for a synthetic 6D pose estimation task based on the YCB object models from Calli et al. [39] across a testbed of neural pose estimation networks. Each point corresponds to a model evaluation shown with 95% Clopper-Pearson confidence intervals. The distribution shift corresponds to varying the background texture for rendered images of the YCB objects. See Figure 4.13 for example images both in and out-of-distribution. Appendix 4.11 describes the dataset and the model testbed in more detail.

Experimental details. We train two variants of each model. The first variant is trained with standard ℓ_2 loss on the distance between the predicted heatmap and the ground truth keypoint location (with a Gaussian blur of $\sigma = 0.2$). These models predict keypoint locations by taking an arg max over the heatmap. The other variant is trained with and makes predictions using the integral pose regression technique of [220]. We train each model using SGD with momentum and learning rate annealing. For each model, we optimized the learning rate in $[10^{-4}, 10^{-1}]$ and weight decay in $[10^{-4}, 1]$.

(a) Example images from the YCB-Objects in-distribution test set. Each object is rendered on a background whose texture has similar material properties.

(b) Example images from the YCB-Objects out-of-distribution test set. The distribution shift corresponds to rendering objects on a held out set of textures with a different set of material properties than the in-distribution textures. Aside from the texture change, the set of objects, the lighting environments, and the sampling distribution for objects, poses, and lighting is held fixed between datasets.

(c) Examples images from the YCB-Objects in-distribution test set shown with keypoint annotations. Each image is annotated with nine keypoints corresponding to the corners of the 3D bounding box and the object center. Models in the testbed predict these keypoints, and the object's 6D pose is recovered from keypoints using the PnP algorithm [134].

Variations in model hyperparameters, training duration, and training dataset size

In this section, we explore the sensitivity of the linear trends discussed in Appendix 4.11 to variation in model hyperparameters, training duration, and training set size.

We focus much of our exploration on two datasets CIFAR-10 and FMoW-WILDS. We selected CIFAR-10 for ease of experimentation, and we selected FMoW-WILDS in order to understand the sensitivity of the linear trends outside the context of machine learning benchmark or synthetic shifts.

CIFAR-10

In Figures 4.14, 4.15, and 4.16, we probe the sensitivity of the linear trend between in and out-of-distribution test accuracy for CIFAR-10 models to three types of variation: variation in hyperparameters, variation in training duration, and variation in training set size. For ease of visualization, we focus our experiments on three model families spanning low, moderate, and high accuracy regimes: a ridge regression classifier on image pixels, the random feature model from Coates, Ng, and Lee [52], and a ResNet [97]. The results are virtually identical, but harder to visualize, when considering a larger number of model families simultaneously.

We systematically vary the hyperparameters, number of training epochs (for ResNets), and the size of the training set for models from each class. We plot model evaluations on the same linear trend line as found in Appendix 4.11. We show variation along these three dimensions moves models along the linear trend line for each dataset, but does not change the linear fit. For each of the dataset reproduction shift CIFAR-10.2 (Figure 4.14), the benchmark shift CINIC-10 (Figure 4.15), and the synthetic CIFAR-10-C fog shift (Figure 4.16), the R^2 statistic of the fit is greater than 0.99.

Experimental details. We briefly provide details about the specific variations we consider for each model class.

1. **Hyperparameter variation:** For the ridge regression classifier, we vary the ℓ_2 regularization parameter in $[10^{-6}, 10^{10}]$. For the random features models, we vary the ℓ_2 regularization parameter in $[10^{-4}, 10^{6}]$ and the number of random features in $[2^0, 2^{14}]$. For the ResNet model, we vary network depth in $\{18, 34, 50, 101\}$, learning rate in $[10^{-5}, 10]$, momentum in $[0.33, 0.99]$, and weight decay in $[10^{-5}, 10^{5}]$.

2. **Training duration variation:** To understand sensitivity to training duration, we save and evaluate each ResNet model after every epoch of training. We train each model for 350 epochs, giving 350 evaluations per run.

3. **Training set size variation:** To understand sensitivity to the amount of training data, we subsample the CIFAR-10 dataset from the original 50,000 samples to i.i.d. class-balanced subsets of size 1000, 5000, 10000, 15000, 25000, and 40000 examples. We

train each of the hyperparameter configurations for each model class on each of the 6 subsets of the original dataset and evaluate them on the same in and out-of-distribution test sets as before.

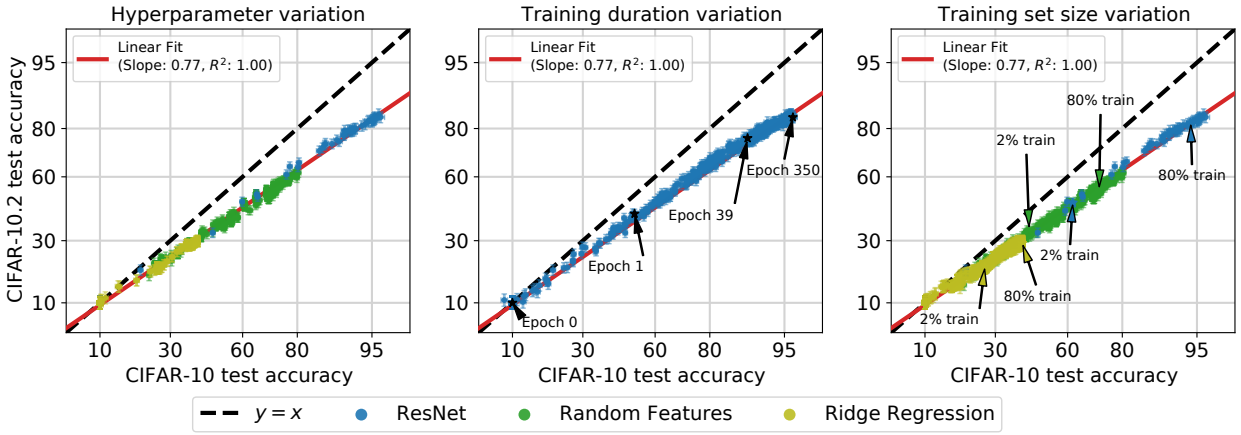


Figure 4.14: Out-of-distribution CIFAR-10.2 accuracies vs. in-distribution CIFAR-10 test accuracies under variations in model hyperparameters, training duration, and the size of the training set. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals (mostly covered by the point markers). In each panel, we compare models with the linear trend line from Appendix 4.11. **Left:** For each model family, we vary model-size, regularization, and optimization hyperparameters. **Middle:** We evaluate each network after every epoch of training. **Right:** We train models on randomly sampled subsets of the training data, ranging from 2% to 80% of the original CIFAR-10 training set size. In each setting, variation in hyperparameters, training duration, or training set size moves models along the trend line, but does not affect the linear fit.

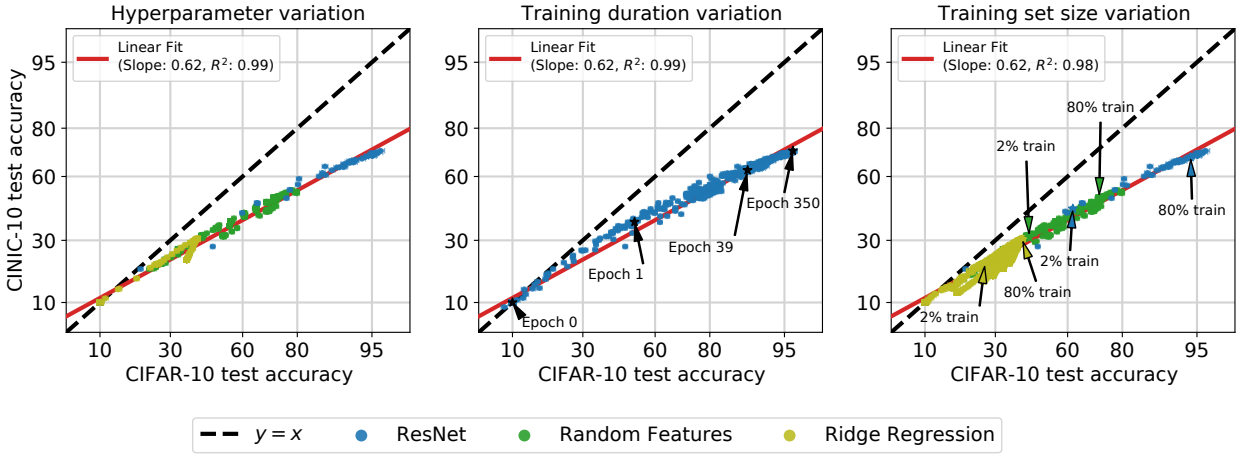


Figure 4.15: Out-of-distribution CINIC-10 accuracies vs. in-distribution CIFAR-10 test accuracies under variations in model hyperparameters, training duration, and the size of the training set. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals (mostly covered by the point markers). In each panel, we compare models with the linear trend line from Appendix 4.11. **Left:** For each model family, we vary model-size, regularization, and optimization hyperparameters. **Middle:** We evaluate each network after every epoch of training. **Right:** We train models on randomly sampled subsets of the training data, ranging from 2% to 80% of the original CIFAR-10 training set size. In each setting, variation in hyperparameters, training duration, or training set size moves models along the trend line, but does not affect the linear fit.

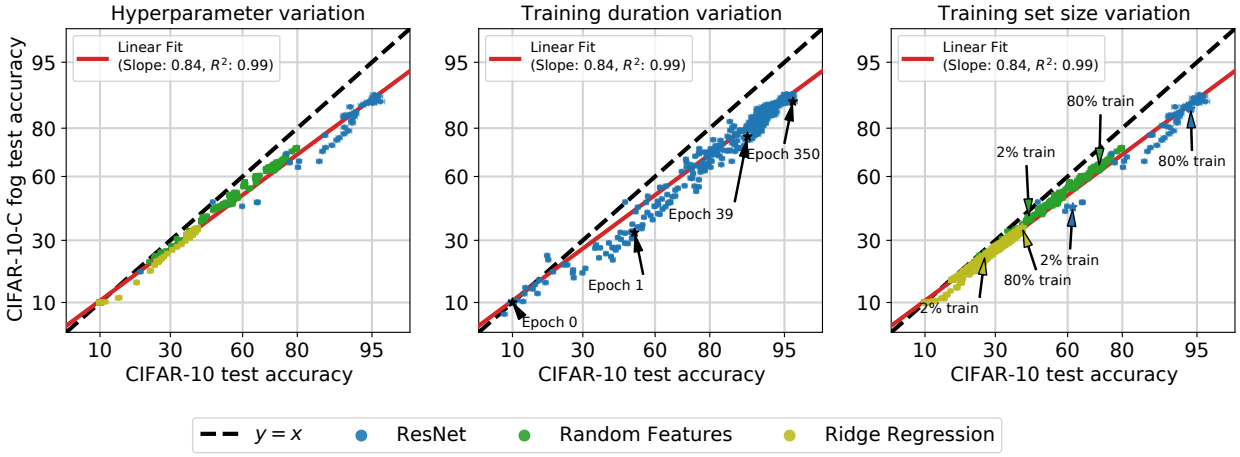


Figure 4.16: Out-of-distribution CIFAR-10-C fog accuracies vs. in-distribution CIFAR-10 test accuracies under variations in model hyperparameters, training duration, and the size of the training set. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals (mostly covered by the point markers). In each panel, we compare models with the linear trend line from Appendix 4.11. **Left:** For each model family, we vary model-size, regularization, and optimization hyperparameters. **Middle:** We evaluate each network after every epoch of training. **Right:** We train models on randomly sampled subsets of the training data, ranging from 2% to 80% of the original CIFAR-10 training set size. In each setting, variation in hyperparameters, training duration, or training set size moves models along the trend line, but does not affect the linear fit.

FMoW-WILDS

As in the previous section, in Figure 4.17, we probe the sensitivity of the linear trend for FMoW-WILDS models to variation in hyperparameters, variation in training duration, and variation in training set size. For ease of visualization, we focus our experiments on three model families spanning low, moderate, and high accuracy regimes: a random forest model on image pixels, the random feature model from Coates, Ng, and Lee [52], and a ResNet [97]. We plot model evaluations on the same linear trend line as found in Appendix 4.11. We show variation along these three dimensions moves models along the linear trend line for each dataset, but does not change the linear fit: the R^2 statistic of the fit is greater than 0.99 for every setting under the accuracy metric and greater than 0.91 for the worst-region accuracy metric.

Experimental details. We briefly provide details about the specific variations we consider for each model class.

1. **Hyperparameter variation:** For the random forest classifier, we vary the maximum depth in $\{1, 3, 10, 20\}$, the number of trees in $\{10, 20, 50, 200\}$, and the splitting criterion between entropy and gini impurity. For the random features models, we vary the ℓ_2 regularization parameter in $[10^{-4}, 10^6]$ and the number of random features in $[2^0, 2^8]$. For the ResNet model, we vary network depth in $\{18, 34, 50, 101\}$, learning rate in $[10^{-5}, 10]$, momentum in $[0.33, 0.99]$, and weight decay in $[10^{-5}, 10^5]$.

2. **Training duration variation.** We train each configuration of the ResNet for 70 epochs and evaluate each model after every epoch of training.

3. **Training set size variation.** We i.i.d. subsample the FMoW-WILDS train dataset from the original 76,863 examples to subsets of size 1000, 5000, 10000, 20000, and 50000 examples. We train each of the hyperparameter configurations for each model class on each of the 5 subsets of the original dataset and evaluate them on the same in and out-of-distribution test sets as before.

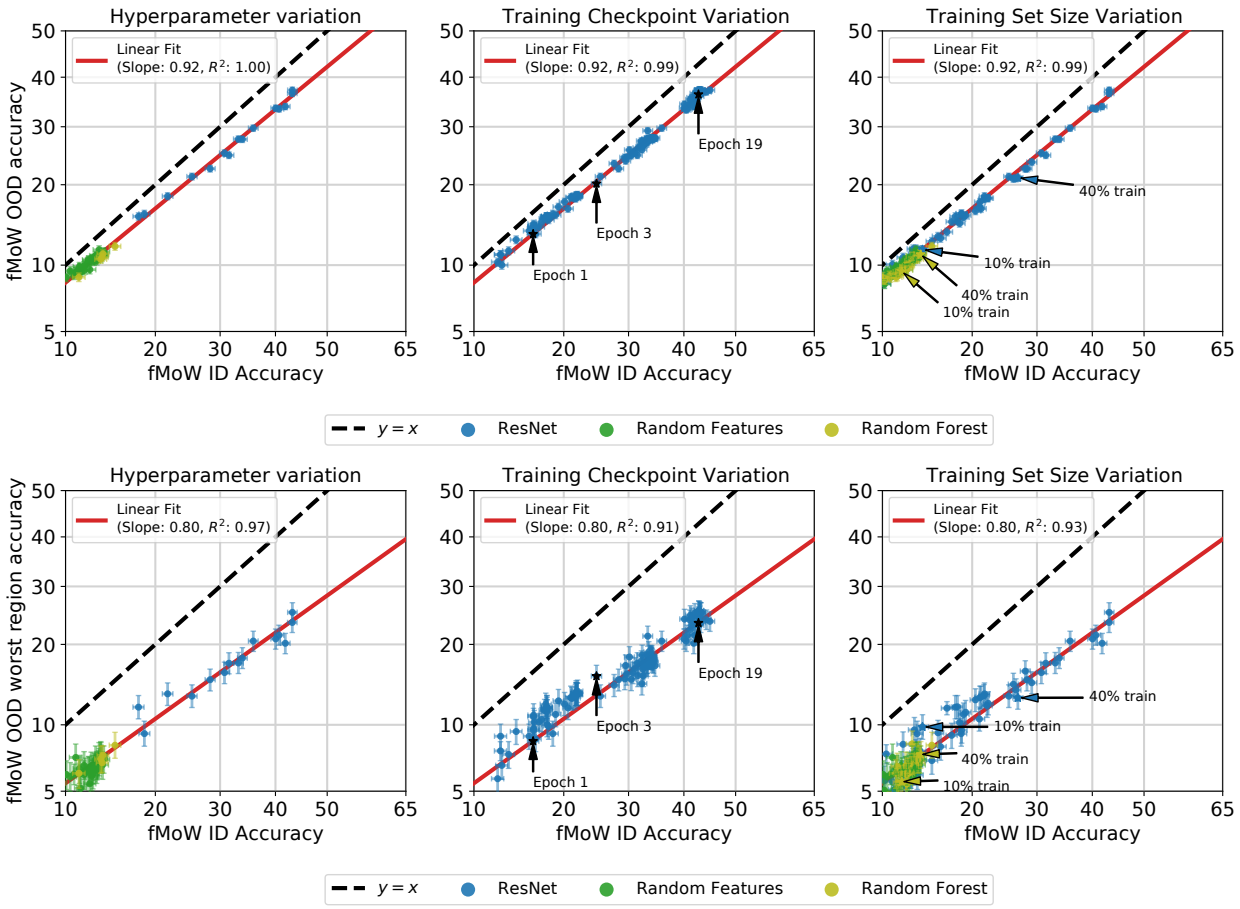


Figure 4.17: Out-of-distribution FMoW-WILDS test accuracies vs. in-distribution FMoW-WILDS test accuracies under variations in model hyperparameters, training duration, and the size of the training set. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals. In each panel, we compare models with the linear trend line from Appendix 4.11. The top row compares model trends using average accuracy as the OOD metric, and the bottom rows uses worst-region accuracy as the OOD metric. **Left:** For each model family, we vary model-size, regularization, and optimization hyperparameters. **Middle:** We evaluate each network after every epoch of training. **Right:** We train models on randomly sampled subsets of the training data, ranging from 2% to 80% of the original CIFAR-10 training set size. In each setting, variation in hyperparameters, training duration, or training set size moves models along the trend line, but does not affect the linear fit.

YCB-Objects

In Figure 4.18, we see that the linear fit for the YCB-Objects experiment from Appendix 4.11 is also invariant to changes in the amount of training data.

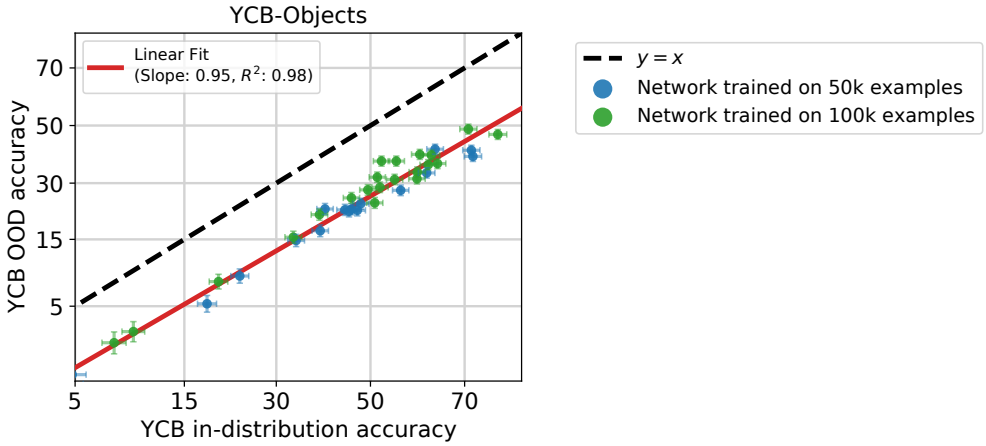


Figure 4.18: Out-of-distribution YCB-Objects test accuracies vs. in-distribution YCB-Objects test accuracies under variations in training set size. Each point corresponds to a model evaluation, shown with 95% Clopper-Pearson confidence intervals. The linear trend line is the same as Figure 4.12. The linear trend still well explains the data ($R^2 = 0.98$), and increasing training set size moves models along the linear trend, but does not affect the linear fit.

Comparison of axis scaling

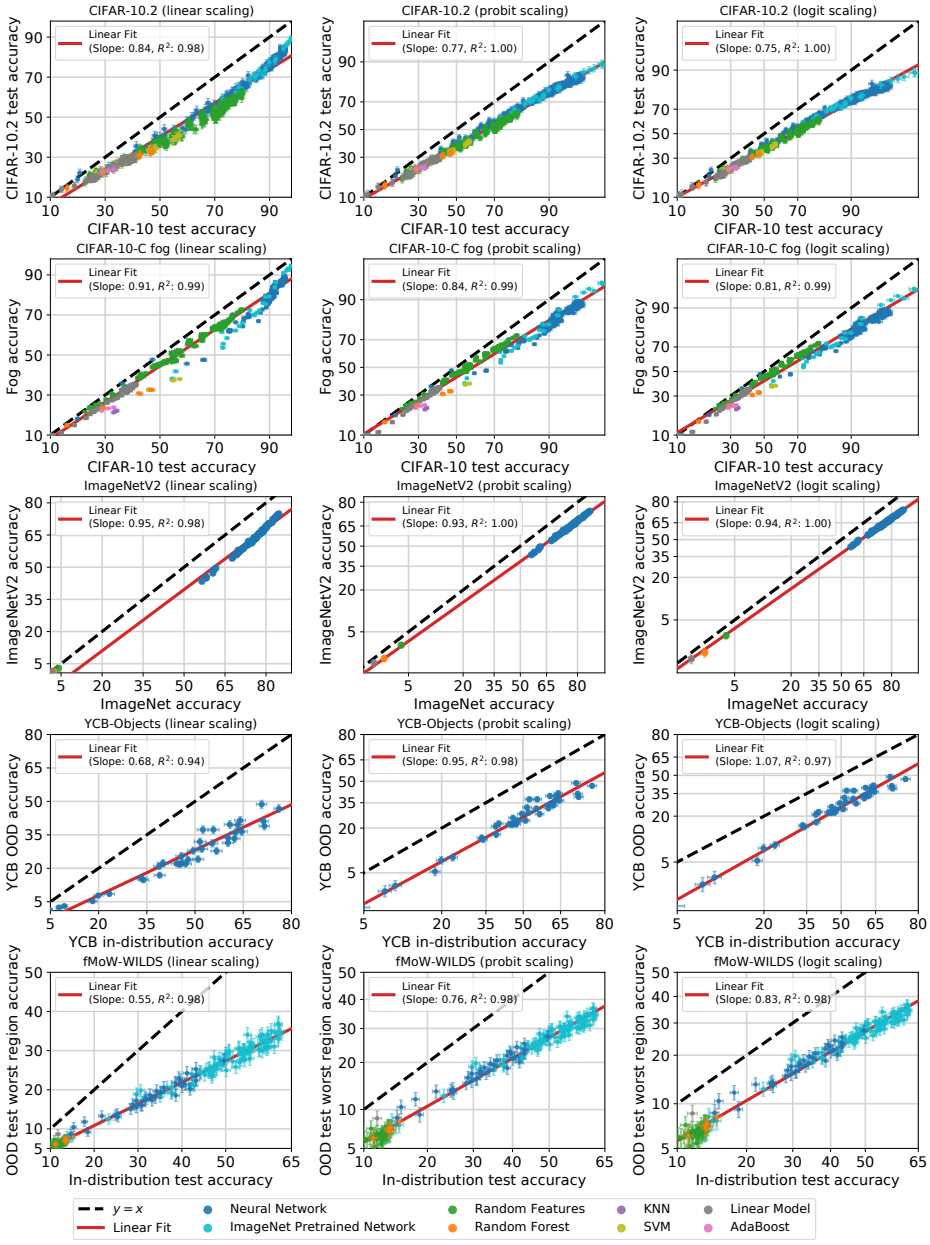


Figure 4.19: OOD test accuracies vs. ID test accuracies for several pairs of ID and OOD test sets visualized with three different axis scalings. **Left:** The left column shows model accuracies with a linear axis scaling. **Middle:** The middle column shows model accuracies with a probit scale on both axes. In other words, model accuracy x appears at $\Phi^{-1}(x)$ where Φ^{-1} is the inverse Gaussian CDF. **Right:** The right column shows model accuracies with a logit scale on both axes: model accuracy x appears at $\sigma^{-1}(x)$ where σ^{-1} is the inverse logistic function. Visual inspection shows the linear fit is better in the logit or probit domain, especially when model accuracies span a wide range. Quantitatively, the R2 statistics are higher in the probit or logit domains than with linear axis scaling. For instance, on ImageNetV2 and CIFAR-10.2, the R^2 is 0.98 in the linear domain compared to 1.0 in the probit or logit domains.

4.12 Appendix: Details on distribution shifts with weaker correlations

Camelyon17-WILDS

In this section, we first explore the role of training randomness on the observed ID-OOD correlation for Camelyon17-WILDS. Remember that in Figure 4.3, we found a very high degree of variability between ID and OOD performance. To see if the performance variation was due to training randomness, we train each model ten times and then average the final model accuracies together. The result of these averaged runs is displayed in Figure 4.20 (left). The R^2 value for the averaged runs comes in at $R^2 = 0.39$, which is approximately equivalent to the R^2 value in Figure 4.3 ($R^2 = 0.40$); this suggests that training randomness is not enough to account for the performance variability.

In Figure 4.20 (right), we also attempted early-stopping each trained model on a separate OOD validation set (different in distribution from the OOD test set), as is recommended in [120], before averaging model accuracies; the result is largely unchanged and comes in at $R^2 = 0.46$. Early-stopping on the in-distribution validation set, however, does increase ID-OOD correlation significantly to $R^2 = 0.77$, as seen in Figure 4.20 (middle); further investigating the mechanisms at play here is an interesting direction for future work.

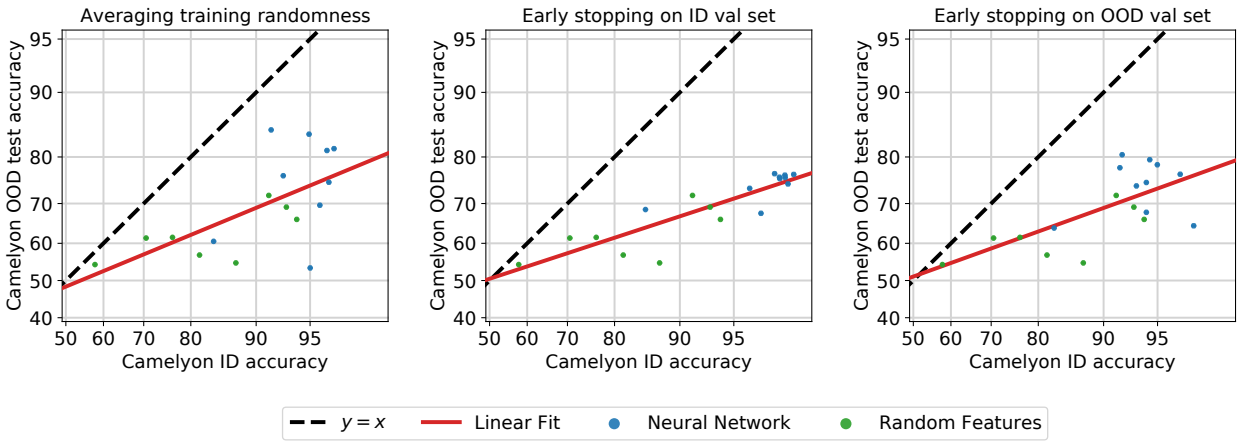


Figure 4.20: Model accuracies on the Camelyon17-WILDS distribution shift. Each point gives average accuracies for models trained with ten different random seeds, and error bars give the standard deviation. **Left:** Models trained to convergence and then averaged over seeds. **Middle:** Each model is early-stopped on the ID validation set then averaged over seeds. **Right:** Each model is early-stopped on the OOD validation set then averaged over seeds.

As a next step, to study the role of the training data distribution on the observed trends, we conduct two specific training-time interventions: pretraining on ImageNet, and training

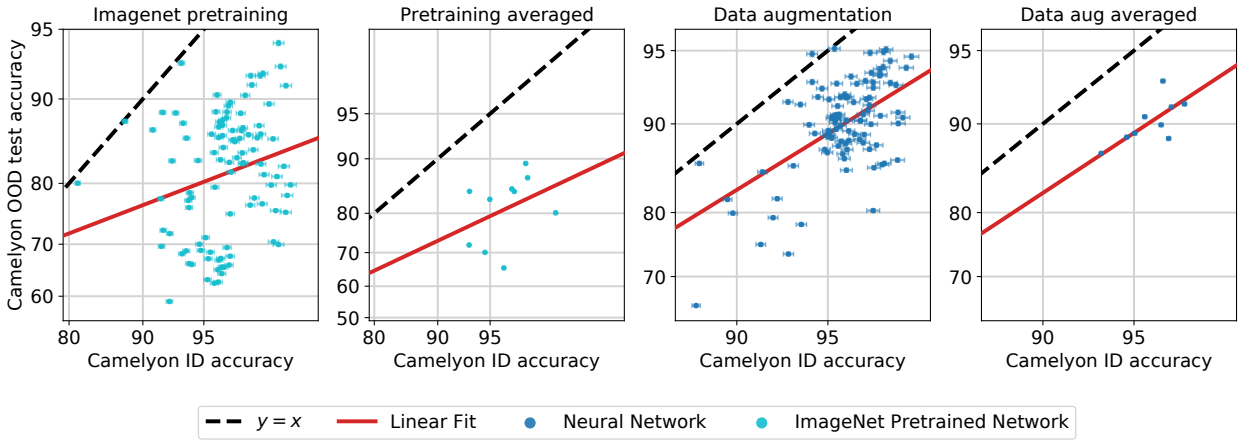


Figure 4.21: Model accuracies on the Camelyon17-WILDS distribution shift. **Left:** ImageNet pretrained models finetuned to convergence. **Middle Left:** ImageNet pretrained model accuracies averaged over ten random seeds. **Middle Right:** Models trained with targeted color jitter data augmentation. **Right:** Data augmentation model accuracies averaged over ten random seeds.

using a specific color-jitter data augmentation.

We show results for models pretrained on ImageNet in Figure 4.21 (left). As is evident, the variability in model performance is still extremely high ($R^2 = 0.05$). Averaging over training randomness does not seem to help either ($R^2 = 0.14$).

We also train using a domain-specific color-jitter data augmentation designed to mimic the visual differences in samples from different hospitals, a technique that has previously been found to have been beneficial on a similar task [230, 229]. As seen in Figure 4.21 (middle right), training with the data augmentation both considerably increases average OOD performance and significantly reduces the amount of OOD accuracy variation ($R^2 = 0.77$). However, even with the targeted data augmentation, large OOD accuracy fluctuations still exist. Averaging over training randomness greatly increases the correlation further and mitigates these fluctuations ($R^2 = 0.95$), as seen in Figure 4.21 (right); however, the data augmentation causes all models to have relatively high ID accuracy, and it is unclear whether this tight trend would hold for models in the low accuracy regime as well.

One possible reason for the high variation in accuracy is the correlation across image patches. Image patches extracted from the same slides and hospitals are correlated because patches from the same slide are from the same lymph node section, and patches from the same hospital were processed with the same staining and imaging protocol. In addition, patches in Camelyon17-WILDS are extracted from a relatively small number of slides (the dataset includes 50 slides total, and there are 10 slides from a single hospital in the OOD test set [120]). Prior work in the context of natural language processing tasks have shown that these correlations can result in instabilities in both training and evaluation [261], and

investigating their effect on OOD variation in Camelyon17-WILDS is interesting future work.

As an initial exploration of the effect of highly correlated test examples, we observe that correlated examples can result in high OOD variation in a simulated environment on CIFAR-10 and CIFAR-10.2. Concretely, we subsample CIFAR-10 and CIFAR-10.2 and then apply data augmentation to each example to generate a test set of the same size as the original but with significant correlation between examples. In each panel in Figure 4.22, we train models on CIFAR-10 and then evaluate them on CIFAR-10 and CIFAR-10.2 with effective test size k for varying k. Concretely, we subsample k images from each class, and then apply RandAugment `rand-m9-mstd0.5-inc1` [58] to each example to generate test sets of size 10,000. We work with a binary version of CIFAR-10 and CIFAR-10.2, restricting both datasets to two classes: `airplanes` and `cats`. When the effective test set size is small, e.g. $k = 1$ or $k = 2$, the linear fit is very poor. However, as the effective test set size k increases to $k = 100$ or $k = 500$, the linear fit is much better ($R^2 = 0.94$ vs. $R^2 = 0.66$), and the variance between model evaluations is substantially smaller.

In contrast to highly correlated test examples, highly correlated *training* examples appears to have substantially less effect on the amount of OOD variation or the quality of the linear fit. Using the same simulated CIFAR-10 and CIFAR-10.2 environment as the previous paragraph, we generate a sequence of training sets with varying degrees of correlation between training examples. Concretely, we subsample the CIFAR-10 and CIFAR-10.2 training sets and then apply data augmentation (RandAugment [58]) to each example to generate a training set of the same size as the original but with significant correlation between examples. In each panel in Figure 4.23, we train models on CIFAR-10 and then evaluate them on CIFAR-10. Even with the effective training set size is small, e.g. $k = 2$, the linear fit is fairly good ($R^2 = 0.89$), and there is substantially smaller variance between model evaluations than in the corresponding effective test-size experiment.

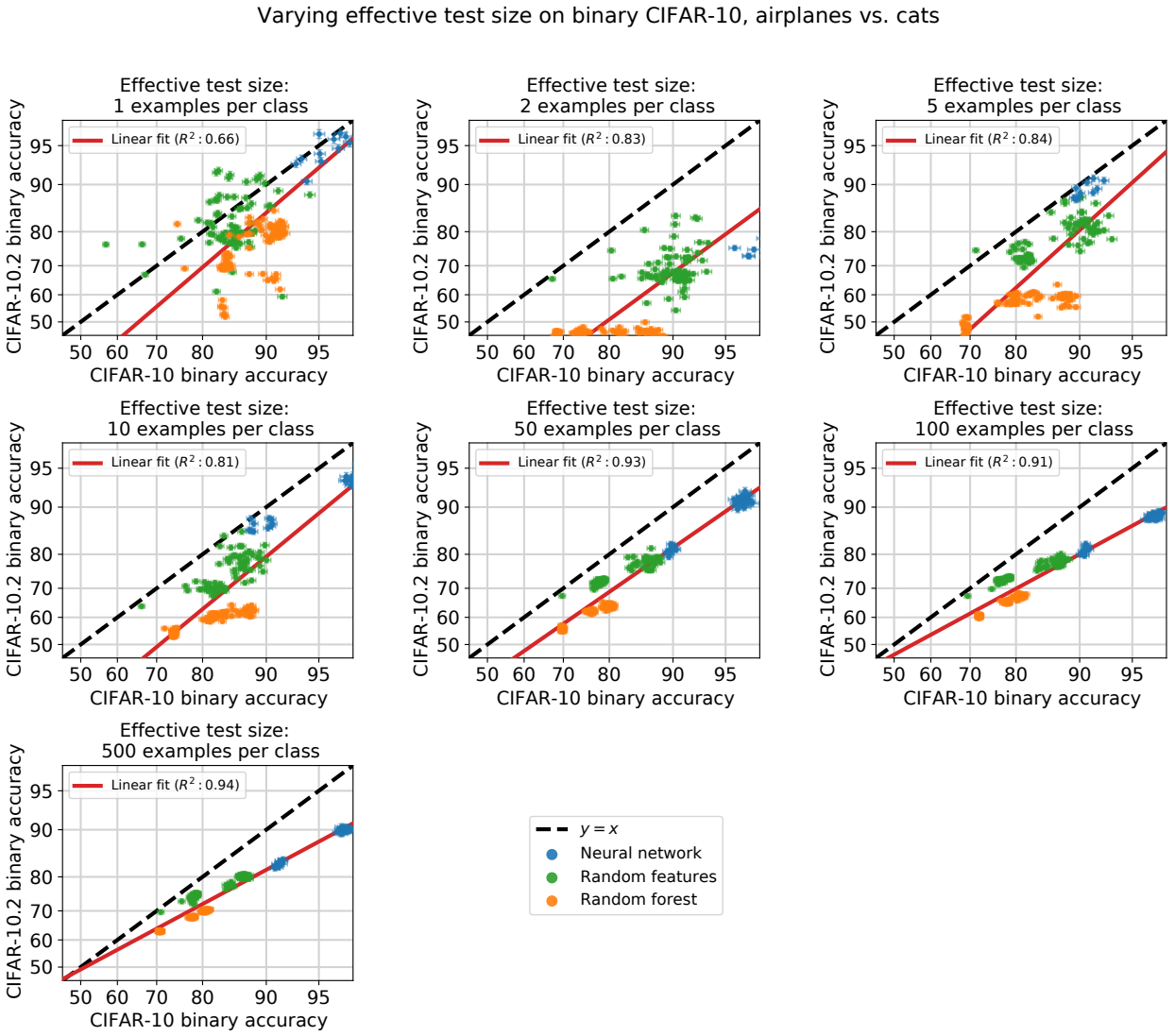


Figure 4.22: Models trained on CIFAR-10 and evaluated on CIFAR-10.2 for binary classification: `airplanes` vs `cats`. Each panel depicts evaluating the models with varying effective test set sizes k, where k images are subsampled from each class and then repeated data-augmented using RandAugment [58] to generate a consistent test set size of 10,000 examples. For smaller effective test set sizes, the linear fit is very poor, and this variance decreases substantially for larger k.

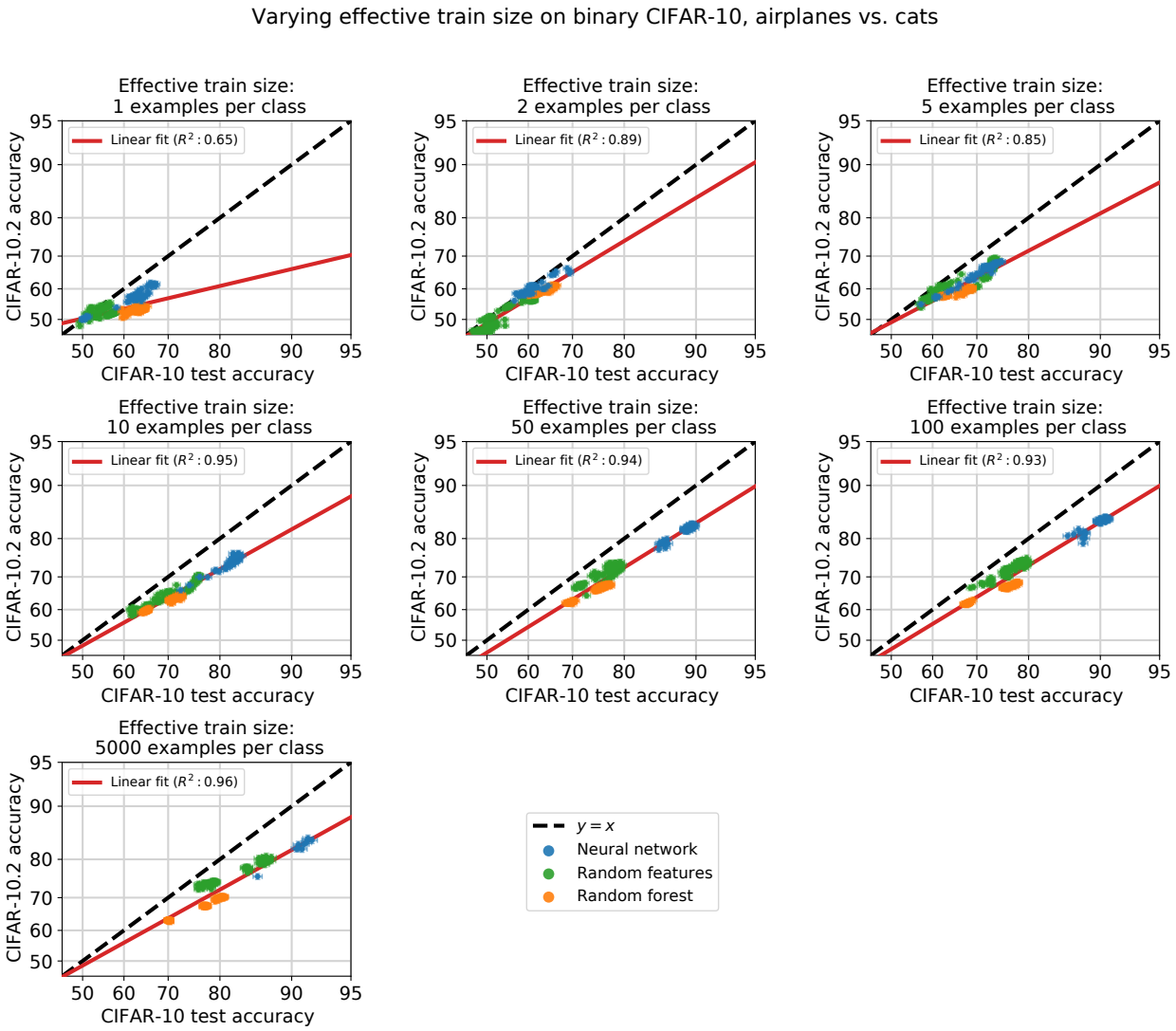


Figure 4.23: Models trained on CIFAR-10 and evaluated on CIFAR-10.2 for binary classification: `airplanes` vs `cats`. Each panel depicts evaluating the models with varying effective train set sizes k, where k images are subsampled from each class and then repeatedly data-augmented using RandAugment [58] to generate a consistent train set size of 50,000 examples. In contrast to varying the effective test set size (see Figure 4.23), varying the effective train set has little effect on the quality of the linear fit. For instance, with as few as two effective examples per class, the linear fit is fairly precise ($R^2 = 0.89$).

CIFAR-10-C

In this section, we look at distribution shifts induced by image corruptions in more detail. Specifically, in Figures 4.24–4.28, we plot neural networks trained on either CIFAR-10 or ImageNet and evaluated on a similar set of image corruptions. Interestingly, the choice of corruption can have a significant effect on the strength of the linear trend between ID and OOD accuracy, as we have already explored in Sections 4.3 and 4.4. Comparing the plots in Figures 4.24–4.28 side-by-side, we also observe that many corruptions behave more linearly on ImageNet-C than on CIFAR-10-C. Investigating this discrepancy further is an interesting direction for future work.

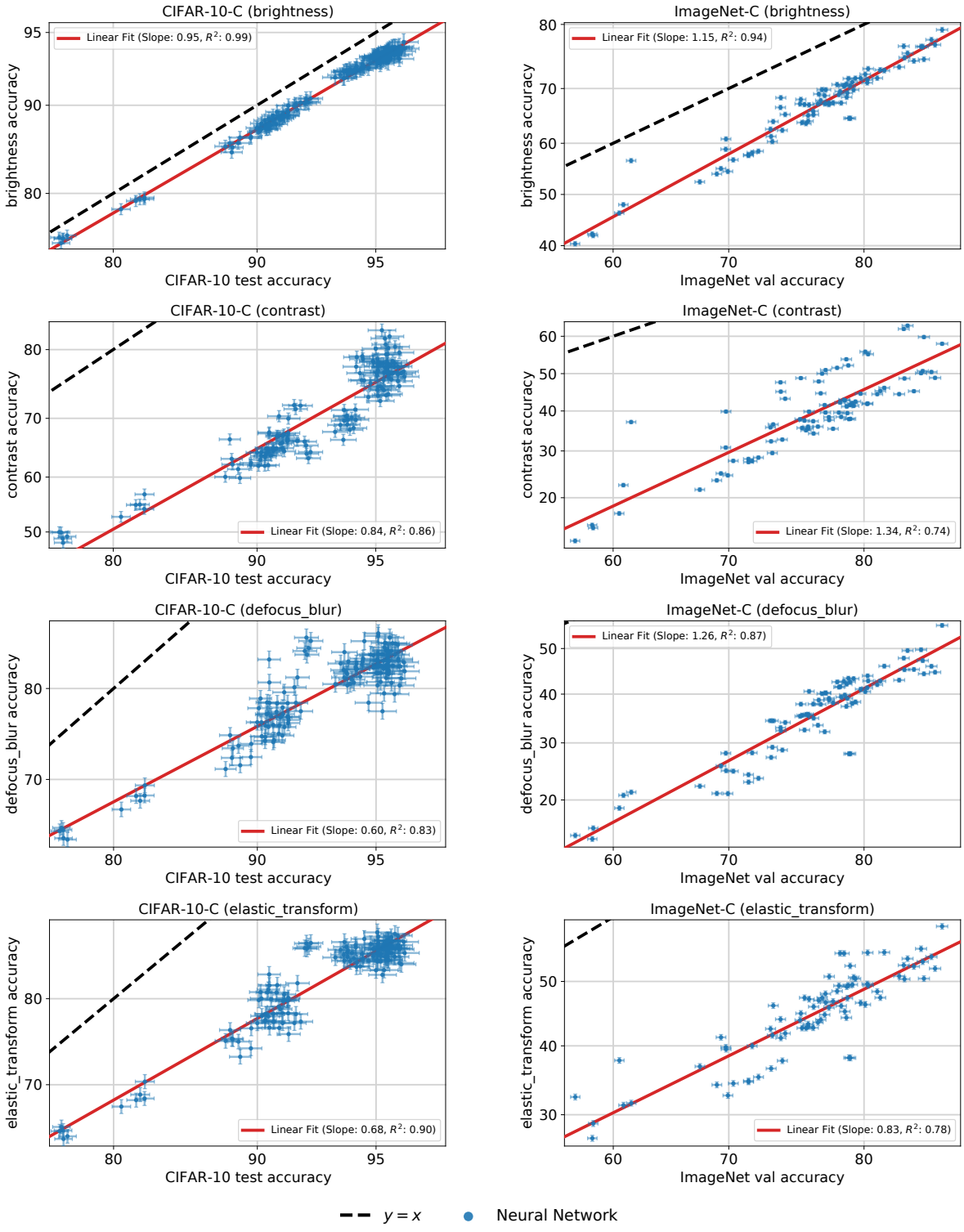


Figure 4.24: Models trained on either CIFAR-10 (left) or ImageNet (right) and evaluated under distribution shift due to image corruptions. This figure continues for the next few pages.

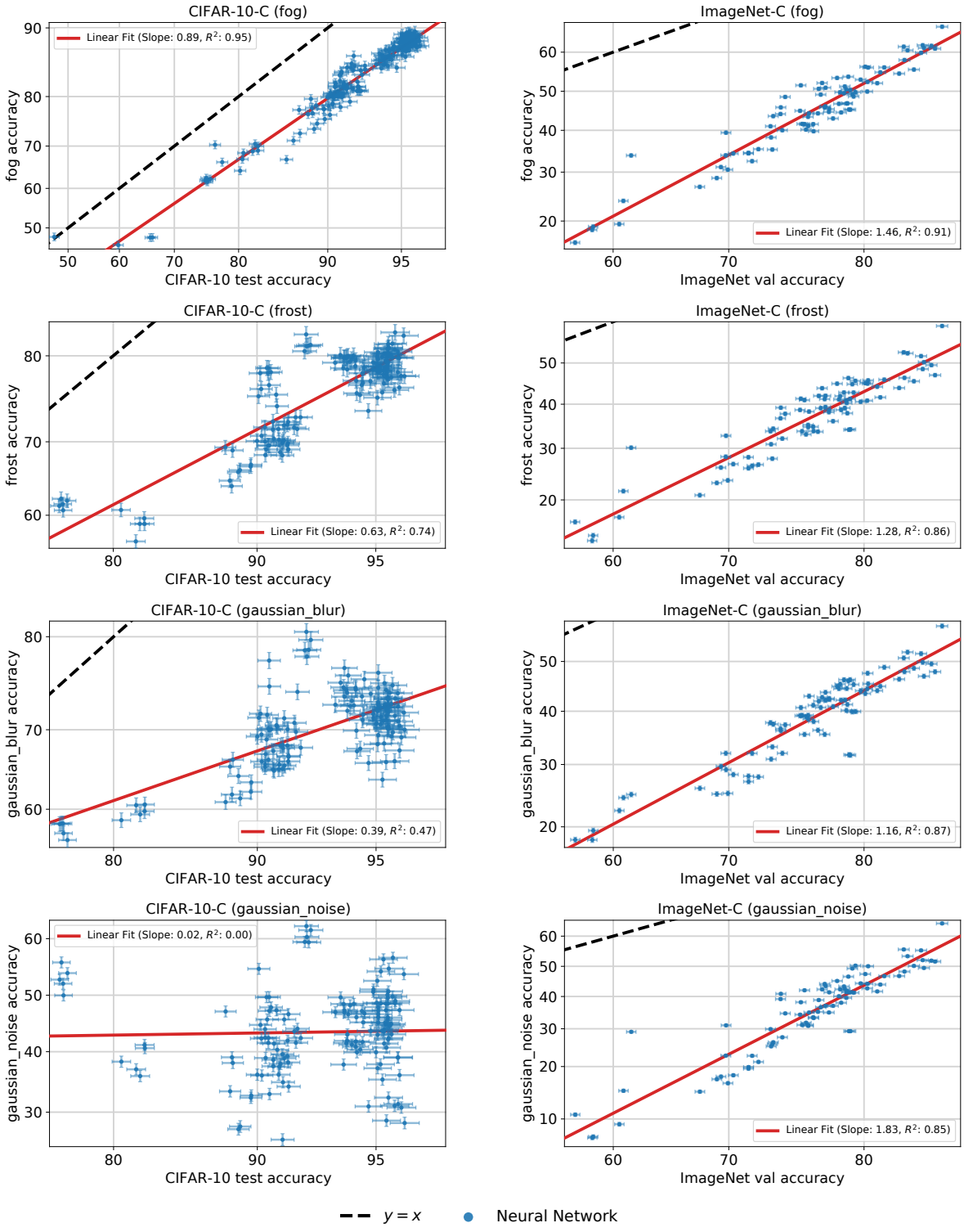


Figure 4.25: Continuation of the corruption plots.

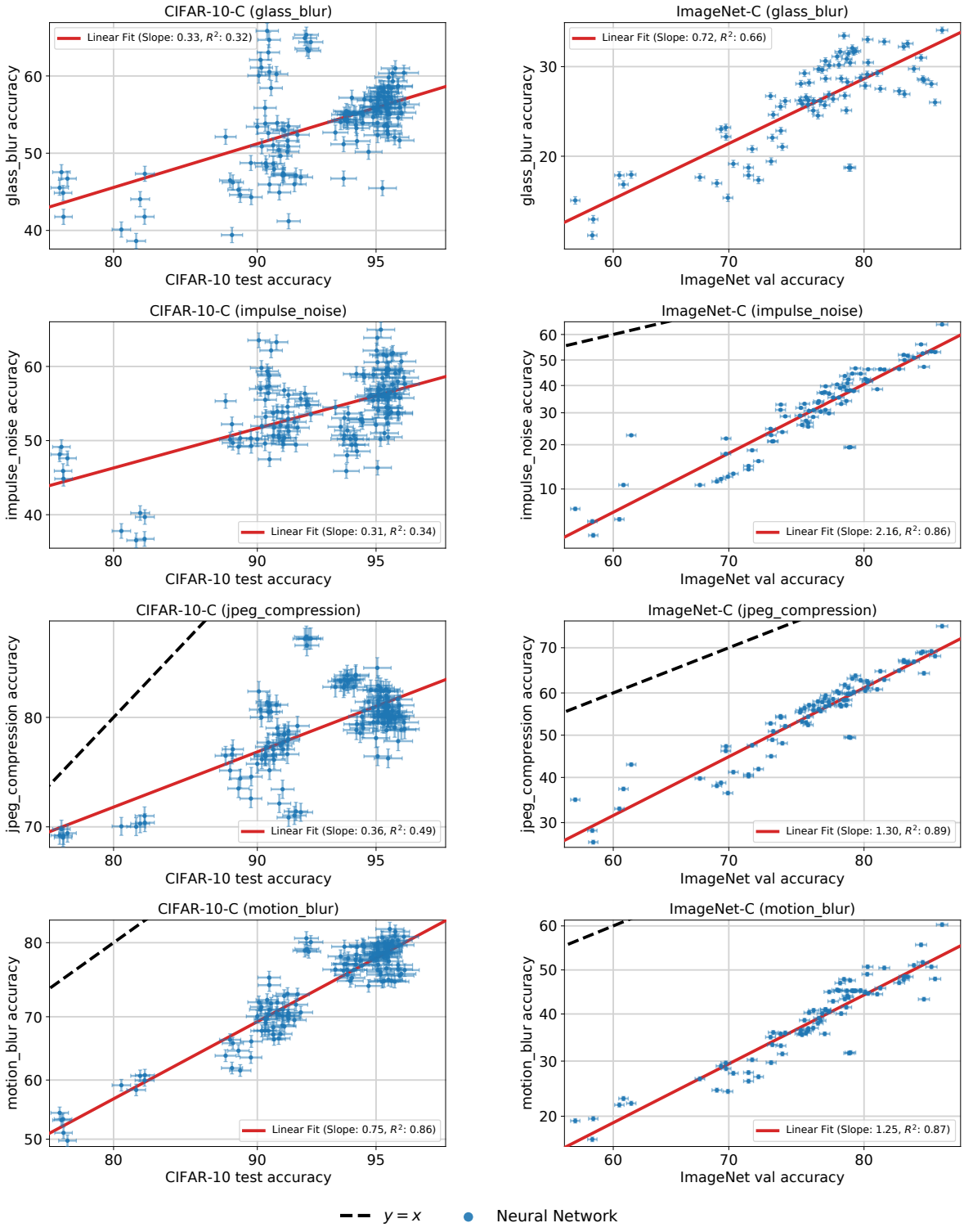


Figure 4.26: Continuation of the corruption plots.

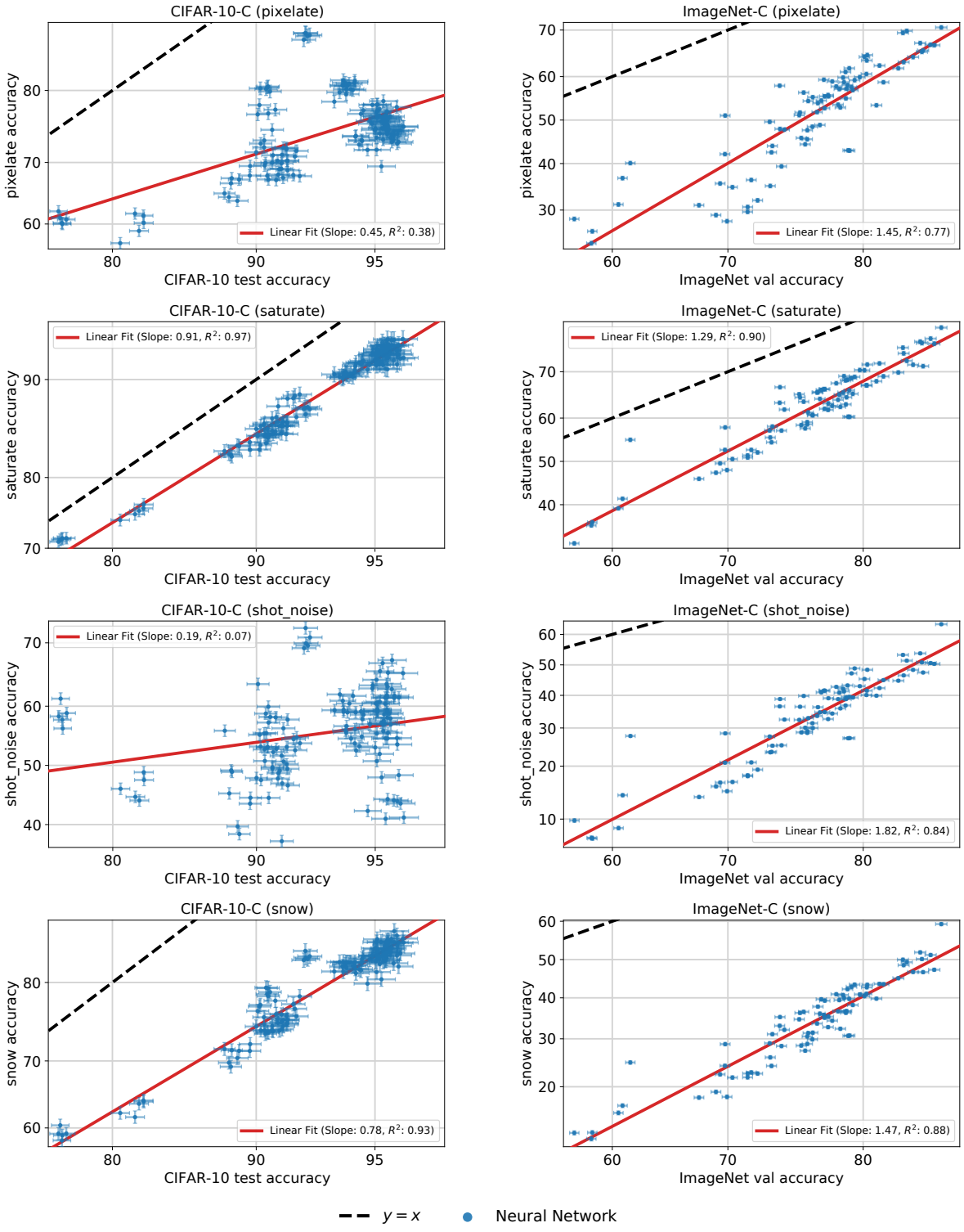


Figure 4.27: Continuation of the corruption plots.

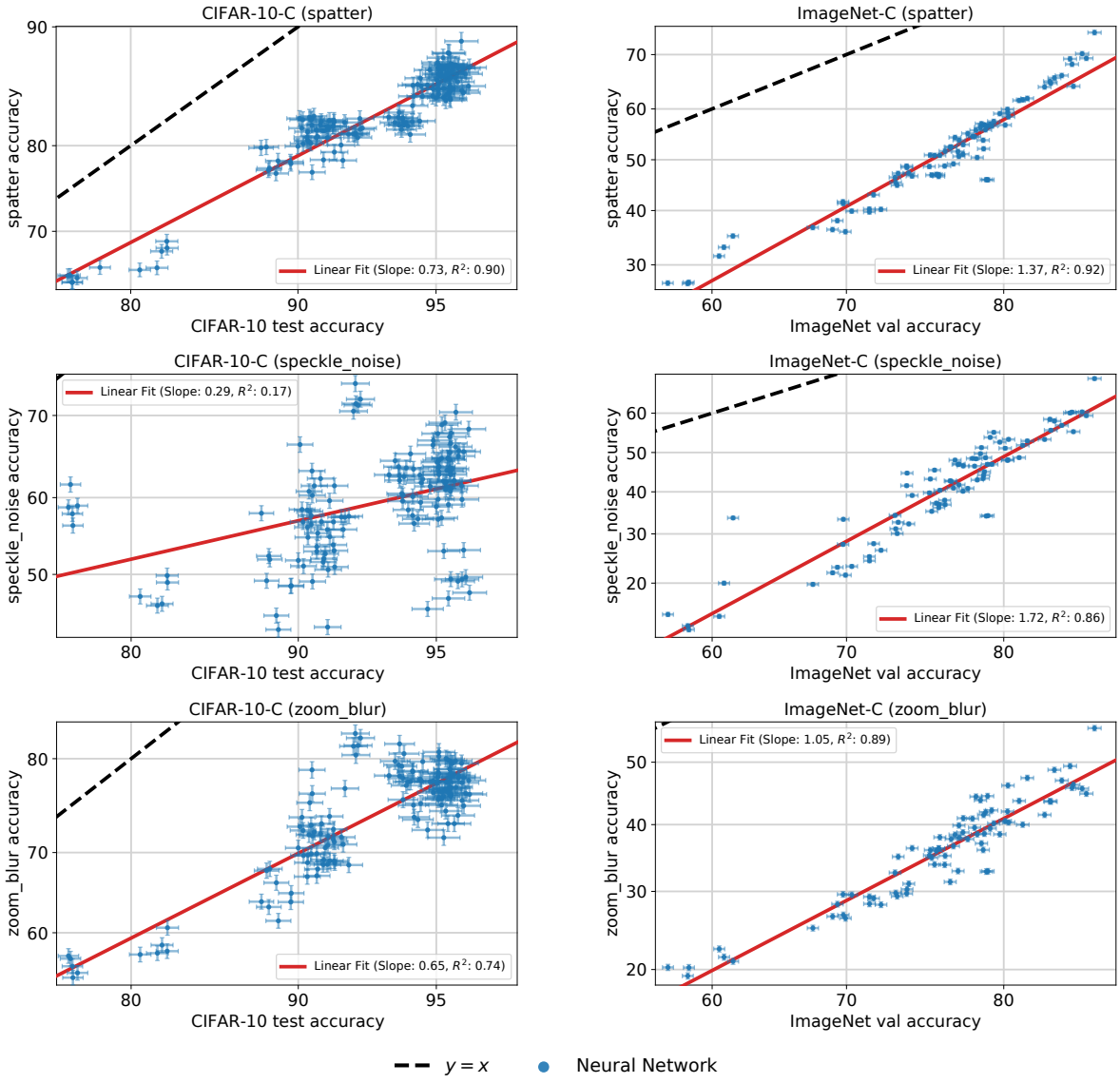


Figure 4.28: Continuation of the corruption plots.

iWildCam-WILDS-v1.0

We now study version 1.0 of iWildCam-WILDS (from WILDS version 1.0), which has a different split between training and ID test sets, compared to the version of iWildCam-WILDS we have studied thus far (iWildCam-WILDS version 2.0 from WILDS version 1.1). In version 1.0, images from training cameras are assigned uniformly at random between the train and ID test sets, whereas images are randomly partitioned by date between train and ID test splits in version 2.0. Since the images tend to be taken in bursts, the earlier

version of the dataset contains some training and ID test examples that are taken within the same image sequence, and these images tend to be similar because they often capture the same animal at the same location. In other words, we are changing how we measure in-distribution performance, and in this way, our investigation on ID/OOD correlations study different distribution shifts between the two versions. Nevertheless, both versions of the dataset measure out-of-distribution performance in the same way, with train and OOD test splits containing images from disjoint cameras.

While we use version 2.0 in all other sections, it is still interesting to understand how a different in-distribution train-test split affects the ID/OOD correlation. In Figure 4.29, we repeat the experiment reported in Figure 4.11 on the v1.0 split.[4] As the figure shows, the ID/OOD correlation is far less pronounced when using the v1.0 split. Moreover, the fine-tuned models show a near-vertical line, with models concentrated around high ID accuracy values but spread across many OOD values, and this could potentially be explained by the high image similarity between train and ID test sets.

Finally, we remark that while the v2.0 split eliminates overlap in image sequence between train and ID test sets, some near-duplicates inevitably persist in that version as well, particularly for empty frames taken during similar times in the day by the same camera. Investigating the effect of this on the linear trend on iWildCam-WILDS v2.0, in which we observe higher variation in performance than in other datasets, is interesting future work.

4.13 Appendix: Details on the effect of pretrained models

Detailed findings for CIFAR-10

In Figure 4.30 (left) we reproduce the results shown in Figure 4.5 and add to it a number of additional models; Figure 4.30 (right) graphs the performance of the same model when measuring their OOD performance on CIFAR-10.1 instead of CIFAR-10.2. Let us describe the additional models and their relationship to the linear trend.

First, as a middle ground between zero-shot use of ImageNet models (which is above the line) and fine-tuning (which is on the line), we consider neural network models trained only on the subset of CINIC-10 that originates from ImageNet (as opposed to CIFAR-10). It is worth noting that in this case, the CINIC-10 subset includes images from ImageNet-21k, which is a superset of the more common ImageNet-1k dataset containing approximately 21,000 classes. Similar to the zero-shot case, these models use only ImageNet data and so we expect their accuracy to not obey the same CIFAR-10/CIFAR-10.2 relationship of models trained on CIFAR-10 data (in fact, for these models both CIFAR-10 and CIFAR-10.2 are

[4] There are some differences in training hyper-parameters: Our v1.0 experiments used images with resolution 224x224, slightly different learning rates and number of epochs, and no label noise reduction via MegaDetector-based filtering as described in Appendix 4.10. Nevertheless, we are confident that the primary cause for the difference between Figures 4.29 and 4.11 is the change in test/train split.

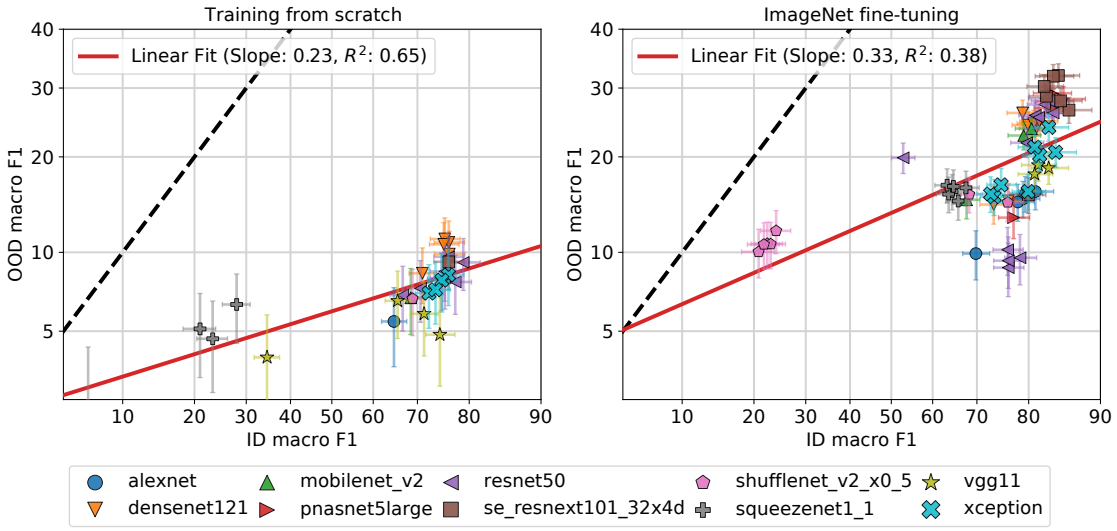


Figure 4.29: OOD vs. ID macro F1 scores for iWildCam-WILDS-v1.0 models trained from scratch (left) or fine-tuned from pretrained ImageNet models (right), with varying model architecture and learning rate, but weight decay fixed to zero. Contrast with Figure 4.11 for results on the v2.0 ID test/train split.

OOD). Similar to fine-tuned models, these models are specialized to the task of classifying only the 10 CIFAR-10 classes (as opposed to the 1000 ImageNet classes), and so we expect them to have better accuracy. Figure 4.30 lists these models as "Training on ImageNet data," and confirms our expectations: these models are above the linear fit and have better accuracy than the ImageNet zero-shot models. However, in comparison to the zero-shot models, they appear to lie closer to the linear fit for CIFAR-10-trained models.

Second, we consider two publicly released CLIP models [182], based on ResNet 50 and Vision Transformer, respectively. Both zero-shot application of CLIP and the training of only its final layer (denoted "linear probe") produce performance that is above the line, particularly for the higher-performing Vision Transformer. See below for additional details on the use of CLIP in our experiments.

Finally, we consider models trained on auxiliary unlabeled data originating from the 80 Million Tiny Images [234], abbreviated 80MTI below, which is a superset of both CIFAR-10 and its reproductions CIFAR-10.1 and CIFAR-10.2. In particular, we consider a model trained via self-training using a subset of TinyImages [41], listed as 80MTI ST in Figure 4.30), and two models trained via out-distribution aware self-training [5], listed as 80MTI ODST. As the figure shows, despite using auxiliary data, the "80MTI ST" performance is precisely on the CIFAR-10-only linear trend. This might be due to the fact that Carmon et al. [41] filter 80MTI, using a model supervised with the CIFAR-10 training set, thereby possibly losing

the additional diversity of TinyImages.[5] The performance of the ODST models appear to deviate from the linear trend. However, the direction of the deviation is inconsistent, being below the line on CIFAR-10.2 and above the line on CIFAR-10.1. Since these are the highest-accuracy models in our testbed it is not completely clear whether these deviations are due to use of extra data or a deviation of the overall ID-OOD trend from a perfect probit linear fit at high accuracies.

Experiment details. Below we provide some additional details on out CIFAR-10 auxiliary data experiments.

- **Zero-shot classification with ImageNet models.** To investigate models that are minimally affected by the CIFAR-10 training set, we utilized pre-trained ImageNet models directly for the CIFAR-10 classification task without any fine-tuning ("zero-shot"). A complication here is that the CIFAR-10 classes do not match the ImageNet classes. For instance, ImageNet contains more than 100 different dog classes corresponding to different breeds while CIFAR-10 contains only one dog class. To address this point, we manually constructed a mapping from CIFAR-10 classes to ImageNet classes. Our mapping roughly followed the WordNet hierarchy with some refinements from the class structure used in the human annotation experiments conducted by [210]. We then evaluated the ImageNet models using only the logits for classes appearing in this mapping and picked the CIFAR-10 class as prediction that corresponded to the ImageNet class with the largest logit.

- **Zero-shot classification with CLIP models.** For the models described as "CLIP zero-shot", we use the publicly released CLIP package, which includes the ResNet and VisionTransformer models as well as the text tokenizer and encoder used to encode the zero-shot text prompts. We obtained the CIFAR-10 prompts through private correspondence with the OpenAI team. For each CIFAR-10 class, we ensembled the prompts by averaging the embeddings of the prompts together before using it for final classification.

- **Linear probes.** In the models described above as "linear probes" we train only the last layer of a pre-trained neural network by performing (exact) least-squares linear regression of 1-hot class representation using the activations of the network's penultimate layer.

Detailed findings for FMoW-WILDS

Figure 4.31 shows a reproductions of Figure 4.5 (middle) when using different combination of worst-region and average-region accuracy metric for the ID and OOD data (recall that

[5]We also note that the additional unlabeled data used to train this model potentially contains images from CIFAR-10.1 and CIFAR-10.2; if it does, they seem to do little to help its performance on that dataset.

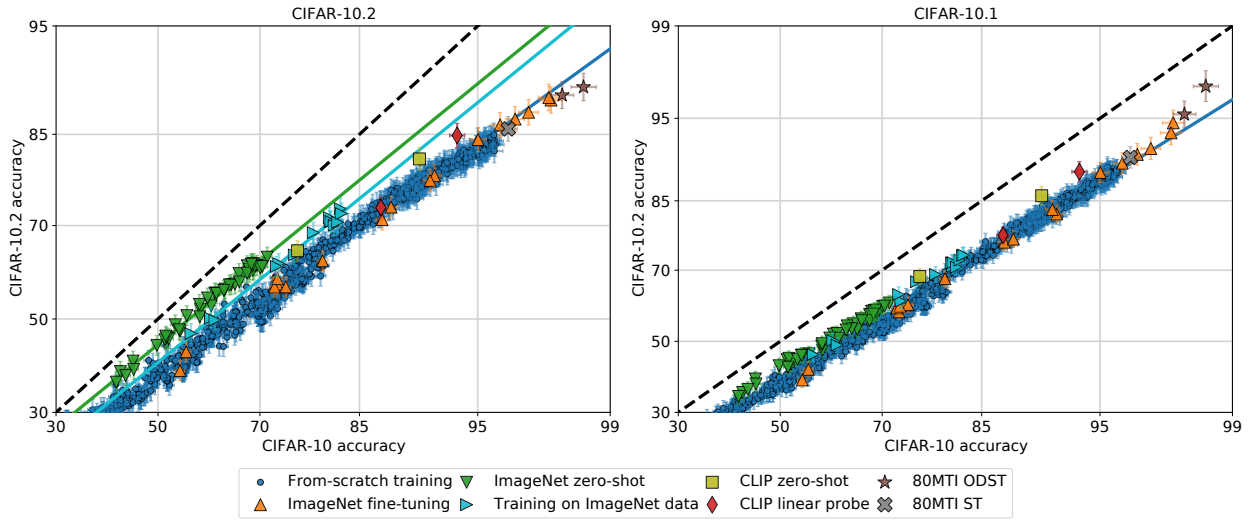


Figure 4.30: The effect of additional training data on OOD accuracy on CIFAR-10.2 (left) and CIFAR-10.1 (right).

the ID/OOD split is based on time). As the figure shows, the effect of fine-tuning pre-trained models is consistent across the four combinations: fine-tuning improves performance without deviating from the line. We also consider a linear probe of CLIP (see description in the previous subsection). Unlike the result on CIFAR-10, here the CLIP models do not significantly deviate from the linear trend. A possible explanation for this difference is that the web images on which CLIP was trend contain far more images of objects relevant for the CIFAR-10 classification task than they do for the FMoW-WILDS satellite image classification task.

Detailed findings for iWildCam-WILDS

Figure 4.32 shows the same models plotted in the iWildCam-WILDS panel of Figure 4.1, but separating the models trained from scratched and the fine-tuned models, and coloring points by the weight decay parameters. (For each weight decay we vary model architecture and learning rate). For fine-tuned models, there is a clear difference in the ID/OOD linear between model using weight decay 0 and models using nonzero weight decay. In particular, points with nonzero weight decay seem to lie above the zero weight decay linear trend. For models trained from scratch the macro F1 measurement error does not allow us to conclude with confidence whether weight decay affects the linear trend. Finally, it is worth noting that—even though increasing weight decay appears to move models above the zero weight decay line—the models with the best performance, both ID and OOD, do not use weight decay.

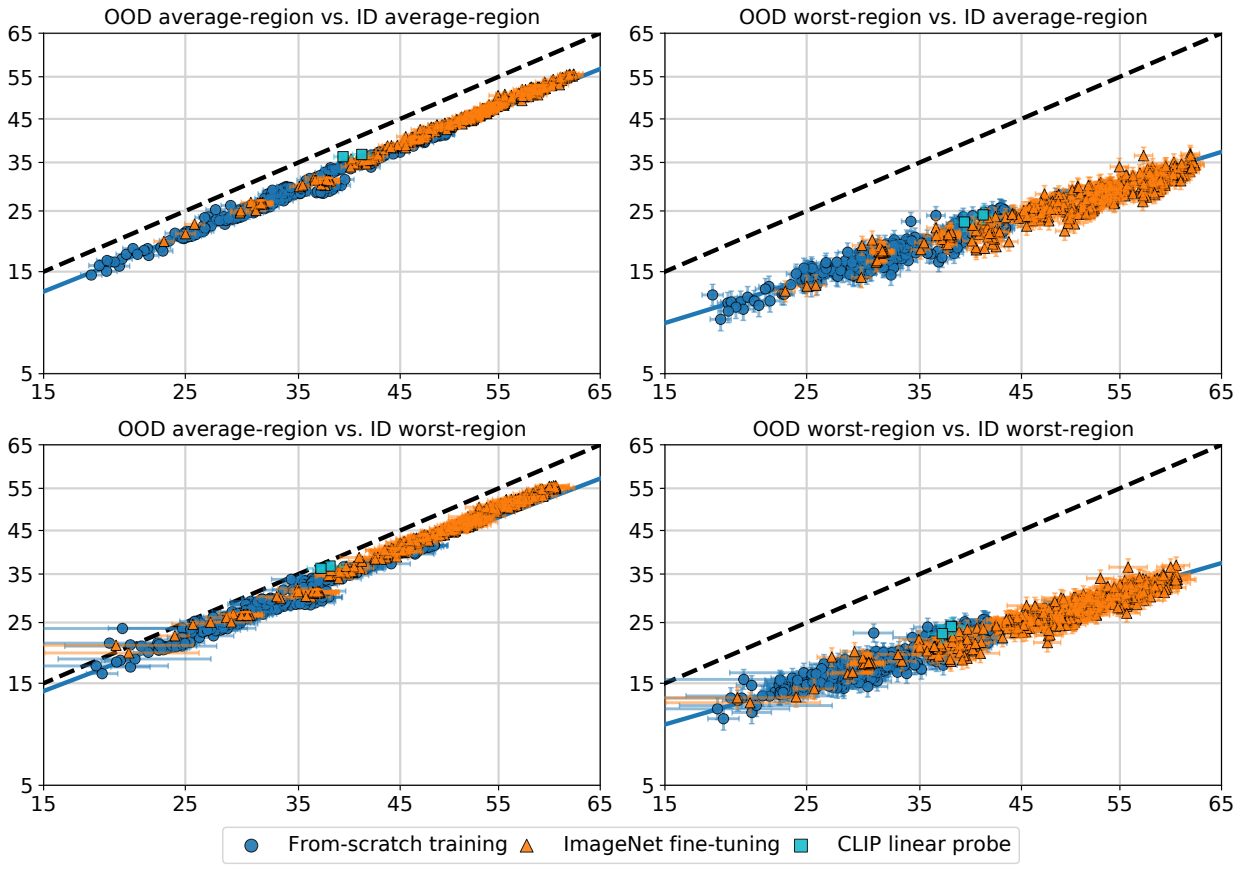


Figure 4.31: The effect of additional training and different accuracy metrics on FMoW-WILDS ID/OOD performance.

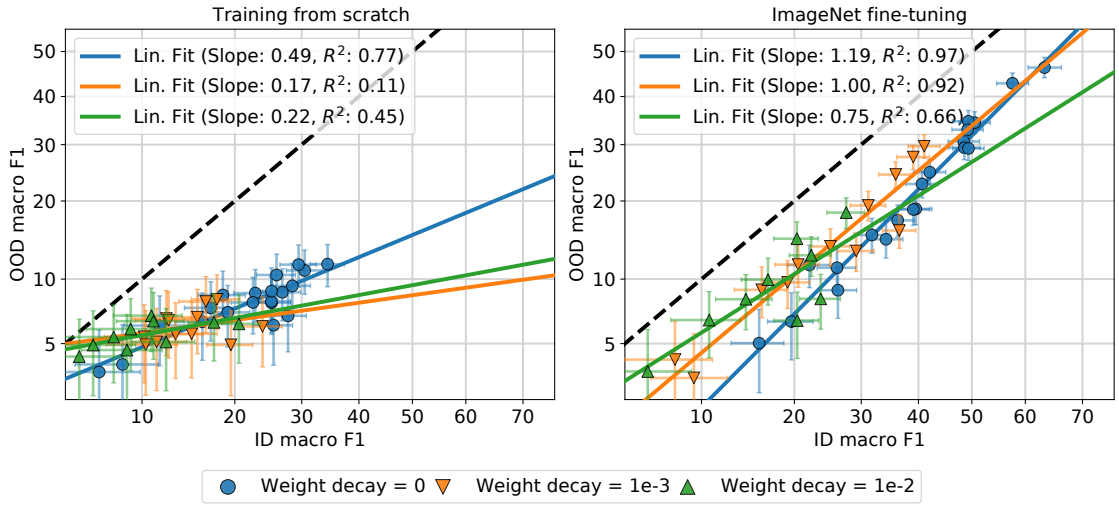


Figure 4.32: OOD vs. ID macro F1 scores for iWildCam-WILDS models trained from scratch (left) or fine-tuned from pretrained ImageNet models (right), with varying model architecture, learning rates, and weight decay. We observe that fine-tuned models exhibit a different linear trend than models trained from scratch, and moreover that the weight decay parameters affects the ID/OOD correlation, at least for fine-tuned models.

4.14 Appendix: Details on theoretical models for linear fits

Proof of Theorem 6

Proof. We begin by deriving expression for the accuracy of linear classifiers in our Gaussian distributional model. Under distribution D, the accuracy of linear classifier $\boldsymbol{\theta}$ is

$$\begin{aligned}\mathrm{acc}_D(\boldsymbol{\theta}) &= \Pr\big(\mathrm{sign}(\boldsymbol{\theta}^\top \boldsymbol{x}) = y\big) = \Pr\big(y \cdot \boldsymbol{\theta}^\top \boldsymbol{x} \geq 0\big) = \Pr\big(\mathcal{N}(\boldsymbol{\theta}^\top \boldsymbol{\mu}; \|\boldsymbol{\theta}\|^2\sigma^2) \geq 0\big) \\ &= \Pr\big(\|\boldsymbol{\theta}\|\sigma \cdot \mathcal{N}(0;1) \geq -\boldsymbol{\theta}^\top \boldsymbol{\mu}\big) = \Phi\left(\frac{\boldsymbol{\theta}^\top \boldsymbol{\mu}}{\|\boldsymbol{\theta}\|\sigma}\right),\end{aligned}$$

where we recall that $\Phi(t) = \int_{-t}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-s^2/2} \mathrm{d}s = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-s^2/2} \mathrm{d}s$ is the standard Normal cdf. Similarly, for the shifted distribution D' we have

$$\mathrm{acc}_{D'}(\boldsymbol{\theta}) = \Phi\left(\frac{\boldsymbol{\theta}^\top \boldsymbol{\mu}'}{\|\boldsymbol{\theta}\|\sigma'}\right) = \Phi\left(\frac{\alpha}{\gamma} \cdot \frac{\boldsymbol{\theta}^\top \boldsymbol{\mu}}{\|\boldsymbol{\theta}\|\sigma} + \frac{\beta}{\gamma\sigma} \cdot \frac{\boldsymbol{\theta}^\top \boldsymbol{\Delta}}{\|\boldsymbol{\theta}\|}.\right)$$

Therefore,

$$\left|\Phi^{-1}(\mathrm{acc}_{D'}(\boldsymbol{\theta})) - \frac{\alpha}{\gamma}\Phi^{-1}(\mathrm{acc}_D(\boldsymbol{\theta}))\right| = \frac{\beta}{\gamma\sigma}\left|(\boldsymbol{\theta}/\|\boldsymbol{\theta}\|)^\top \boldsymbol{\Delta}\right|. \tag{4.2}$$

Since $\boldsymbol{\theta}$ is independent of $\boldsymbol{\Delta}$ and $\boldsymbol{\theta}/\|\boldsymbol{\theta}\|$ is a unit vector, the inner product $(\boldsymbol{\theta}/\|\boldsymbol{\theta}\|)^\top \boldsymbol{\Delta}$ is distributed identically to the first coordinate of $\boldsymbol{\Delta}$. A standard concentration bound on the sphere [see, e.g., 6, Lemma 2.2] states that

$$\Pr(|\boldsymbol{\Delta}_1| > z) \leq 2e^{-dz^2/2}$$

for all $z \geq 0$. Substituting $z = \sqrt{2d^{-1}\log\frac{2}{\delta}}$ completes the proof. □

Remarks. We conclude this subsection with two additional remarks on the application of Theorem 6.

- **Classifiers trained on samples from D.** We note that any mapping of samples from D to a linear classifier results by definition in a classifier independent on $\boldsymbol{\Delta}$, and consequently Theorem 6 applies to it. In particular, it applies to the linear classifier we train in the simulation described in Figure 4.6.
- **A guarantee for multiple models.** Given N linear classifiers $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ such that each one is independent of $\boldsymbol{\Delta}$, we may apply Theorem 6 with probability parameter δ/N in conjunction with a union bound to conclude that, with probability at least $1-\delta$ we have $\left|\Phi^{-1}(\mathrm{acc}_{D'}(\boldsymbol{\theta}_i)) - \frac{\alpha}{\gamma}\Phi^{-1}(\mathrm{acc}_D(\boldsymbol{\theta}_i))\right| \leq \frac{\beta}{\gamma\sigma}\sqrt{\frac{2\log 2N/\delta}{d}}$ for all $i = 1, \ldots, N$. This precisely implies a linear trend in scatter plots such as Figure 4.6.

Departures from the linear trend

We now detail a number of modifications to our distribution model which break the linear trend predicted by Theorem 6 and shown in Figure 4.6. In each case, we mathematically define the modified model, provide intuition for why the linear trend no longer holds, and demonstrate the departure from the linear trend via simulation where train prediction models using samples from D and evaluate them on D'. We defer the full simulation details to the next subsection. The modifications we describe are not the only possible way to depart from the linear fit, but we focus on them because we believe they potentially represent departures from the trend seen in practice,

More data. Section 4.5, as well as prior work, show that using additional training data from a broader distribution can cause departure form the linear trend. To simulate such a scenario, we consider a third distribution D'' defined by $\boldsymbol{x}|y \sim \mathcal{N}(\boldsymbol{\mu}'' \cdot y; (\sigma'')^2 I)$, with $\boldsymbol{\mu}'' = \boldsymbol{\mu}' + \beta\tilde{\boldsymbol{\Delta}}$ and $\tilde{\boldsymbol{\Delta}}$ uniformly distributed on the unit sphere and independent of $\boldsymbol{\Delta}$. (Recall that $\boldsymbol{\mu}' = \alpha\boldsymbol{\mu} + \beta\boldsymbol{\Delta}$). Since D'' is more similar to D', we expect that including D'' samples in the training will result in better OOD performance. However, this inclusion could harm ID performance.

We demonstrate these effects in Figure 4.33 (left), where we train logistic regression models using samples from D and 0, 50 or 100 samples from D', with $\sigma'' = \sqrt{2}\sigma$ and all

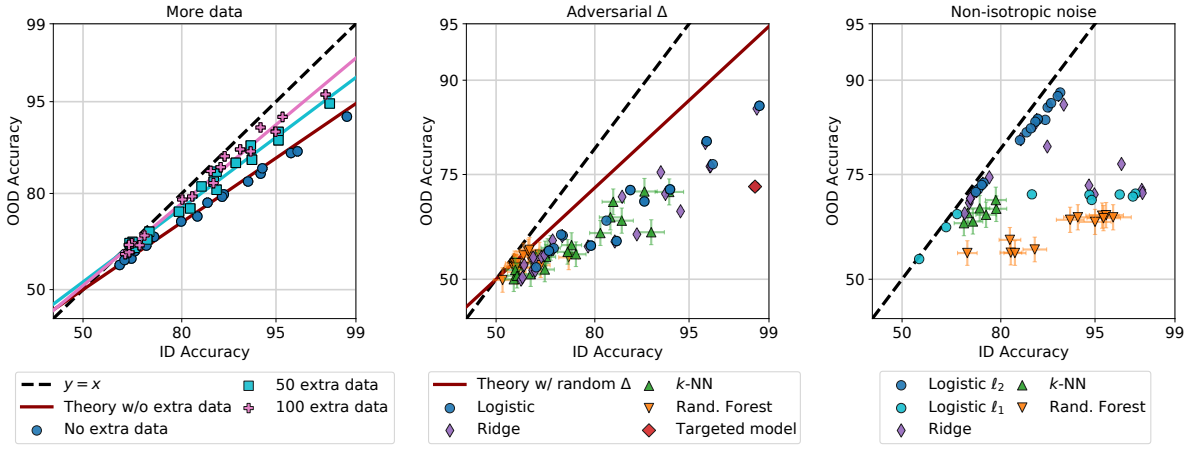


Figure 4.33: Modifications to our theoretical model showcasing departures from the linear trend. **Left:** training on auxiliary data related to D' (this plot only shows logistic regression models). **Middle:** choosing the parameter $\boldsymbol{\Delta}$ adversarially to reduce the OOD performance of a particular targeted model. **Right:** changing the noise covariance to be non-isotropic.

other parameters identical to the experiment shown in Figure 4.6. As expected, the extra data results in better OOD performance but worse ID performance. Moreover, the models trained on each amount of external training data appear to roughly follow linear trends (the plot shows empirical probit linear fits). However, we note that our theoretical analysis does not guarantee such linear fit, because the training data used to compute the classifier depends on $\boldsymbol{\Delta}$ through the samples from D''.

Adversarial distribution shift. As previously mentioned, the randomization of the distribution shift was crucial for our assumption, because if we allow $\boldsymbol{\Delta}$ to be a fix deterministic vector we cannot rule out that it depends adversarially on the trained classifier. Let us now spell out the implications of this possibility, by allowing $\boldsymbol{\Delta}$ to be any arbitrary vector of norm at most 1. Given some target classifier given a target classifier $\boldsymbol{\theta}$, suppose pick $\boldsymbol{\Delta} = c \cdot \boldsymbol{\theta}/\|\boldsymbol{\theta}\| = c \cdot \boldsymbol{\theta}/\|\boldsymbol{\theta}\|$ for some $c \in [-1, 1]$. This makes the inner product $\hat{\boldsymbol{\theta}}^\top \boldsymbol{\Delta} = c$. Recalling Eq. (4.2), this clearly implies a large departure from the linear trend when $|c|$ is close to 1. In particular, by picking negative c we may substantially reduce the performance of the model on D'. We note that this form of distribution shift is precisely the Gaussian model of adversarial examples proposed by Schmidt et al. [205].

In Figure 4.33 (middle), we demonstrate this technique by selecting one of the linear classifiers shown in Figure 4.6, call it $\boldsymbol{\theta}^\star$, and letting $\boldsymbol{\Delta} = -0.03\boldsymbol{\theta}^\star/\|\boldsymbol{\theta}^\star\|$. As the figure shows, the linear trend breaks substantially, particularly for the targeted classifier.

Non-isotropic covariance shifts. Finally, we consider the case where the noise covariance under D is not isotropic. That is, we let $\boldsymbol{x}|y \sim \mathcal{N}(\boldsymbol{\mu} \cdot y; \Sigma)$ for some Σ that is not a multiple of the identity. Instead of considering shifts to the mean $\boldsymbol{\mu}$, we consider random covariance shifts of the form

$$\Sigma' = \Sigma + (\sigma')^2 I_{d\times d},$$

i.e., simple additive white Gaussian noise with variance σ'. Under this distribution shift model, the probit accuracies are

$$\Phi^{-1}(\mathrm{acc}_D(\boldsymbol{\theta})) = \frac{\boldsymbol{\theta}^\top \boldsymbol{\mu}}{\sqrt{\boldsymbol{\theta}^\top \Sigma \boldsymbol{\theta}}} \quad \text{and} \quad \Phi^{-1}(\mathrm{acc}_{D'}(\boldsymbol{\theta})) = \frac{\boldsymbol{\theta}^\top \boldsymbol{\mu}}{\sqrt{\boldsymbol{\theta}^\top \Sigma' \boldsymbol{\theta}}}$$

For a the linear ID-OOD probit accuracy relationship to hold for all $\boldsymbol{\theta}$, we must have that $\Phi^{-1}(\mathrm{acc}_D(\boldsymbol{\theta}))/\Phi^{-1}(\mathrm{acc}_{D'}(\boldsymbol{\theta}))$ is a constant independent of $\boldsymbol{\theta}$, which happens if and only if

$$\frac{\boldsymbol{\theta}^\top \Sigma' \boldsymbol{\theta}}{\boldsymbol{\theta}^\top \Sigma \boldsymbol{\theta}} = 1 + \frac{\sigma'^2 \|\boldsymbol{\theta}\|^2}{\boldsymbol{\theta}^\top \Sigma \boldsymbol{\theta}}$$

is a constant independent of $\boldsymbol{\theta}$. However, this only holds when Σ is a multiple of the identity, contradictory to our assumptions. Indeed, whenever Σ is not a multiple of the identity, there could be a tradeoff between ID and OOD performance: the former favors $\boldsymbol{\theta}$ with small Σ-weighted norms, while the latter also depends on the standard Euclidean norm $\|\boldsymbol{\theta}\|^2$. Consequently, we expect regularization that limits $\|\boldsymbol{\theta}\|^2$ to provide better OOD performance.

Figure 4.33 (right) demonstrates this phenomenon via simulation. In the figure, we set Σ to be diagonal with a portion of the entries close to zero so that giving the corresponding coordinates of $\boldsymbol{\theta}$ larger weight results in better ID accuracy. For the distribution shift we let $(\sigma')^2 = \mathrm{tr}(\Sigma)/d$. As the figure shows, the linear trend no longer holds, despite the fact that the distribution shift is "only" adding isotropic Gaussian noise to the covariates. Moreover, as the above discussion predicts, the logistic and ridge regression models that attain strong OOD performance are those with stronger ℓ_2 regularization. We also show logistic regression trained classifiers with ℓ_1 regularization—these classifiers do not achieve good OOD performance.

Simulation details

Below, we provide additional details about the simulations described in Figures 4.6 and 4.33.

Training parameters. We fit logistic regression, ridge regression, nearest neighbors and random forest models using their scikit-learn implementations [172]. For logistic regression we use values of the inverse-regularization parameter C ranging from 10^{-6} to 1; we use ℓ_2 penalty throughout except the covariance shift experiment where we also consider ℓ_1 penalty. For ridge regression we use values of the regularization parameter α ranging from 10^{-3} to

10. For both types of linear models we do not fit an intercept. For nearest neighbors we use 1 or 3 nearest neighbors, and for random forests we use 3, 30 or 100 estimators. The remaining parameters are set to their scikit-learn defaults.

In addition to varying the learning hyperparameters described above, to produce models with varying accuracy we also modulate the training set size and dimensionality reduction. To reduce the training set size to size n_{sub}, we pick the first n_{sub} entries from a fixed training set generated once. To reduce dimensionality down to d_{proj}, we simply pick the first d_{proj} coordinates of x. For the all simulations except covariance shift, we let n_{sub} range between 30 and 100, and use d_{proj} in the range 50 to 3000. In the covariance shift simulation we use $n_{\text{sub}} \in [100, 2000]$ and fix $d_{\text{proj}} = d = 500$.

Accuracy measurement. For linear models we compute the accuracy exactly (see Subsection 4.14 for closed-form expressions). Consequently, we do not show error bars for these models. For the remaining models we estimate the accuracy on samples from the appropriate distributions and use error bars to show 95% Clopper-Pearson confidence intervals, consistently with the rest of the chapter.

Distribution model parameter setting. Throughout, we pick μ to be random unit vector (i.e., with the same distribution as $\mathbf{\Delta}$). For all simulations except covariance shift, we let $d = 10^5$, $\sigma = 10^{-1.5}$, $\alpha = 0.7$, $\beta = 0.5$ and $\gamma = 1$. For the covariance shift simulation, we found that using a smaller dimension and more training points lead to more noticeable effects. Therefore, for this simulation we let $d = 500$ (recall that α, β and γ do not exist in the covariance shift model). We let the covariance matrix Σ be diagonal, with 490 diagonal entries of size $1/2$ and the remainder of size $1/200$; the locations of the small entries were chosen at random. The shifted covariance is $\Sigma' = \Sigma + \frac{1}{8} I_{d \times d}$.

4.15 Appendix: Additional related work

We now summarize some of the additional work related to the phenomena we study in this chapter. Our focus here is mostly on recent work. For early work on distribution shift, we refer the reader to [180, 233].

PAC-Bayesian analysis of distribution shift. Performance under distribution shift has also been characterized under the PAC-Bayesian setting where the learning algorithm outputs a posterior distribution over the h hypothesis class [138, 85, 84]. Li and Bilmes [138] directly bound the error on the target distribution (OOD) in terms of the empirical error on a small number of labeled samples from the target and a "divergence prior" which measures some divergence between the source and target domains. Germain et al. [85] relate the OOD performance to the ID performance via a disagreement measure induced by the hypothesis class. These bounds do not explain the linear trends we find in this chapter—Li and Bilmes [138] do not relate the source and target error directly, and the bounds in Germain et al. [85]

are functionally similar to those of Ben-David et al. [17] where the ID performance is highly predictive of the OOD performance only if they are equal (Figure 1). Germain et al. [84] present a different analysis where the domain divergence appears as a *multiplicative* term rather than an additive one like in previous bounds. However, this bound expresses a linear relation between the OOD performance and some exponent of the "expected joint error" on the source domain which is different from the ID performance. Furthermore, the bound is an inequality which only provides an upper bound on the OOD performance, while our empirical results require a bound in the other direction as well.

Reliability of machine learning benchmarks. When assessing the reproducibility and reliability of statistical findings, Yu [254] advocates for considerations of *stability* of statistical results under perturbations of the model and data. Key to this account is a notion of a stability target that is estimated under changes to the data used to estimate the target. Viewed in this light, our experimental methodology of training models on one distribution and evaluating them on a collection of out-of-distribution test sets corresponds to testing the stability of an appropriately chosen target under perturbations to the test set. The consistent accuracy drops we observe across out-of-distribution test sets suggests that model accuracies are not stable; however, the precise linear trends we find suggest that *model rankings* are in fact stable. As a consequence, benchmarks like CIFAR-10 or ImageNet where we observe precise linear trends may provide more reliable knowledge about relative model performance than absolute performance on new out-of-distribution test sets.

Theoretical models for linear trends in earlier work on dataset reproduction. Both Recht et al. [189] and Recht et al. [190] contain simple models for the linear fits observed in their reproductions of CIFAR-10 and ImageNet. Recht et al. [189] propose a mixture model with an "easy" and "hard" component and model the distribution shift as a change in the weigts of these two components. Their model does indeed give a linear fit, but only with linear axis scaling. As we have seen several times throughout this chapter, the scatter plots show cleaner linear trends with logit or probit scaling on the axes. It is also not clear what the "easy" and "hard" components correspond to in distribution shifts such as CIFAR-10.1.

Recht et al. [190] developed their model further. Instead of discrete mixture components, each distribution is now parametrized by a Gaussian distribution over the "hardness" of each image. In addition, every model has a scalar "skill" parameter that determines the probability of a model classifying an image with a given hardness correctly. This model now produces linear fits in the probit domain, which yields a closer fit to empirical results. While a continuous hardness parametrization also is more plausible, it is again unclear what this hardness corresponds to.

Neither the models of Recht et al. [189, 190] nor our model of Section 4.7 allow us to predict where linear trends occur in actual data; such predictive power is important because—as we demonstrate—some distributions do not yield linear trends. However, our

theoretical analysis is based on a concrete generative is based on a concrete generative, rather than postulated abstract properties of data and classifiers. One advantage of this fact is that it allows us to consider modifications of our generative models which show departures from the linear trend, as we do in Appendix 4.14.

Linear trends in image classification with natural language supervision. Among other results, Radford et al. [182] show two important phenomena that are closely related to this chapter. First, their training approach (contrastive language image pre-training, "CLIP"), which combines a large training set and natural language supervision, produces image classifiers substantially above the linear trend given by a wide range of ImageNet model in the distribution shift testbed of Taori et al. [228]. This result provides further evidence for the hypothesis that training data plays an important role in the linear trends we describe in this chapter. Second, Radford et al. [182] find that once their training set is fixed and they vary model architecture (ResNet variants and Vision Transformers [70]) and compute available for training, the resulting models again follow a clear linear trend. This demonstrates that linear trends between in-distribution and out-of-distribution accuracy occur in a diverse range of settings.

Linear trends under sub-population shift. One specific type of distribution shift is *sub-population* shift. In sub-population shift, each class is composed of a set of sub-populations, e.g., the "dog" class in an image classification task may be composed of images from a specific set of dog breeds. A natural goal then is that a trained classifier should generalize to previously unseen dog breeds and still correctly labels them as "dog". Hendrycks and Dietterich [98] found that a set of eight convolutional neural networks follow a linear trend on a sub-population shift derived from ImageNet-22K. Santurkar, Tsipras, and Madry [203] construct a range of sub-population shifts from ImageNet and find approximately linear trends for several of the shifts they consider. Their testbed contained 13 convolutional neural networks, some of them with interventions such as adversarial training [150]. Some of the plots in [203] are not directly comparable to ours since they display a relative accuracy measure on the y-axes, not the absolute accuracy (i.e., average 0-1 loss).

Underspecification as defined in D'Amour et al. [59]. D'Amour et al. [59] conduct a broad empirical study and show that out-of-distribution performance can vary widely even for models with the same in-distribution performance. Since this result may at first glance disagree with our results here, we now discuss their empirical results most relevant to this chapter in detail. In particular, we focus on their results in computer vision domains.

- D'Amour et al. [59] point out ImageNet-C as an example of underspecification in image classification. Similar to [228], we also find in Figure 4.24 that some of the perturbations in ImageNet-C show substantial variation as a function of ImageNet accuracy. In addition, we find that this variation occurs in CIFAR-10-C. As mentioned

before, not all shifts in ImageNet-C and CIFAR-10-C are affected by underspecification, with some shifts exhibiting comparatively clean linear trends.

- The second example for underspecification in image classification is ObjectNet [9]. While it is indeed true that the accuracy variation on ObjectNet may increase compared to ImageNet, overall ObjectNet still shows predictable behavior as a function of ImageNet accuracy. See Figure 2 in [228].

- In addition to standard computer vision benchmarks, D'Amour et al. [59] also investigate two medical imaging datasets, which give an important complementary perspective. In the first dataset (ophthalmological imaging), they find evidence of underspecification. In the second dataset (dermatological imaging), the evidence is less clear since the tests for statistically significant variation in the four domains give p-values of 0.54, 0.42, 0.29, and 0.03. While the fourth p-value is below 0.05, the authors did not correct for multiple hypothesis testing and remark that this is an exploratory data analysis.

Overall, we find that the empirical evidence for underspecification in computer vision tasks is nuanced. As in our work, some distribution shifts studied by D'Amour et al. [59] exhibit stronger correlation between in-distribution and out-of-distribution than others. Hence there is no clear contradiction between our results and those of D'Amour et al. [59]. Understanding when precise linear trends occur and when underspecification is dominant is an important direction for future work.

Further distribution shifts without universal linear trends. While we have seen several distribution shifts with clean linear trends between in-distribution and out-of-distribution generalization in this chapter, there are also obvious counterexamples. One prominent counterexample are adversarial distribution shifts, e.g., ℓ_p adversarial examples [21, 224, 20]. For models trained without a robustness intervention, it is usually easy to construct adversarial examples that cause the model to misclassify most inputs despite high accuracy on unperturbed examples. While adversarial robustness is far from solved, it is now possible to train CIFAR-10 networks with about 65% accuracy against the common ℓ_∞ adversary with $\varepsilon = 8/255$ and standard (unperturbed) accuracy of 91% [86]. Since CIFAR-10 classifiers without a robustness intervention have only 0–10% robust accuracy in this setting, it is clear that there cannot be a precise linear trend between in-distribution and out-of-distribution accuracy. Adversarial distribution shifts can bring about departures from the linear trend in our theoretical setup as well, as we discuss in Appendix 4.14. We refer the reader to Taori et al. [228] and Hendrycks et al. [100] for additional examples of models not following a linear trend in ImageNet variants, e.g., on some of the ImageNet-C corruptions and ImageNet-R.

Benchmarks for distribution shift. Recently several groups conducted broad empirical surveys of distribution shift, comparing a wide range of available methods. Most closely

related to this chapter is Taori et al. [228], where the authors also find clean linear trends on multiple distribution shifts related to ImageNet. Djolonga et al. [67] also observed high correlations on the same distribution shifts for a smaller number of models. Both experiments were limited to ImageNet as in-distribution test set and convolutional neural networks. Here we study multiple different in-distribution datasets for image classification, an additional task (pose estimation), and several models beyond convolutional neural networks.

Gulrajani and Lopez-Paz [89] conduct a broad survey of algorithms for the closely related problem of domain generalization. In domain generalization, the training set is drawn from multiple distinct domains, and the learning algorithm has access to the domain labels. At test time, the trained models is evaluated on samples from a new domain. Gulrajani and Lopez-Paz [89] found that on a range of datasets, current domain generalization algorithms perform only as well or worse as an empirical risk minimization baseline that ignores the domain structure. At a high level, this result is similar to the aforementioned distribution shift benchmarks that also found small or no gains from current robustness interventions on most distribution shifts.

Our results raise similar questions as these benchmarks for distribution shift and domain generalization: when and how is it possible to improve over empirical risk minimization as a baseline for robustness to distribution shift, i.e., to "go above the line" in our scatter plots?

Training methods to improve robustness. Researchers have proposed a large number of robustness interventions over the past few years. Due to the volume of papers, we only refer to recent surveys here. Methods for improving robustness divide into two categories: those which use samples from the target distribution (which we refer to as the OOD data), and those that do not. The former methods are usually called transfer learning and domain adaptation methods [165, 241]. These methods typically assume that the target distribution data is more constrained than the in-distribution data, either lacking labels or having smaller quantity, and algorithms focus on mitigating these issues. The linear trends observed by Kornblith, Shlens, and Le [122] in the context of transfer learning suggest that there may be important similarities. See Robey, Pappas, and Hassani [194] for one example applying such techniques to the WILDS OOD shifts considered in this work.

While domain adaptation and transfer learning techniques are helpful in many settings, they are not always applicable. For instance, when we want an autonomous vehicle to drive safely in a new town it has not visited before, we have no additional training data available to adapt the car's perception system. Such scenarios motivate our study of the correlation between in-distribution and out-of-distribution generalization in this chapter. The second category of training methods—sometimes referred to as domain robustness or domain generalization—attempt to learn models that are reliable in the presence of distribution shifts for which there is no direct training data. Instead, these methods often leverage data from multiple other, related domains. Gulrajani and Lopez-Paz [89] provide an overview of current methods for domain generalization.

Chapter 5

Retiring Adult: New Datasets for Fair Machine Learning

5.1 Introduction

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 urging better data practices [35, 83, 114, 162, 167]. While the previous chapters have thus far 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 [166, 74, 18].

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 [121]. 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.

We begin this chapter 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 results differ significantly. Recognition of the need for such a choice is still lacking, as

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

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.

5.2 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 [91]. 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 [130]. 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. [37] 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. [174] 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. [257] 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 [22], 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 5.2.

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 [79] 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 5.6. 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.

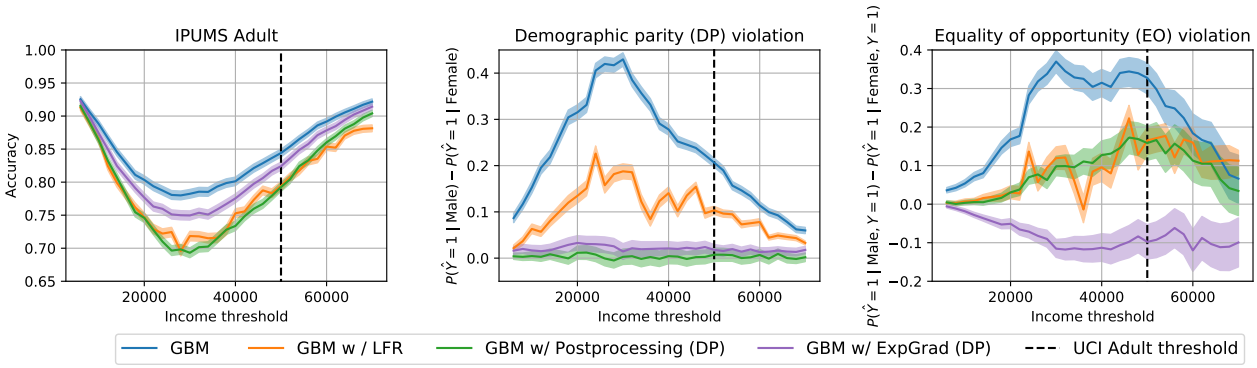


Figure 5.1: Fairness interventions with varying income threshold on IPUMS Adult. We compare three methods for achieving demographic parity: a pre-processing method (LFR), an in-training method based on Agarwal et al. [2] (ExpGrad), and a post-processing adjustment method [90]. We apply each method using 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, Hardt, and Narayanan [10] for background. The results are presented in Figure 5.1, 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 [257] mentioned earlier, an *in-processing* or *in-training* method based on the reductions approach in Agarwal et al. [2], and the post-processing method from Hardt, Price, and Srebro [90]. In Figure 5.1, 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 5.6, 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.

5.3 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 [233, 27, 38, 250], often lacking copyright [135] or consent from subjects [177], and involving unskilled or poorly compensated labor in the form of crowd workers [87].

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 5.1: 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.

- **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 [11]. 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 5.1 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.

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 chapter.

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 [8].

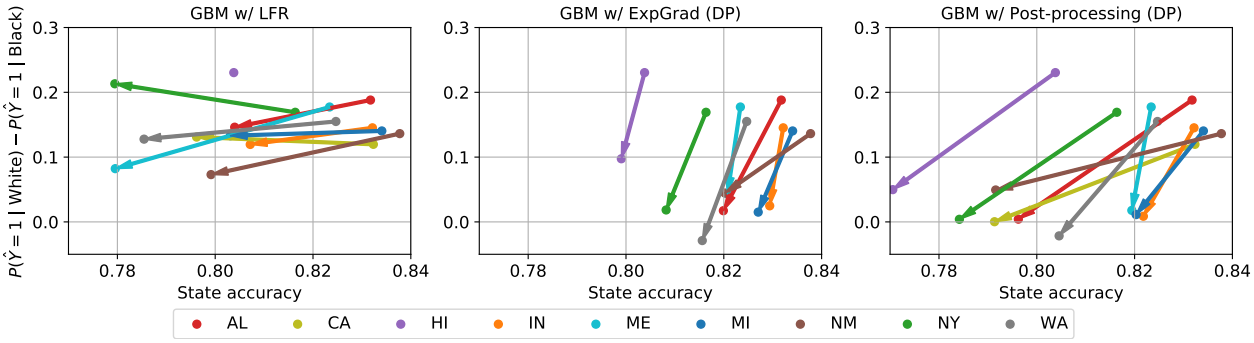


Figure 5.2: The effect size of fairness interventions varies by state. Each panel shows the change in accuracy and demographic parity on the ACSIncome task after applying a fairness intervention to an unconstrained gradient boosted decision tree (GBM). Each arrow corresponds to a different state distribution. 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.

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

5.4 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.

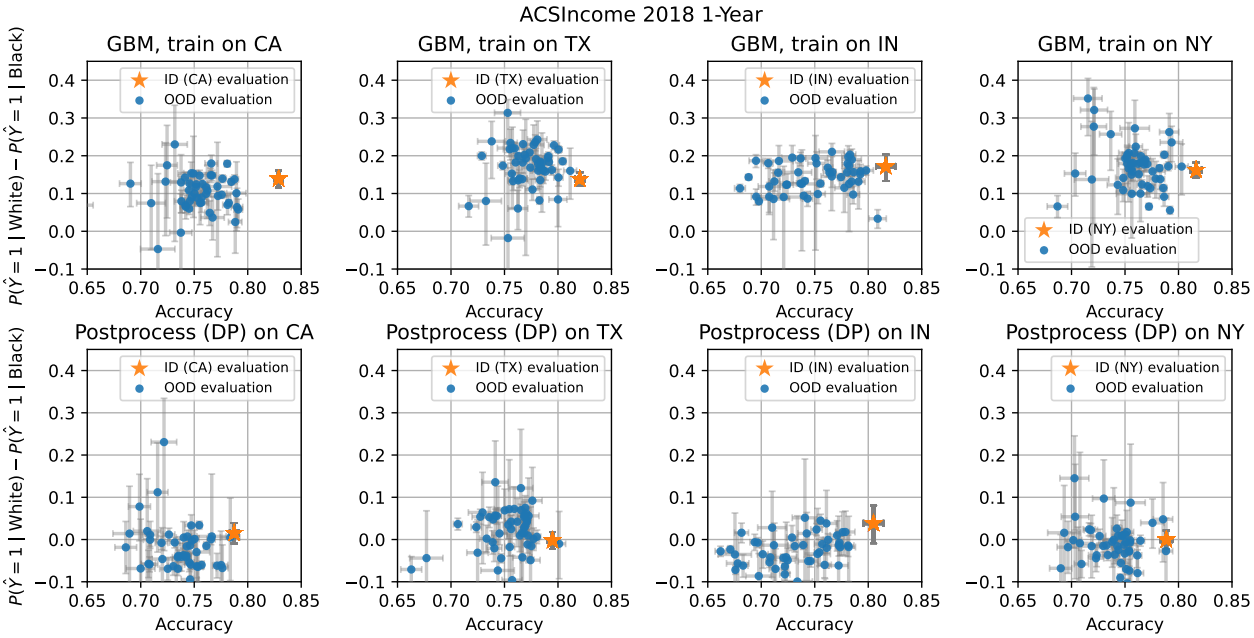


Figure 5.3: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSIncome task and evaluated both in-distribution (ID) on the same state and out-of-distribution (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.

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 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 5.8.

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 eval-

uate 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. [2] (ExpGrad), and the simple post-processing method that adjusts group-based acceptance thresholds to satisfy a constraint [90]. Figure 5.2 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 5.3, 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 5.4, 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.

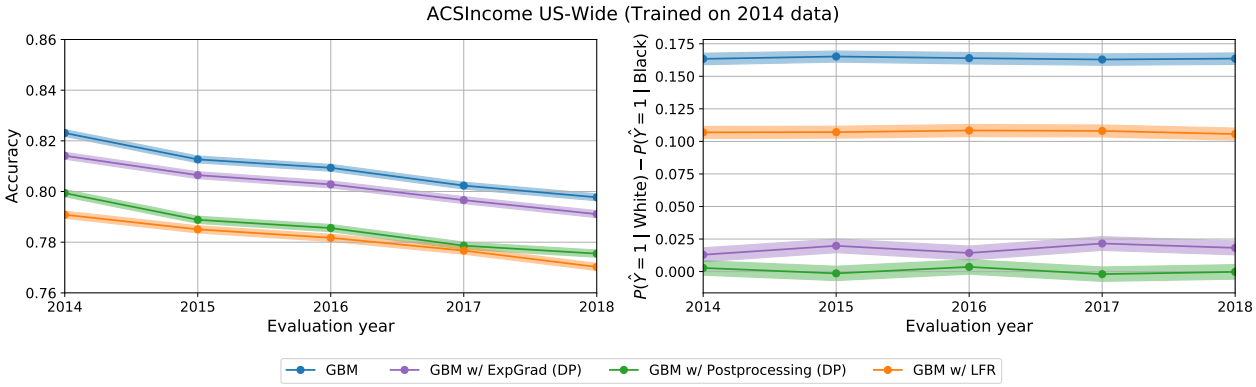


Figure 5.4: Fairness criteria are more stable over time than accuracy. **Left:** Models trained in 2014 on US-wide ACSIncome with and without fairness interventions to achieve demographic parity and evaluated on data in subsequent years suffer a drop in accuracy over time. **Right:** However, the violation of demographic parity remains essentially constant over time. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for demographic parity.

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 5.2, 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 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).

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 [35], 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 [178]. However, on our tabular datasets, larger datasets collected in more socially progressive times do not automatically mitigate disparities. Table 5.3 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

Table 5.2: 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+LFR	78.7 ± 0.1 %	16.6 ± 0.2%	79.4 ± 0.1%	14.0 ± 0.2%
GBM+post-processing (DP)	79.2 ± 0.1 %	0.3 ± 0.3 %	80.2 ± 0.1%	-0.6 ± 0.3%

Table 5.3: 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%

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 [49], 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.

5.5 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 [90]) 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* [4] perform on different geographic contexts?

5.6 Appendix: Additional details about adult reconstruction

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. [173] 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 [257] 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. [2] provided by Bird et al. [22] with the default hyperparameters, and we used implementation of post-processing [90] provided by Bellamy et al. [15].

In Figure 5.1 in the main text, we compare the performance of these three fairness interventions when enforcing demographic parity as the threshold varies. In Figure 5.5, 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.

Table 5.1 additional details

For each of the tasks listed in Table 5.1 (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. `Constant 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. [173] with the default hyperparameters.

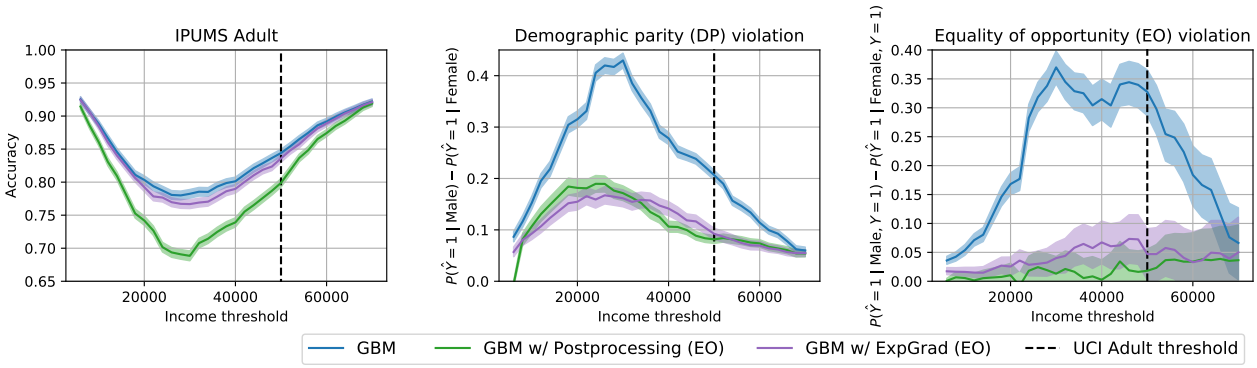


Figure 5.5: Fairness interventions with varying income threshold on IPUMS Adult. Comparison of in-processing and post-processing methods for achieving equality of opportunity (EO). LFR does not target EO, so we exclude it from the comparison. Confidence intervals are 95% Clopper-Pearson intervals for accuracy and 95% Newcombe intervals for equality of opportunity.

5.7 Appendix: Omitted 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 chapter. We use the implementation provided by Pedregosa et al. [173] and use `exponential` 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 [257] 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. [2], we use the implementation from Bird et al. [22] with the default hyperparameters, and for the post-processing method, we use the threshold adjustment method of Hardt, Price, and Srebro [90], which is also implemented in Bellamy et al. [15]. In Section 5.4, we use all of the methods to enforce demographic parity. We detail additional experiments enforcing equality of opportunity in Appendix 5.8.

Datasets. Throughout this section, we use the ACSIncome task described in Section 5.3. 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.

5.8 Appendix: Additional experiments using folktables

In this section, we conduct the same set of experiments conducted in Section 5.4 on the 5 other prediction tasks we introduced in Section 5.3. Throughout we keep the experimental details (models, hyperparameters, etc) identical to those detailed in Appendix 5.7.

Intervention effect sizes across states

As in Section 5.4, 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. [2] (ExpGrad), and the simple post-processing method that adjusts group-based acceptance thresholds to satisfy a constraint [90]. Figure 5.6 shows the result of this experiment for the ACSIncome prediction task for interventions to achieve equality of opportunity.

In Figure 5.7, 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 5.8, 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 state test set) and the out-of-distribution test sets for the 49 other states. This is the same

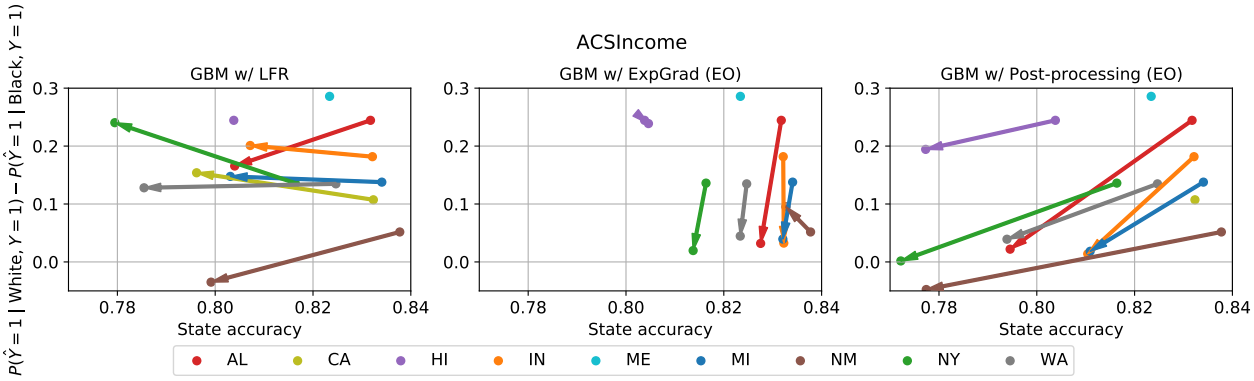


Figure 5.6: The effect size of fairness interventions varies by state. Each panel shows the change in accuracy and equality of opportunity violation (EO) on the ACSIncome task after applying a fairness intervention to an unconstrained gradient boosted decision tree (GBM). Each arrow corresponds to a 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.

experiment as in Section 5.4, but with equality of opportunity rather than demographic parity as the metric of interest. In Figures 5.9, 5.10 5.11, and 5.12 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 5.13, 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 5.4; however, here we consider interventions to satisfy equality of opportunity rather than demographic parity. In Figure 5.14, we conduct repeat this experiment for interventions to satisfy demographic parity on 4 other ACS PUMS predictions tasks: ACSPublicCoverage, ACSMobility, ACSEmployment, and ACSTravelTime.

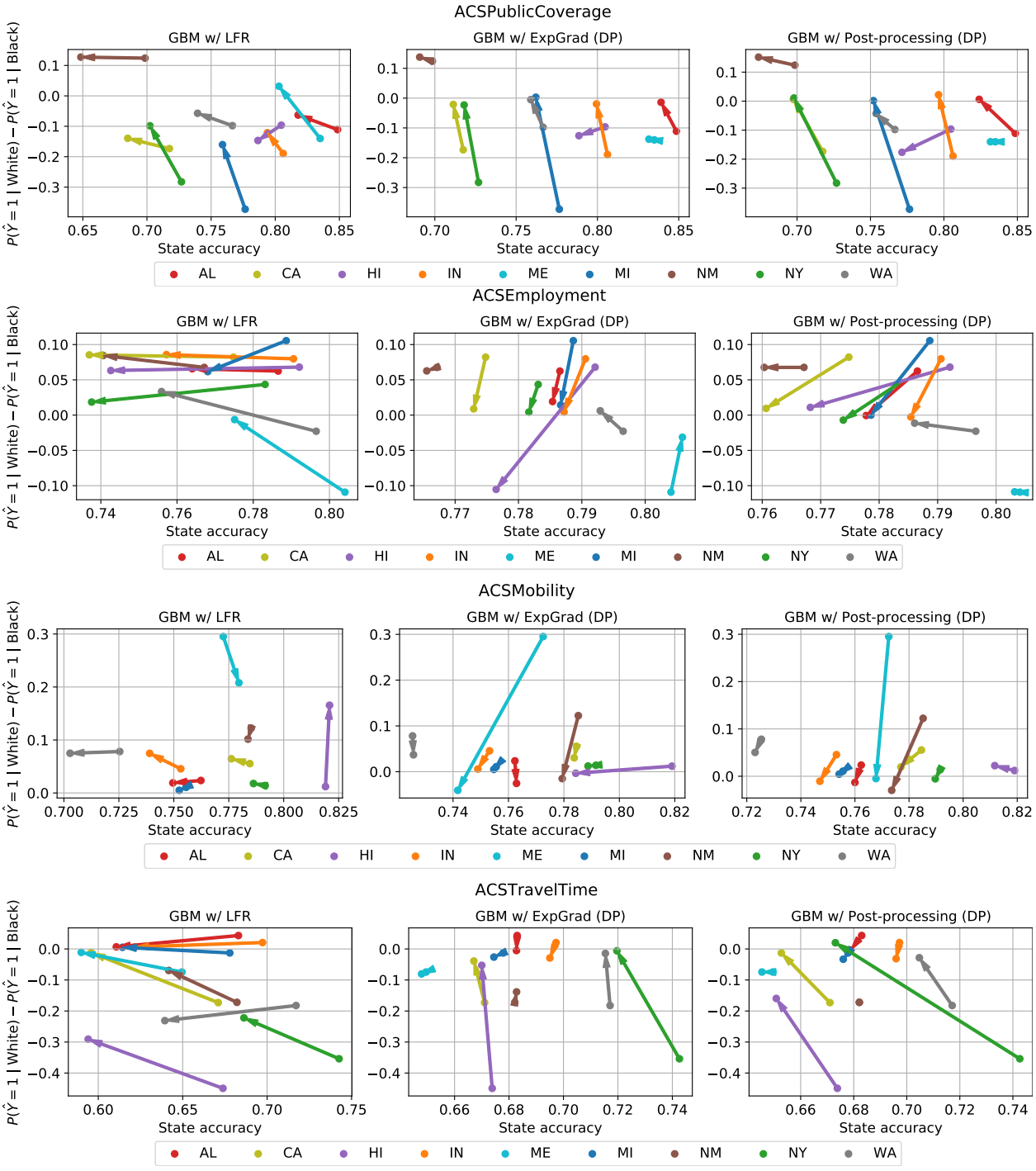


Figure 5.7: The effect size of fairness interventions varies by state. Each panel shows the change in accuracy and demographic parity violation (DP) on the ACSIncome task after applying a fairness intervention to an unconstrained gradient boosted decision tree (GBM). Each arrow corresponds to a different 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.

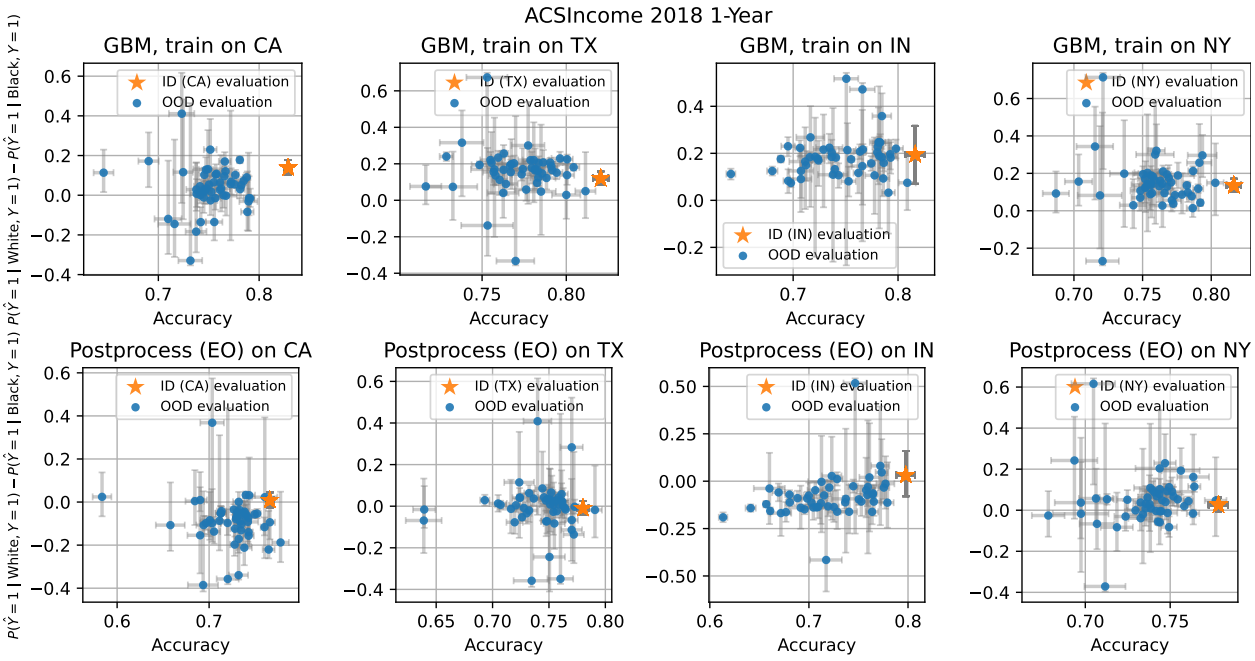


Figure 5.8: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSIncome task and evaluated both in-distribution (ID) on the same state and out-of-distribution (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.

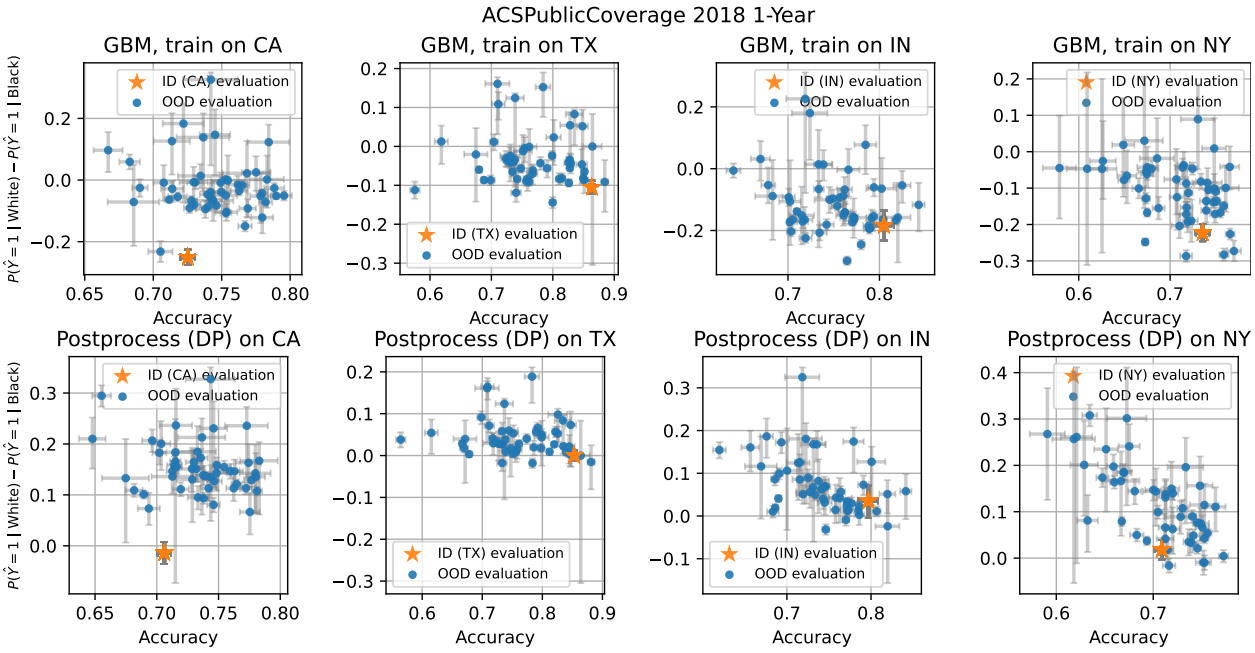


Figure 5.9: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSPublicCoverage task and evaluated both in-distribution (ID) on the same state and out-of-distribution (OOD) on the 49 other states in terms of accuracy and 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.

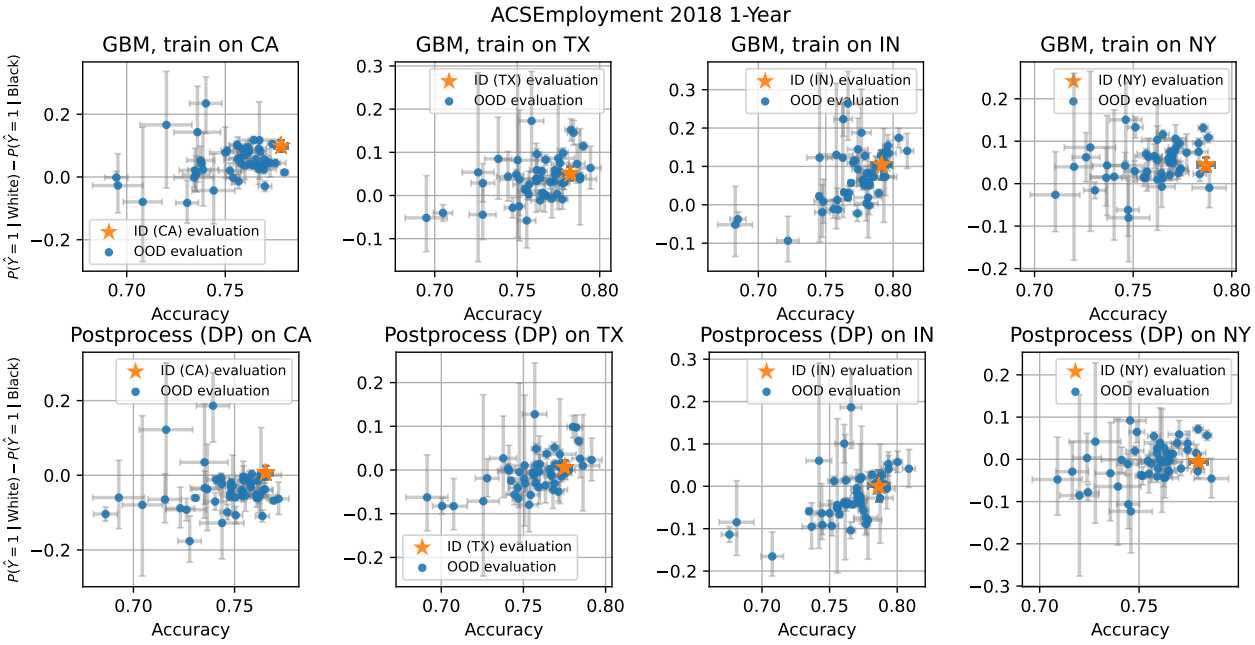


Figure 5.10: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSEmployment task and evaluated both in-distribution (ID) on the same state and out-of-distribution (OOD) on the 49 other states in terms of accuracy and 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.

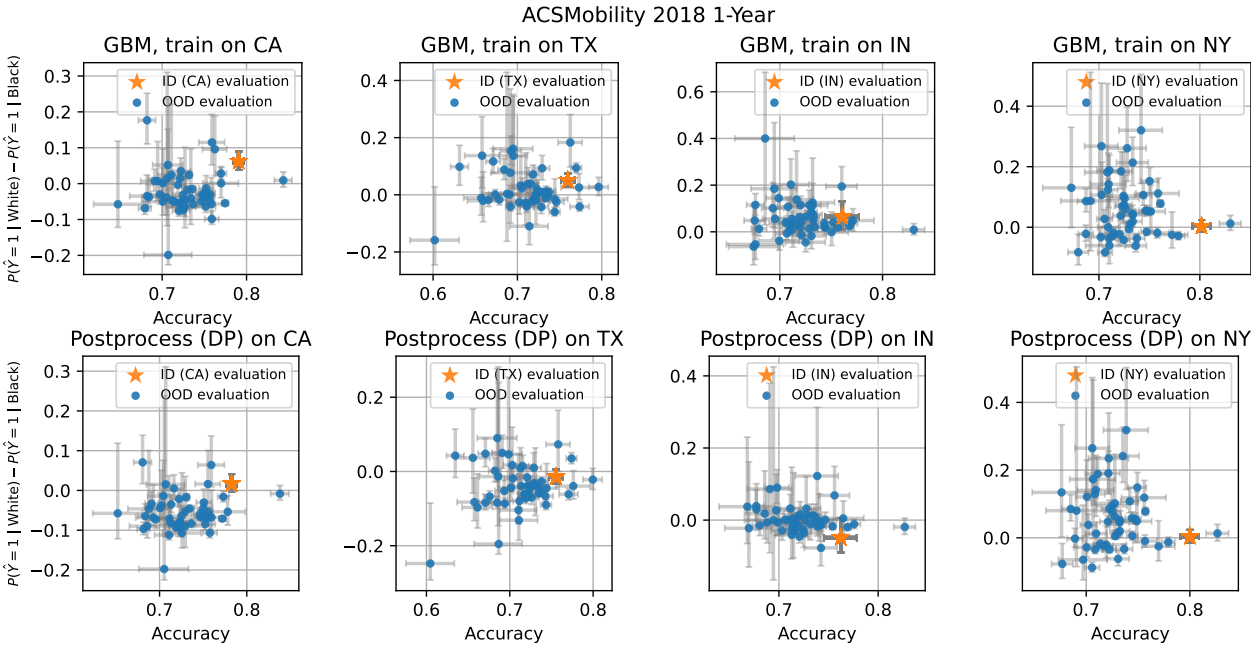


Figure 5.11: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSMobility task and evaluated both in-distribution (ID) on the same state and out-of-distribution (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.

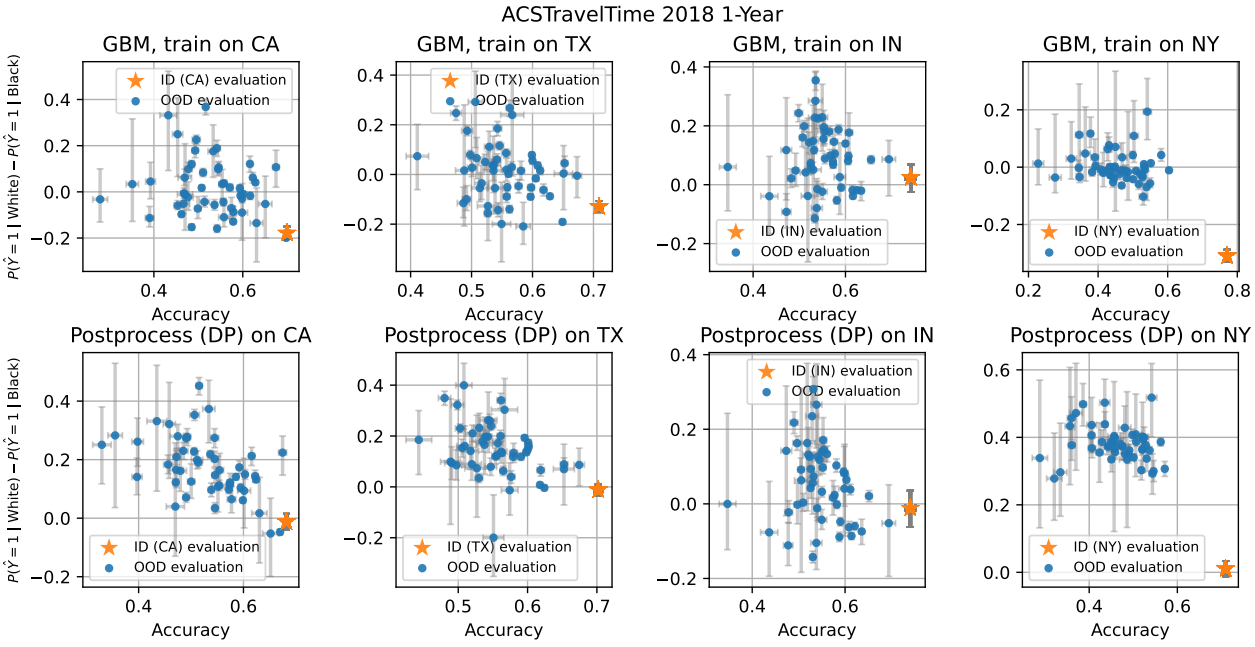


Figure 5.12: Transfer from one state to another gives unpredictable results in terms of predictive accuracy and fairness criteria. **Top:** Each panel shows an unconstrained GBM trained on a particular state on the ACSTravelTime task and evaluated both in-distribution (ID) on the same state and out-of-distribution (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.

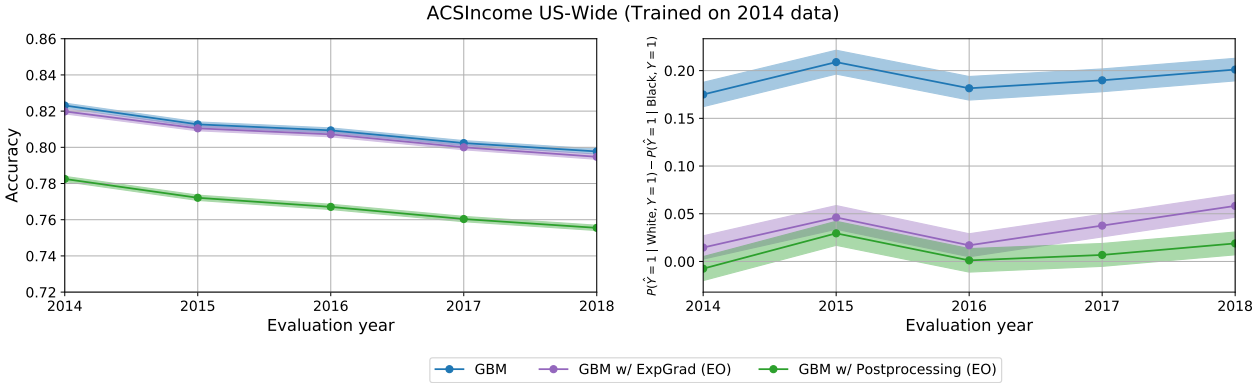


Figure 5.13: Fairness criteria are more stable over time than accuracy. **Left:** Models trained in 2014 on US-wide ACSIncome with and without fairness interventions to achieve equality of opportunity and evaluated on data in subsequent years. **Right:** Violations of equality of opportunity for the 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.

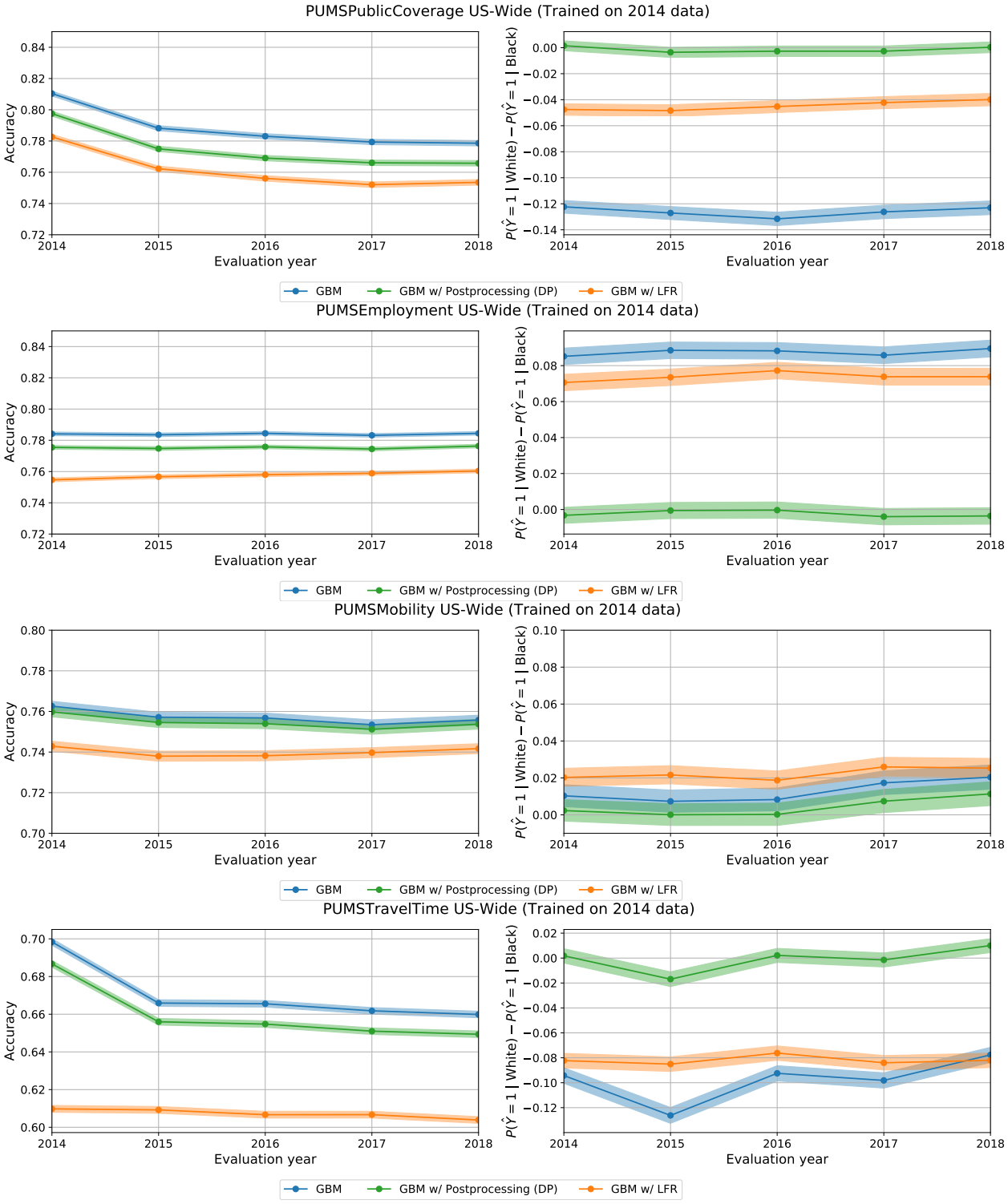


Figure 5.14: Fairness criteria are more stable over time than accuracy. **Left:** Models trained in 2014 on US-wide ACS data with and without fairness interventions to achieve demographic parity and evaluated on data in subsequent years. **Right:** Violations of demographic parity for the same collection of models. Although accuracy drops over time for most problems, violations of demographic parity remain essentially constant.

Chapter 6

WhyNot: Simulation Benchmarks for Causal Inference

6.1 Introduction

Does additional low income housing cause an increase in urban crime? Causal questions such as these have long been a subject of active research. Today, researchers increasingly tackle such questions with the growing technical repertoire of statistical causal inference from observational and experimental data. High stakes interventions in domains including climate, development, education, health and finance often start with causal reasoning. Consequential policy-making on the basis of formal causal analysis puts pressure on the validity and the robustness of the analysis. The previous chapters have highlighted validity challenges associated with machine learning benchmarks for predictive tasks. The validity of causal inference is even more difficult question because it is a matter of formal assumptions that are often subtle and generally untestable [170, 101].

Causal inference has not been the only route by which scientists have attempted to anticipate the effect of interventions. Nearly as old as the digital computer is the idea of computer simulation of mathematical models designed to capture aspects of the real world. Dating back more than half a century, a field known as system dynamics sought to derive policy recommendations from computer simulation of dynamical systems in a range of domains including industrial management, urban planning, and climate forecasting [80, 160]. Similarly, agent-based modeling, an active research area with applications ranging from economics to epidemiology, rests on the idea of simulating the collective interactions of many agents [28, 82]. Robotics, control theory, and reinforcement learning all actively deploy simulated environments in order to improve systems and anticipate their failure points [231, 151]. As compute resources continue to grow, simulation is an increasingly powerful tool.

In this chapter, we connect computer simulation and causal inference by using the former as a testbed for the latter. Simulation inevitably shows discrepancies with the real world. But even when a simulator fails to faithfully represent the real world, it can nonetheless

provide a valuable environment in which causal questions have non-trivial answers. Ground truth is available through simulation where formal analysis is impenetrable. Each simulator we consider naturally leads to a range of causal questions by varying which variables to include, what time horizons to consider, and how to generate data from the simulator. With this powerful tool at hand, we begin to investigate how today's practice of causal inference fares in challenging simulated environments. Or, to speak by analogy, we ask: *Would causal inference understand Sim City?*

Our contributions. We designed and implemented an extensible, easy-to-use system, called *WhyNot*, that provides several complex simulators, numerous experimental designs, and a representative suite of state-of-the-art causal inference methods.

Underlying our framework is a novel methodology of how complex dynamical systems can elucidate questions of robustness and validity in causal inference. We start from dynamical systems that have been proposed over the years independently of statistical causal inference. Therefore, these simulated environments do not encode predetermined causal questions or answers. Practitioners and researchers may instead use their subject matter knowledge in combination with our framework to design tasks that probe what they believe is relevant to the success of their method in practice.

To showcase our system, we design several experiments aimed at key issues in causal inference. Specifically, we focus on analyzing the impact of observed and unobserved confounding, mediation analysis, and interference between experimental units on the validity of inference procedures. In each case, we let a key assumption fail in a controlled manner to see how a set of methods responds to the failure of the assumption. Our experiments surface a new empirical understanding of how today's causal inference methods perform in such cases. Although we can necessarily only highlight a few representative experiments, our framework is by no means limited to these. Additional simulators, experiments, and methods are straightforward to add as our framework is agnostic to any particular modeling language. At the outset, *WhyNot* is an experimental platform for creating challenging test cases, rather than a set of fixed benchmarks.

Related work. The current practice of simulation captures only a small fraction of real world phenomena. Though there have been many efforts to use synthetic and simulated data sets to test statistical methods, these often study idealized conditions where all of the assumptions for inference are satisfied, and deviations from these settings, if they are studied at all, rely on simple parametric models [23, 69, 112, 110, 118]. *WhyNot* expands the kinds of environments in which we can robustly evaluate causal inference.

Most closely related to our work is the simcausal R package [217], which provides a tool for generating synthetic data based on structural equation models, requiring the user to provide functional relationships governing the data-generating process. Our framework takes a different philosophical and technical approach. We employ existing dynamical systems as

the simulated environments, and the functional relationships are not explicitly user specified but rather determined by the dynamics.

Recently, there have also been efforts to rigorously compare causal inference methods via data analysis competitions and simulation benchmarks on quasi-synthetic datasets, notably those by the Atlantic Causal Inference Conference (ACIC) [69, 118]. Hernán [102] criticizes competitions such as these for lacking settings that "violate the key assumptions of ignorability and overlap" as well as cases "that include failure time outcomes and time-varying treatments and covariates." We see our framework as a way of addressing many of the concerns that Hernán raises.

6.2 Causal inference through the lens of dynamical systems

Dynamical systems naturally create interesting causal inference problems involving *confounding* and *mediation*. The connection between dynamical systems and causal inference is central to our work and rests on a methodological and philosophical perspective that we now develop.

Models and bounding boxes

Any causal analysis implicitly draws a "bounding box" around a set of variables included in the study. The bounding box separates the variables from their possibly chaotic surroundings in the universe. Pearl summarizes this philosophical point crisply [170, p. 420]:

> If you wish to include the entire universe in the model, causality disappears because interventions disappear—the manipulator and the manipulated lose their distinction. However, scientists rarely consider the entirety of the universe as an object of investigation. In most cases the scientist carves out a piece from the universe and proclaims that piece *in*—namely, the *focus* of investigation. The rest of the universe is then considered *out* or *background* and is summarized by what we call *boundary conditions*. This choice of *ins* and *outs* creates asymmetry in the way we look at things, and it is this asymmetry that permits us to talk about "outside intervention" and hence about causality and cause-effect directionality.

What we learn from causal inference therefore depends on what bounding box we choose. Too broad a bounding box can lead to a Byzantine model. Too narrow a bounding box can fail to capture salient aspects of the object of our study, leading to problems such as unobserved confounding. The bounding box must also resolve the implicit temporal aspects in a causal analysis, such as the delay between the time of measurement of different variables.

From a methodological perspective, the simulators in our work provide the surrounding system in which causal analysis takes place. Put in control of the surrounding system, we

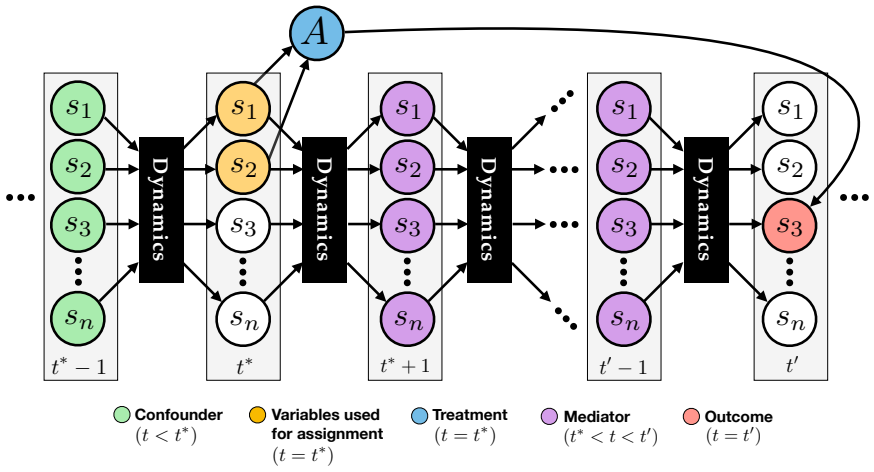


Figure 6.1: In *WhyNot*, scenarios for confounding and mediation naturally arise due to the temporal dynamics of the simulators. Abstractly, a simulation from *WhyNot* can be described by a state vector s_t and a dynamics model D that transitions the state vector from one time step to the next: $s_{t+1} = D(s_t)$. If treatment at time t^* is based on only part of the system's current state, then the variables in the past ($t < t^*$) are confounders. Similarly, any variable that occurs in between the time t^* of the treatment and the time t' of the outcome is a mediator.

can investigate how the analysis responds to changes in the surrounding system, such as choice of bounding box, which variables are in, which are out, and what temporal effects are at play.

Dynamics as background models

We adopt dynamical systems as *background models*, that is, models of the *outside* of the bounding box in which causal analysis takes place. Broadly speaking, a dynamical system is described by an initial state s_0 and a state-transition map D. Focusing on discrete time dynamics, the state of the system evolves according to the equation $D(s_t) = s_{t+1}$. In a dynamical system, an intervention corresponds to changing the state-transition rule from D to D' at an intervention time t^*, and then evolving the system according to D' after time t^*. The outcome variable is a measurement of the state $s_{t'}$ at some time t' in the future. For example, in the context of a city simulation, the intervention could correspond to raising the minimum wage at time t^*, and the outcome variable could be the level of unemployment at time t'.

Dynamical systems are a compelling choice as the *background model* for causal inference not least because *the temporal structure of the dynamics naturally gives rise to confounding and mediation*. A confounder has a causal effect on both the treatment and the outcome. In

a dynamical system, even if each component of the initial state is independent, the dynamics generally *couple the past* and render all of the state variables dependent. Thus, the entire collection of state variables from time $t = 0$ to $t = t^* - 1$ are confounders. Similarly, a mediator is causally affected by the treatment assignment and affects the outcome variable, and so all of the state variables from time $t = t^* + 1$ to $t = t' - 1$ are mediating variables. This mechanism of confounding and mediation is illustrate in Figure 6.1.

This connection between dynamics and questions of confounding and mediation provides the conceptual link between causal inference and dynamical systems. By simulating and intervening on a wide array of dynamics, we can generate a wide array of non-trivial causal inference problems.

Purpose of background models

The simulators in our work are not intended to be realistic models of the world, but rather realistic models of the technical difficulties that the real world poses for causal inference.

Put simply, the physical universe is a dynamical system. Causal inference in the real world therefore inevitably has to grapple with data that were generated by some dynamical process. The way confounding and mediation arise from dynamical systems, as we illustrated in Figure 6.1, is in a sense universal: We may postulate that all confounding and mediation in the real world would have to arise from the mechanism described in Figure 6.1 applied to some dynamical system. In other words, by varying the dynamical system used as the background model, we can in principle tease out all the challenges that the real world might pose.

Even if a simulator does not precisely capture the real world, it can entail confounding, mediation, and other issues in ways that resemble how the same phenomena would arise in reality. *WhyNot* invites the researcher to think about background models as a fruitful component of causal reasoning.

6.3 Background on causal inference

In this chapter, we consider estimating the causal effect of a binary treatment variable A on an outcome variable Y. Specifically, we are interested in the *average treatment effect*, [see e.g. 101],

$$\text{ATE} := \mathbb{E}[Y \mid do(A = 1)] - \mathbb{E}[Y \mid do(A = 0)],$$

where expectation is taken with respect to the population. We use the mathematical operator $do(a)$ introduced by Pearl [170] to denote an intervention that holds the value of a variable constant at level $A = a$. The average treatment effect can also be defined using the Neyman-Rubin potential outcomes framework [198, 218].[1] Our work is agnostic to which formalism the practitioner applies.

[1] In the potential outcomes framework, the ATE is commonly denoted using $\mathbb{E}[Y(1) - Y(0)]$, where $Y(a)$ is the potential outcome under treatment level $A = a$.

Due to the possibility of *confounding*, [170, 101, ch. 3, ch. 7 resp.] we cannot in general estimate the average treatment effect, and other quantities involving do-operations, using only observational data, i.e., samples $(A_i, Y_i)_{i=1}^n$ drawn i.i.d. from the joint distribution. In other words, the ATE is not always equal to the expected difference in outcome conditional on treatment, $\mathbb{E}[Y \mid A = 1] - \mathbb{E}[Y \mid A = 0]$.

Accurately estimating the effect of an intervention requires adjusting for confounders, that is variables that causally influence both the treatment and the outcome. In general, determining an *admissible* set of variables to properly adjust for confounding is a subtle question. Given a structural equation model of the causal problem, ideas like Pearl's *backdoor criterion* can solve this task. In the potential outcomes framework, a set of covariates X is admissible if the potential outcomes $Y(0), Y(1)$ are conditionally independent of treatment conditional on X, that is $Y(0), Y(1) \perp\!\!\!\perp A \mid X$. If X is a measured set of admissible covariates we say that *ignorability* or *unconfoundedness* hold.

In cases where ignorability is satisfied, all units have positive probability of treatment (*positivity*), and there is no interference across units (SUTVA), then causal inference is in principle possible and several algorithms (see Appendix 6.8) are able to consistently estimate causal effects. Unfortunately, these assumptions cannot be verified with observational data alone.

If there are confounders that are not adjusted for by the experimenter, then we say that there is *unobserved confounding* in the causal study. However, failing to include confounders is not the only way analysis can be biased. Mistakenly adjusting for *mediators*, that is variables that lie on a causal path from treatment to outcome can also introduce bias. For further discussion on these ideas, we refer the reader to the extensive literature on the subject [171, 169, 170, 212, 149, 101].

6.4 The *WhyNot* simulation framework

WhyNot is a software package designed to empirically evaluate causal inference methods with data from sophisticated computer simulations. The simulators incorporated within *WhyNot* provide a broad set of background models that allow the user to probe the causal inference toolkit in a variety of different contexts. A key facet of our framework is that ground truth causal effects are always available. We simply run the simulator both with and without treatment for a given unit. In this section, we give an overview of *WhyNot* and describe the simulators and inference algorithms included.

Overview of simulators. *WhyNot* provides a simple Python interface to several dynamical system simulators and a framework for constructing causal inference experiments on top of these simulators. The simulators in *WhyNot* can be split into two main categories: *system dynamics models* which use differential equations to model interactions between different components of a system over time and *agent-based models* which represent a system

as a collection of interacting, heterogeneous agents. We briefly describe the simulators used in our experiments and detail the remaining simulators in Appendix 6.7.

- **World3** is a system dynamics model that attempts to capture how quantities like population, industrial output, and pollution evolve in a hypothetical model of the entire world [160].
- **JAMA Opioid** is a system dynamics model of the US Opioid epidemic [44]. The model aims to describe how the populations of opioid users fluctuate over time and simulates the effect of policy interventions on opioid use and opioid-related deaths.
- **Civil Violence** is an agent-based model which attempts to capture the dynamics of civil uprisings amongst a heterogeneous group of protestors [73].

At the outset, these simulators are at best simplifications of the world they purport to describe. For instance, the validity and fidelity of the World3 has been widely challenged [56, 214]. Rather than using these simulators as proxies for real world phenomena we wish to understand, we instead view them as testbeds to examine causal inference in a multitude of experimental settings. The temporal nature of the system dynamics models naturally generate a host of non-trivial problems with mediation and confounding. The agent-based models allow for rich experiments in which there is interaction and interference between units. In each case, *WhyNot* broadens the scope of dynamics and experimental conditions that are amenable to study via simulation.

Algorithms. *WhyNot* makes a wide variety of causal inference methods readily accessible for evaluation. The set of methods available extends from classical workhorses like linear regression, IP weighting [238], and matching methods [207] to sophisticated modern methods like causal forests [237], causal BART [103], and TMLE [88]. The suite of methods we evaluate is described in Appendix 6.8.

6.5 Experiments

Our experiments focus on four fundamental issues in causal inference: (i) the statistical efficiency and estimation error of causal inference algorithms under *observed confounding*, (ii) the robustness and validity of causal procedures under *unobserved confounding*, (iii) the sensitivity of causal methods to *mediation*, and (iv) the impact of interference between units on causal estimates.

Our experiments are illustrative, intended to highlight the perspective and flexibility that *WhyNot* can contribute to evaluating the robustness inference procedures. The goal of the experiments is not to provide an exhaustive overview of all the questions that one can answer using our framework. We hope to inspire other researchers to use it as a creative tool to study settings of their own choosing.

Observed confounding—understanding sample efficiency and model misspecification

As a sanity check, we first examine causal inference methods under the conditions in which they are supposed to work, all standard causal inference assumptions—positivity, ignorability, and no interference—are *satisfied* and all confounders are *known and observed*. (See Appendix 6.3)

In this idealized setting, causal inference algorithms can in principle correct for the bias arising from confounders and consistently identify causal effects. Hence, our main targets for study are model misspecification and sample efficiency. Are the estimation methods expressive enough to accurately recover the causal effect? If so, how many samples do they need to return a precise answer?

Experimental setup. As described in Section 6.2, *WhyNot* allows us to easily create natural instances of confounding. To illustrate, we use the World3 and JAMA Opioid simulators as background models. Within these models, we consider the following causal questions:

- *What is the effect of reducing the rate of pollution generation by 50% in 1970 on the world population in 2050?*
- *What is the effect of lowering non-medical prescription opioid use in 2015 on the number of opioid overdose deaths in the United States in 2025?*

The worse the crisis, the more likely policy-makers are to intervene. For example, we imagine governments are more likely to intervene to reduce opioid abuse if the number of opioid overdose deaths is high. Concretely, we use the number of illicit opioid overdose deaths in 2015 to bias treatment assignment. Similarly, the worse pollution levels are in 1970, the more likely it is that an intervention occurs to reduce the rate of pollution generation.

Confounding now naturally arises from the temporal dynamics. In particular, as explained in Section 6.2, variables prior in time to the intervention are confounders. For example, the number of opioid overdose deaths in 2014 is a confounder because not only does it clearly influence the number of opioid overdose deaths in 2015, through its interaction with other variables via the system dynamics, it also influences the outcome, opioid overdose deaths in 2025.

To generate an observational dataset, we run multiple rollouts of the simulators under different initial conditions. Rollouts with higher levels of pollution in 1970 or opioid use in 2015 are more likely to receive treatment. To adjust for confounding, we control for the entire state vector at the time of treatment. Full details of the experimental procedure for all experiments are contained in Appendix 6.9.

Results: trading off bias and sample efficiency. In Figure 6.2, we plot the estimation error for the collection of estimators listed in Section 6.4 as a function of the sample size. On World3, the tree-based estimators (causal forest and causal BART) perform very

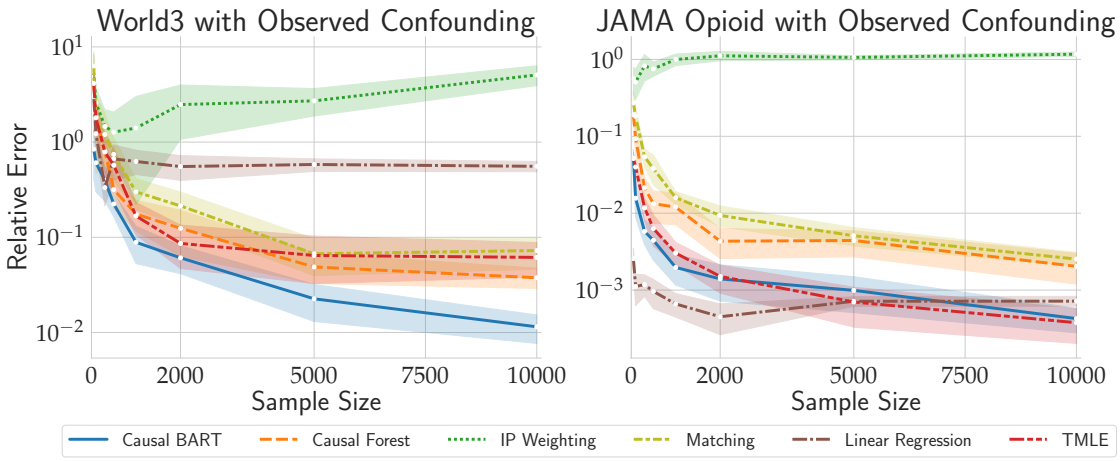


Figure 6.2: The performance of different causal methods as a function of sample-size. The more complex estimators are less biased than linear regression, but are also less sample efficient.

well, and the bias of the linear model is clearly visible. Although the tree-based estimators achieve roughly 10^{-2} relative error in estimating the ATE, linear regression flattens at 10^{-1}. Conversely, in the Opioid simulator, the misspecification of the linear model is mild, hence the improved statistical efficiency. The tree-based methods need 10000 samples to achieve the same error as linear regression with 100 samples.

These experiments are in the same vein as causal inference "data-science" competitions and previous simulation studies [118, 69]. Our findings corroborate the findings from the competitions: tree-based methods like causal forest and causal BART perform very well, and simple linear methods can provide a competitive baseline, but suffer from bias issues.

Unobserved confounding—do causal methods degrade gracefully?

We now illustrate how our framework can be used to better understand the validity of causal inference in settings where the causal assumptions are violated. In particular, in this section, we study *unobserved confounding*. It is well-known that whenever there is unobserved confounding, we cannot in general precisely estimate the true causal effect. However, practically speaking, how sensitive are inference methods to modest amount of unobserved confounding?

Experimental setup. We repurpose the experiments from the previous section to study plausible forms of unobserved confounding. Failing to control for all confounders often occurs when the analyst adjusts for variables used to determine treatment, e.g. 1970 pollution in World3, but not background variables that are superficially unrelated to the assignment rule, e.g. land fertility in 1969. Using dynamical systems as background models, we can natu-

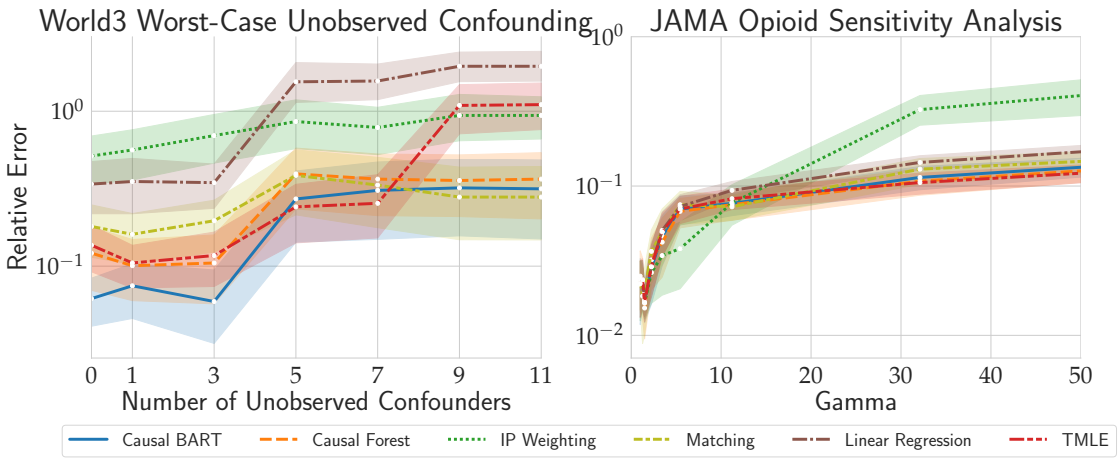


Figure 6.3: The performance of all the causal methods smoothly degrades as the strength of unobserved confounding increases.

rally generate unobserved confounding by excluding variables that do not directly determine treatment assignment from the analysis (see Section 6.2).

We probe the impact of varying amounts of unobserved confounding on the bias of causal procedures in two ways. First, for the World3 simulator, we analyze the effects of omitting the worst-case subset of *at most* k confounders as we increase k. Clearly, increasing k can only increase the amount of bias. Next, in the JAMA Opioid simulator, we smoothly vary a scalar sensitivity parameter Γ introduced by Rosenbaum [197]. In settings with no unobserved confounding, Γ equals 1, while domains with arbitrary amounts of confounding have Γ tending to ∞. In Appendix 6.9, we formally define this parameter Γ and provide describe how it can be manipulated in simulation.

Results: varying confounding strength. In Figure 6.3, we plot the performance of the estimators on (i) World3 when excluding the worst-case subsets of varying size and (ii) the JAMA Opioid simulator when varying Γ from $\Gamma = 1$ to $\Gamma = 50$. In the World3 experiment, the performance of estimators severely degrades after excluding as few as five confounding state variables. Moreover, TMLE, though very accurate when all confounders are observed, performs poorly with large numbers of omitted variables and eventually approaches the error of the linear regression baseline. In the JAMA Opioid experiment, the performance of all estimators degrades by a factor of 10 for Γ as small as 5. Moreover, for large Γ, the performance difference between estimators largely vanishes— the estimators all perform equally poorly when Γ is large.

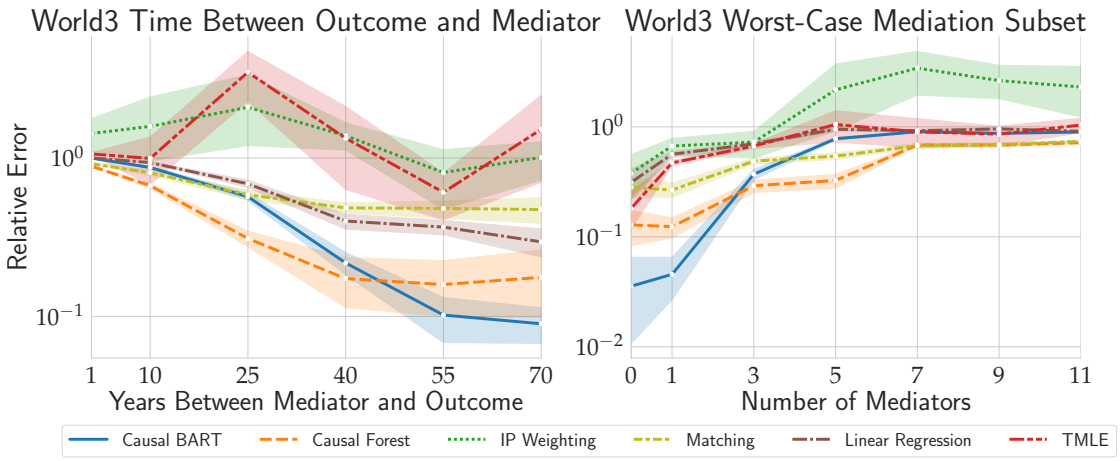


Figure 6.4: The performance of the causal methods as a function of the proximity in time between the mediator and the outcome variable (left) and as a function of the number of mediators (right). Once mediation becomes strong, all methods perform equally poorly.

Mediation—impact of adjusting for the wrong variables

In the same way that failing to adjust for confounders can bias inference results, mistakenly adjusting for *mediators* can also bias causal methods. However, mild mediation should only mildly impact estimator performance. In the same spirit as the previous section, we ask: how sensitive are causal inference methods to mistakenly adjusting for mediators?

Experimental setup. As discussed in Section 6.2, mediation arises naturally in the context of dynamical systems by *adjusting for state variables in time steps after the intervention*. Consequently, we repurpose the World3 and Opioid experiments to study mediation by adding state variables to the causal analysis after treatment assignment. To isolate the effect of mediation, all the confounders are always fully observed.

We explore two ways of generating causal problems with varying mediation strength. First, we vary the time index of the variables we choose to include in the analysis. State variables in the time steps immediately after intervention are only weakly related to the final outcome, while state variables immediately before the outcome time step have a strong mediating influence. Second, for a fixed time step, we include different subsets of the state variables. In general, the worst-case set of mediators of size at most $k-1$ has a weaker effect on estimation that the worst-case set of mediators of size at most k. The procedure to select the worst-case subset of mediators is detailed in Appendix 6.9.

Results: varying mediation strength. In Figure 6.4, we apply both strategies to generate instances with varying levels of mediation on World3 and plot the estimation error for each of the inference methods. First, we vary the distance in time between the medi-

ating variables and the outcome variable from 70 years (weak mediation) to 1 year (strong mediation). In the weak mediation regime, all the methods perform comparably to the no mediation regime, and there is a clear separation between tree-based methods and simpler models. However, with strong mediation, these performance differences disappear, and the models all have a constant error factor. A similar observation holds as we vary the size of the worst-case mediation set for a fixed time delay of 20 years. Including the single worst-case mediator only slightly increases estimation error for most methods. However, after adding the worst-case subset of 3 variables, the performance of all the methods rapidly deteriorates to little better than chance. In both cases, mediation experiments on JAMA Opioid simulator show similar phenomena. These plots are included in Appendix 6.9.

SUTVA—evaluating causal inference under interference

The starting point for many causal analyses is the assumption that the causal effect for one unit does not depend on the treatment received by any other unit. This *non-interference* assumption forms the core of the stable unit treatment value assumption (SUTVA) [57, 199]. However, in practice, treatment often does have spill-over effects and can cause interaction between units [216]. And notably, while randomized controlled trials (RCTs), the gold standard of causal inference, can guard against the confounding and mediation problems we have previously examined, they may still be biased due to interactions between units.

The agent-based models (ABMs) implemented in *WhyNot* provide controlled, yet non-trivial environments to test effect of SUTVA violations on causal methods. Since we can vary the amount of agent interaction within an ABM, we can use the these models to test whether the methods degrade gracefully or not by assessing their performance as a function of the level of agent interaction.

Experimental setup. We make use of Epstein's Civil Violence model, an agent-based simulator meant to capture the dynamics of urban uprising. In the model, agents have diverse features like individual hardship, belief in regime legitimacy, and risk aversion. If the individual's grievance (a function of hardship and legitimacy) exceeds the perceived risks of arrest, the agent chooses to rebel. In this simulator, we ask: *What is the effect of increasing risk aversion on an individual's frequency of rebellion?*

The treatment in this question (increasing risk aversion) naturally results in spill-over effects. Essentially, an agent is more likely to rebel if nearby agents are also rebelling. In particular, an agent's perceived arrest risk is inversely proportional to the number of other agents who are also rebelling in the same neighborhood.

We run an RCT in which 50% of agents randomly receive elevated levels of risk aversion. We evaluate the extent to which causal methods estimate the treatment effect for a single agent using this observational dataset. As a proxy for interaction strength, we vary the density of agents in the simulator. If the density is low, few agents appear in the neighborhood of a single agent, and thus interactions are weak, and vice-versa.

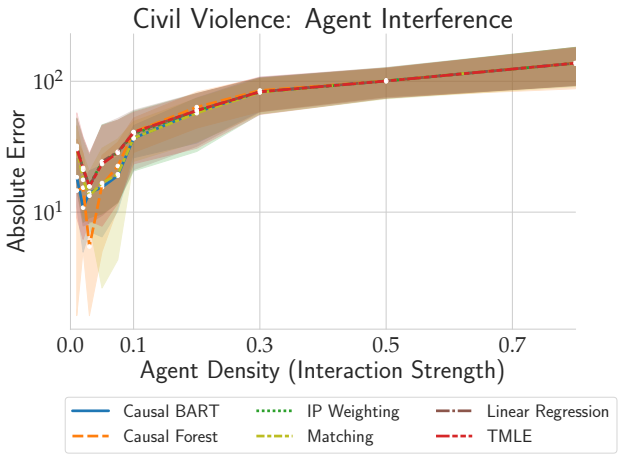


Figure 6.5: The estimation error of the causal methods as a function of the strength of interaction between agents. Surprisingly, none of the methods are particularly more robust than the others. As the agent density varies, the true causal effect always lies between 10 and 100 days active.

Results. In Figure 6.5, we plot the estimation error as a function of agent density. As the density of agents increases and interactions become stronger, all the causal inference procedures rapidly degrade. We expected some methods to do significantly better than others, particularly since in the experiments with unobserved confounding and mediation, we found a large gap in how robust different methods were. But surprisingly, there is little difference in performance between inference procedures when SUTVA is violated.

6.6 Conclusion and future work

At a conceptual level, our work introduces the idea of a background model as an important component of causal analysis. We argue that dynamical systems are natural background models in how they lead to interesting and general test cases for causal inference. Building on this perspective, we developed a powerful software package to advance the state of empirical robustness and validity analysis of causal inference. The package already includes numerous simulators and experimental designs.

Our work suggests many interesting future directions. Time-varying treatments and causal effects are a natural domain of investigation. Testing sequential decision-making and reinforcement learning is another task that our system in principle supports, but we haven't yet pursued. Expanding the set of simulators and inference methods is a useful task. Finally, establishing a set of interesting benchmarks for the causal inference community is now a realistic goal.

Table 6.1: Simulators currently implemented within *WhyNot*

Model	Domain	Model Class
World2	Conservation - Public Policy	System Dynamics
World3	Conservation - Public Policy	System Dynamics
JAMA Opioid	Epidemiology - Public Health	System Dynamics
Lotka-Volterra	Ecology	System Dynamics
SimpactCyan	Epidemiology - Public Health	Agent-Based Model
Civil Violence	Computational Social Science	Agent-Based Model
Schelling	Computational Social Science	Agent-Based Model
Graph Epidemic	Epidemiology - Public Health	Graph-Based Simulation
LaLonde	Economics	Response Model

6.7 Appendix: Additional *WhyNot* simulation environments

WhyNot currently includes nine different simulators which encompass a diverse set of dynamics of varying complexity. In Table 6.1, we provide a full list of all the different simulators currently implemented within the package and describe the domains for which the models were initially designed. We also include more detailed descriptions of the models below.

World2. This simulator is a system dynamics model developed by Jay Forrester to demonstrate the tension between industrial growth and natural resource limitations. The model was used to study how natural resource constraints can lead to slowdowns in industrial growth and eventual population collapse. The model is a system of differential equations in 5 variables corresponding to quantities like "population", "natural resources", and "pollution", and 43 algebraic equations governing their evolution over time [81].

World3. Designed as a successor to World2, World3 was introduced by the Club of Rome in the early 1970s to more accurately capture long term changes in dominant components of the "world system" [160, 159]. The model was revamped to have 41 state variables corresponding to quantities like "population between ages 15-44", "arable land", and "pollution" and 245 algebraic equations governing their evolution over time.

SimpactCyan. This is an agent-based simulator used within HIV epidemiology. The model was designed to simulate and study the transmission, treatment and prevention of HIV infections across a network of diverse individuals who interact over a long time window [140]. For instance, in the paper originally introducing the software, the authors estimate the impact of progressive changes to the eligibility criteria for HIV treatment on HIV incidence.

JAMA Opioid. Proposed by Chen et al. [44], the JAMA Opioid epidemic simulator is a system dynamics model of the US opioid epidemic. The model is calibrated based on past

opioid use data from the Center for Disease Control and was originally used to simulate the effect of various public health interventions at the national level, e.g. reducing the number of new non-medical users of opioids, on future opioid overdose deaths in the United States until the year 2030.

Civil Violence. This simulator is an agent-based model of civil violence introduced by Joshua Epstein in 2002 [73]. The model was originally used to study the complex dynamics of urban revolt by a set of agents who oppose the existing state. The model consists of two types of actors: agents and cops. Agents are heterogeneous, and their varied features make them more or less likely to actively rebel against the state. The complex dynamics of the model emerge from the interactions across different agents and police: agents are more likely to begin rebel if other agents start to rebel, and the cops attempt to arrest rebelling agents.

Lotka-Volterra. This is a classical differential equation model of the interactions between predator and prey in a single ecosystem. The model was originally developed to understand and explain perplexing fishery statistics during World War I— namely why the hiatus of fishing during the war led to an observed increase in the number of predators. The model itself consists of a coupled system of ordinary differential equations in two variables.

Schelling. This is one of the oldest agent-based models, originally used to illustrate how weak individual preferences regarding one's neighbors can lead to strong segregation of an entire population. Each individual prefers to live where at least some fraction of his neighbors are the same race as he is and will move if this constraint is not met. As this process is iterated, an originally well-mixed city rapidly becomes segregated by group [204].

Graph Epidemic This simulator describes the spread on a disease on a graph according to the SIS model. At each step, nodes in the graph can either be susceptible (S) or infected (I). If a susceptible node shares an edge with an infected node, then it can also become infected with a certain probability. Furthermore, infected nodes also have a constant probability of recovering and becoming susceptible again. For a given graph, SIR models allow for the evaluation of causal inference methods in contexts where the effects of interventions can rapidly spread through a network due to agent interactions [136].

LaLonde. The LaLande simulator is based off data from Robert LaLonde's 1986 study evaluating the impact of the National Supported Work Demonstration, a labor training program, on post-intervention income levels [128]. Since the actual function mapping the measured covariates to the observed outcomes is unknown, we instead simulate random functions of varying complexity on the data to generate synthetic outputs. This procedure allows us to generate causal inference problems with response surfaces of varying, but known complexity.

6.8 Appendix: Additional algorithms in *WhyNot*

As discussed previously, we complement our testbed of simulation environments with a comprehensive set of causal inference methods, all easily accessible through a simple Python interface. In particular, we integrate popular open source implementations of widely used

causal inference algorithms in order to ensure that the tools in *WhyNot* are closely representative of the kinds of approaches employed by practitioners.

To introduce these methods, we assume that all algorithms take in a dataset of covariates X, treatment assignments A, and outcomes Y, and return an estimate of the ATE, along with the relevant confidence intervals. If the dataset contains n observations, then X is an $(n \times d)$ matrix where each row contains the values of d adjustment variables. A is a binary vector with entry i equal to 1 if the ith unit received treatment. Lastly, Y is a real-valued vector of outcomes for each observed unit.

Linear Regression. One of the most common methods amongst causal inference practitioners, linear regression estimates the average treatment effect by using ordinary least squares to find the parameters of the following model:

$$Y = \tau \cdot A + X\beta$$

In this setup, the parameter τ denotes the ATE. We make use of standard implementations of OLS included within the Python `https://github.com/statsmodels/statsmodels` library to calculate model parameters and confidence intervals.

Matching. Another popular estimator, matching calculates the average treatment effects by pairing observations across treatment and control groups whose covariates are similar in some metric. Causal effects are then estimated by looking at the average difference in outcomes across the different pairs. We implement matching estimators by leveraging the R `https://cran.r-project.org/web/packages/Matching/index.html` library. For our experiments, we match covariates according to the inverse variance metric. We point the reader to the package documentation for specific implementation details.

IP Weighting. Inverse propensity score weighting estimates causal effects by creating a "pseudo-population" in which conditional ignorability holds and hence causal effects can be estimated via a difference of means (See [101, 111] for a full treatment of IP Weighting). We implement this approach within *WhyNot* by interfacing with the `https://cran.r-project.org/web/packages/WeightIt/index.html`WeighIt and `https://cran.r-project.org/web/packages/survey/index.html` packages in R. We fit propensity scores using generalized linear models.

Causal Forest. In addition to traditional causal inference procedures, we include more recent algorithms such as Causal Forest, introduced by Wager and Athey [237]. This method adapts Breiman's Random Forest algorithm [31] to causal inference settings. For our experiments, we make use of the `https://cran.r-project.org/web/packages/grf/index.html` R implementation, which was developed by the authors of the algorithm.

Causal BART Similarly to Causal Forest, Bayesian Additive Regression Trees algorithm (BART) is a tree-based method for estimating causal effects. Introduced by Hill [103], BART has consistently been one of the best performing algorithms in the ACIC Causal Inference Competitions [69]. We use Causal BART within the *WhyNot* framework by calling out to the `https://github.com/vdorie/bartCause/tree/master/R` package.

TMLE. Targeted Maximum Likelihood Learning (TMLE), introduced by Gruber, Laan, et al. [88] is another sophisticated causal inference estimator. We use the default implemen-

Table 6.2: Initial state distribution used in World3 experiments

Variable	Sampling Distribution
Population 0 to 14	Uniform[3.1e8, 1.4e9]
Population 15 to 44	Uniform[3.3e8, 1.5e9]
Population 45 to 64	Uniform[9.0e7, 4.0e8]
Population 65 and over	Uniform[2.8e7, 1.3e8]
Industrial Capital	Uniform[9.9e10, 4.4e11]
Service Capital	Uniform[6.8e10, 3.1e11]
Arable Land	Uniform[4.3e10, 1.9e9]
Potentially Arable Land	Uniform[1.1e9, 4.9e9]
Urban Industrial Land	Uniform[3.9e6, 1.7e7]
Land Fertility	Uniform[284, 1270]
Nonrenewable Resources	Uniform[4.7e11, 2.1e12]
Persistent Pollution	Uniform[1.2e7, 5.3e7]

Table 6.3: Initial state distribution used in JAMA Opioid experiments.

Variable	Sampling Distribution
Non-medical Opioid Users	$\mathcal{N}(1.0e7, 3.3e7)$
Non-medical Opioid Users with Prescription Use Disorder	$\mathcal{N}(1.4e6, 1.2e6)$
Illicit Opioid Users	$\mathcal{N}(3.3e5, 6.05e5)$

tation provided by the authors within the `https://cran.r-project.org/web/packages/tmle/index.html`tmle R package. We refer the reader to the original paper for a full description of the underlying algorithm.

6.9 Appendix: Additional experimental details

In this section, we provide complete details for our experimental protocol. We first precisely describe how we generate causal inference datasets for both the World3 and JAMA Opioid simulators since they appear repeatedly in our experiments. All of our experiments are run on a cluster computer with 72 Intel Xeon E7 CPUs and 1 TB of RAM.

World3. Each unit of analysis in World3 is a single run of the simulator. The original implementation of World3 is deterministic and studies a single "standard run" of the dynamics from fixed initial conditions, $s_0^* \in \mathbb{R}^{13}$. To generate stochasticity for the units, we start the simulator at a random initial state s_0 sampled from a simple distribution centered on s_0^*. Concretely, we sample each component independently as $s_{0i} = U_i s_{0i}^*$, where the U_i are independent and $U_i \sim \text{Uniform}[0.5, 2.5]$. The precise distribution for each variable is given in Table 6.2. A run of World3 then evolves the dynamics from the initial state in the year 1900 until 2010.

One of the parameters governing the evolution of World3 is the pollution generation rate. By default, this unit-less quantity is set to 1. Within the context of World3, we ask

What is the effect of reducing the rate of pollution generation by 50% in 1970 on the world population in 2050?

In the language of Appendix 6.3, the outcome variable Y is the total population in 2050. Regardless of treatment status, all rollouts evolve from 1900 to 1970 with pollution generation equal to 1. If a rollout is assigned to treatment $A = 1$, then the dynamics switch and evolve with pollution generation equal to 0.5 from 1970 to 2050. Otherwise, if the rollout is not treated, $A = 0$, the pollution generation parameter is unchanged, and the state evolves according to the standard dynamics from 1970 to 2050. Treatment assignment is based on the state of system in 1970. Specifically, the rollouts with the highest 10% of persistent pollution in 1970 are treated with 90% probability, while the remaining rollouts are treated with 10% probability. As discussed in Section 6.2, treatment assignment on the basis of the state in 1970 generates non-trivial confounding.

Generating the ground truth treatment effect for each individual rollout is trivial in simulation. We run the simulator forward under both sets of dynamics, under intervention and without intervention. For a collection of n rollouts, we take as ground truth the *sample average treatment effect*:

$$\text{SATE} := \hat{\mathbb{E}}[Y_i \mid do(A_i = 1)] - \hat{\mathbb{E}}[Y_i \mid do(A_i = 0)]\,,$$

where $\hat{\mathbb{E}}$ denotes the sample average.

JAMA Opioid A unit in the JAMA Opioid experiments is a single run of the simulator. The model is originally calibrated using data from the Center for Disease control, and, reflecting measurement uncertainty, Chen et al. [44] provide a distribution over the initial state. Each variable is sampled independently from this distribution, which is specified in Table 6.3. A run of the simulator then evolves the dynamics forward from this initial state in 2002 until 2015.

In JAMA Opioid, the causal question of interest is

What is the effect of lowering non-medical prescription opioid use in 2015 on the number of opioid overdose deaths in the United States in 2025?

The outcome variable Y is the total number of opioid overdose deaths in 2025. Regardless of treatment, all units evolve from 2002 to 2015 under the same base dynamics. If a rollout is treated, the incidence of non-medical opioid use is decreased by 11.3%, and the rollout is evolved forward under this new set of dynamics from 2015 to 2025. Otherwise, run is evolved under the base dynamics from 2015 to 2025. Treatment is assigned on the basis of the state in 2015— if the number of overdoses among illicit opioid users is in the top 10%, the run is treated with probability 90%. Otherwise, the run is treated with probability 10%. As in World3, we evaluate all rollouts under both treatment and control, and take the sample average treatment effect as ground truth.

Observed confounding

For all experiments, we plot the estimation error of each of the methods described in Appendix 6.8. Each trial is repeated 20 times, and we display 95% bootstrap confidence intervals.

Unobserved confounding

Defining the sensitivity parameter Γ. Rosenbaum [197] introduced Γ as a quantitative measure of the strength of unobserved confounding. Concretely, for a given unit i, let x_i the observed covariates, u_i denote the unobserved covariates, and $\pi_i = \Pr(A = 1 \mid x_i, u_i)$ denote the probability of treatment. Then, we say a causal inference problem has sensitivity Γ if, for any two units with the same observed covariates $x_i = x_j$, the log-odds $\pi/(1-\pi)$ differ by a factor of at most Γ. Formally,

$$\frac{1}{\Gamma} \leq \frac{\pi_i/(1-\pi_i)}{\pi_j/(1-\pi_j)} \leq \Gamma \quad \text{whenever } x_i = x_j.. \tag{6.1}$$

Notice Γ interpolates between fully observed confounding, $\Gamma = 1$, and unobserved confounding that makes inference arbitrarily bad, $\Gamma = \infty$.

Figure 6.3—varying Γ on JAMA Opioid. We generate problem instances with varying Γ as follows. In the JAMA Opioid experiment, rollouts are treated with probability p if they have a high levels of illicit opioid overdoses in 2015 and probability $1-p$ otherwise. By default, $p = 0.9$. Thus, in equation (6.1), we trivially have $\Gamma \leq \max\left\{\left(\frac{1-p}{p}\right)^2, \left(\frac{p}{1-p}\right)^2\right\}$. Therefore, we can construct a sequence of problems with varying (upper bounds on) Γ by varying the treatment probability p from $p = 1/2$ with $\Gamma = 1$ to $p \to 1$ which gives $\Gamma \to \infty$. In Figure 6.3, for each value of Γ, we generate 20 problem instances with a fixed sample size of 2000 units and report 95% bootstrap confidence intervals.

Figure 6.3—worst-case subsets of unobserved confounders. We use the following heuristic to identify the worst-case subset of unobserved confounders for a fixed size k. First, we sample a dataset of $n = 2000$ examples and compute the ground truth SATE. For each possible set of k state variables to exclude, we evaluate the estimation error of linear regression on the resulting inference problem. The subset of omitted variables which causes linear regression to have the highest error is declared the approximate worst-case subset of size k.

To generate Figure 6.3 for World3, we resample the data to avoid statistical dependencies between our evaluation and the procedure in which we choose the worst-case subset. For each subset of fixed size, we run 20 trials with $n = 2000$ points and report 95% bootstrap confidence intervals.

Mediation

In Figure 6.4, each experiment is run for 20 trials with $n = 2000$ samples, and we report 95% bootstrap confidence intervals. For the worst-case mediating variables experiment, we fix the mediation year to be 2020. For completeness, we include an identical time-varying experiment on the Opioid simulator in Figure 6.6.

Constructing the worst-case mediation set. Similar to Section 6.9, we employ the error of linear regression as a heuristic to identify worst-case mediating subsets. Concretely, we first sample a dataset of $n = 2000$ points. For a fixed choice of time index (2020 in the World3 experiment), we iterate over all possible subsets of mediating variables of size k, compute the estimation error of linear regression with this choice of mediating set, and declare the worst-case set of mediating variables to be the subset of size k that causes linear regression to have the highest error. To avoid statistical dependencies between the dataset in which we compute the worst-case mediator and the reported results, after fixing the worst-case subsets, we resample the data for each experiment.

SUTVA

As discussed previously, simulations within Epstein's model of Civil Violence contain two types of agents, citizens and cops, that randomly move on a grid. Each citizen is described via a set of five features: `vision`, `hardship`, `regime_legitimacy`, `risk_aversion`, and `active_threshold`. `vision` is an integer valued parameter that determines the size of the neighborhood of agents that are visible to a citizen, while the rest are all scalar values in the range $[0, 1]$

A citizen decides to go active, if the difference between `grievances` and `perceived_risk` are greater than the `active_threshold`. These quantities are defined as follows:

$$\texttt{grievances} := \texttt{hardship} \times (1 - \texttt{regime\_legitimacy})\,,$$
$$\texttt{perceived\_risk} := \texttt{arrest\_probability} \times \texttt{risk\_aversion}\,.$$

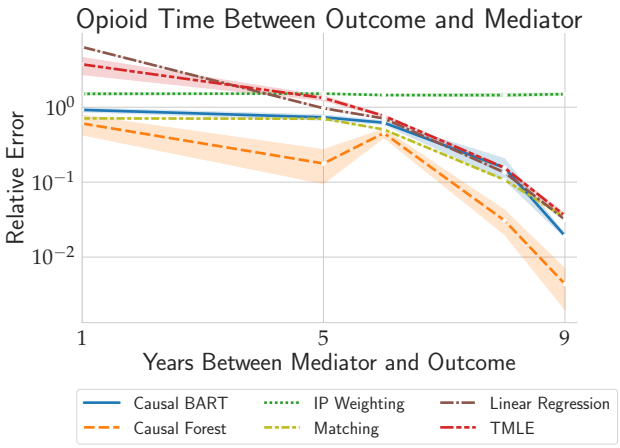


Figure 6.6: JAMA Opioid simulator as the time index of the mediating state variables, a proxy for mediation strength, varies from 1 year (strong mediation) to 10 years (weak mediation) before the outcome is measured.

The `arrest_probability` is inversely proportional to the ratio between the number of citizens that are active and the number of cops in vision. At every time step, cops choose a random active agent within their neighborhood to arrest. If arrested, an agent is in "prison" and cannot move for a fixed number of steps (an agent is not considered active if in prison). To further explore the effects of interactions across agents, we modify the original Epstein model to include a `prison_interaction` parameter. Defined as a constant within $[0, 1]$, this parameter varies the degree to which prison encourages groupthink across agents. In particular, during the start of a new prison sentence, we compute the maximum value of risk aversion across all citizens who are also in the prison state and happen to be in an agent's neighborhood (`max_aversion`). We then update the citizen's `risk_aversion` according to:

$$\texttt{new\_risk\_aversion} \mathrel{+}= \texttt{prison\_interaction} \times (\texttt{max\_aversion} - \texttt{old\_risk\_aversion}) .$$

The `agent_density` parameter is defined as ratio of the number of agents (citizens and cops) over the number of cells in the simulation grid. For very low density values, agents observe relatively few other citizens and cops in their neighborhood and hence there are few interactions across units. This dynamic changes once we increase the agent density. For our experiment on the Civil Violence simulator, we consider the following causal question:

What is the effect of increasing risk aversion on an individual's frequency of rebellion?

Running the RCT. The outcome variable Y in this experiment is the number of days (time steps) that an agent spends in rebellion over the course of the simulation. Contrary

to the previous two simulators, since we consider citizens to be the individual units of treatment, the outcomes for all units are determined simultaneously from a single rollout of the simulator. We run the simulator with 100 citizens and 5 cops for 1000 time steps and on progressively smaller grid sizes to vary agent density. For each agent, we sample hardship and regime legitimacy parameters uniformly from the range $[0, 1]$. We set the active threshold and risk aversion to 0.1. Furthermore, we set agent vision to 5 for both cops and citizens. Treatment is perfectly randomized in our experiment, with probability .5, each agent sees their risk aversion increased from 0.1 to 0.9.

Quality of recovering unit-level effect. The RCT allows us to estimate a population-level treatment effect:

$$\frac{1}{k}\sum_{i=1}^{k} Y_{i,\,do(A_{1,\ldots,k})=1} - \frac{1}{n-k}\sum_{i=k+1}^{n} Y_{i,\,do(A_{1,\ldots,k})=1}\,,$$

where n is the number of units, units $1, \ldots, k$ are the treated units, and units $k+1, \ldots, n$ are the untreated units. However, due to the agent interactions, the estimate of the population-level treatment effect may be a biased estimate of the unit level effect, which is the marginal effect for a single unit i when *no other agents are treated*: $Y_{i,\,do(A_i=1)} - Y_{i,\,do(A_i=0)}$.

We measure the ground-truth unit level causal effect by running a simulation in which only one agent receives treatment, and then measuring the number of days this agent spends active. Then, we compare the discrepancy between the effect that the causal methods recover using the RCT data and the ground-truth unit level effect.

Bibliography

[1] Rediet Abebe et al. "Narratives and Counternarratives on Data Sharing in Africa". In: *Proc. of the ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 329–341.

[2] Alekh Agarwal et al. "A reductions approach to fair classification". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 60–69.

[3] Anders Andreassen et al. *The Evolution of Out-of-Distribution Robustness Throughout Fine-Tuning*. https://arxiv.org/abs/2106.15831. 2021.

[4] Martin Arjovsky et al. "Invariant risk minimization". In: *arXiv preprint arXiv:1907.02893* (2019).

[5] Maximilian Augustin and Matthias Hein. *Out-distribution aware Self-training in an Open World Setting*. https://arxiv.org/abs/2012.12372. 2020.

[6] Keith Ball. "An elementary introduction to modern convex geometry". In: *Flavors of geometry* (1997). http://library.msri.org/books/Book31/files/ball.pdf.

[7] Peter Bandi et al. "From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge". In: *IEEE Transactions on Medical Imaging* (2018). https://ieeexplore.ieee.org/document/8447230.

[8] Michelle Bao et al. "It's COMPASlicated: The Messy Relationship between RAI Datasets and Algorithmic Fairness Benchmarks". In: *arXiv preprint arXiv:2106.05498* (2021).

[9] Andrei Barbu et al. "ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2019). https://objectnet.dev/.

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

[11] Solon Barocas and Andrew D. Selbst. "Big Data's Disparate Impact". In: *California Law Review* 104 (2016).

[12] Raef Bassily et al. "Algorithmic stability for adaptive data analysis". In: *Symposium on Theory of Computing (STOC)*. https://arxiv.org/abs/1511.02513. 2016.

[13] Jason Baumgartner et al. "The Pushshift Reddit Dataset". In: *arXiv preprint arXiv:2001.08435* (2020).

[14] Sara Beery, Elijah Cole, and Arvi Gjoka. "The iWildCam 2020 Competition Dataset". In: *Fine-Grained Visual Categorization Workshop at the Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/2004.10340. 2020.

[15] Rachel KE Bellamy et al. "AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias". In: *IBM Journal of Research and Development* 63.4/5 (2019), pp. 4–1.

[16] Shai Ben-David et al. "A theory of learning from different domains". In: *Machine learning* (2010). https://link.springer.com/article/10.1007/s10994-009-5152-4.

[17] Shai Ben-David et al. "Analysis of Representations for Domain Adaptation". In: *Advances in neural information processing systems (NeurIPS)*. https://papers.nips.cc/paper/2006/hash/b1b0432ceafb0ce714426e9114852ac7-Abstract.html. 2006.

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

[19] Jonathan Berant et al. "Modeling biological processes for reading comprehension". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1499–1510.

[20] Battista Biggio and Fabio Roli. "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning". In: *Pattern Recognition* (2018). https://arxiv.org/abs/1712.03141.

[21] Battista Biggio et al. "Evasion Attacks against Machine Learning at Test Time". In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD)*. https://arxiv.org/abs/1708.06131. 2013.

[22] Sarah Bird et al. "Fairlearn: A toolkit for assessing and improving fairness in AI". In: *Microsoft, Tech. Rep. MSR-TR-2020-32* (2020).

[23] Matthew Blackwell. "A selection bias approach to sensitivity analysis for causal effects". In: *Political Analysis* 22.2 (2014), pp. 169–182.

[24] Blender Online Community. *Blender - a 3D modelling and rendering package*. http://www.blender.org. Blender Foundation. Blender Institute, Amsterdam, 2021.

[25] John Blitzer et al. "Learning bounds for domain adaptation". In: *Advances in Neural Information Processing Systems (NeurIPS)*. https://papers.nips.cc/paper/2007/hash/42e77b63637ab381e8be5f8318cc28a2-Abstract.html. 2007.

[26] Avrim Blum and Moritz Hardt. "The Ladder: A Reliable Leaderboard for Machine Learning Competitions". In: *International Conference on Machine Learning*. 2015, pp. 1006–1014.

[27] Tolga Bolukbasi et al. "Man is to computer programmer as woman is to homemaker? debiasing word embeddings". In: *Advances in Neural Information Processing Systems* (2016).

[28] Eric Bonabeau. "Agent-based modeling: Methods and techniques for simulating human systems". In: *Proceedings of the national academy of sciences* 99.suppl 3 (2002), pp. 7280–7287.

[29] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101–mining discriminative components with random forests". In: *European conference on computer vision*. Springer. 2014, pp. 446–461.

[30] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000). `https://opencv.org/`.

[31] Leo Breiman. "Random Forests". In: *Mach. Learn.* 45.1 (Oct. 2001), pp. 5–32. ISSN: 0885-6125. DOI: `10.1023/A:1010933404324`. URL: `https://doi.org/10.1023/A:1010933404324`.

[32] Leo Breiman. "Random forests". In: *Machine learning* (2001). `https://link.springer.com/article/10.1023/A:1010933404324`.

[33] Greg Brockman et al. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[34] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems (NeurIPS)*. `https://arxiv.org/abs/2005.14165`. 2020.

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

[36] Benjamin Caine et al. *Pseudo-labeling for Scalable 3D Object Detection*. `https://arxiv.org/abs/2103.02093`. 2021.

[37] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. "Building Classifiers with Independency Constraints". In: *In Proc. IEEE ICDMW*. 2009, pp. 13–18.

[38] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. "Semantics derived automatically from language corpora contain human-like biases". In: *Science* 356.6334 (2017), pp. 183–186.

[39] Berk Calli et al. *Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols*. `https://arxiv.org/abs/1502.03143`. 2015.

[40] Berk Calli et al. "The ycb object and model set: Towards common benchmarks for manipulation research". In: *2015 international conference on advanced robotics (ICAR)*. IEEE. 2015, pp. 510–517.

[41] Yair Carmon et al. "Unlabeled data improves adversarial robustness". In: *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/1905.13736. 2019.

[42] Mathilde Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2006.09882. 2020.

[43] Abhishek Chaurasia and Eugenio Culurciello. "LinkNet: Exploiting encoder representations for efficient semantic segmentation". In: *Visual Communications and Image Processing (VCIP)*. https://arxiv.org/abs/1707.03718. 2017.

[44] Qiushi Chen et al. "Prevention of Prescription Opioid Misuse and Projected Overdose Deaths in the United States". In: *JAMA network open* 2.2 (2019), e187621–e187621.

[45] Ting Chen et al. "Big Self-Supervised Models are Strong Semi-Supervised Learners". In: *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2006.10029. 2020.

[46] Xinlei Chen and Kaiming He. *Exploring Simple Siamese Representation Learning*. https://arxiv.org/abs/2011.10566. 2020.

[47] Yunpeng Chen et al. "Dual path networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/1707.01629. 2017.

[48] François Chollet. "Xception: Deep learning with depthwise separable convolutions". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1610.02357. 2017.

[49] Alexandra Chouldechova. "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments". In: *Big data* 5.2 (2017), pp. 153–163.

[50] Gordon Christie et al. "Functional map of the world". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1711.07846. 2018.

[51] Kenneth Ward Church. "Emerging trends: A tribute to Charles Wayne". In: *Natural Language Engineering* 24.1 (2018), pp. 155–160.

[52] Adam Coates, Andrew Ng, and Honglak Lee. "An analysis of single-layer networks in unsupervised feature learning". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. http://proceedings.mlr.press/v15/coates11a.html. 2011.

[53] Adam Coates and Andrew Y Ng. "Learning feature representations with k-means". In: *Neural networks: Tricks of the trade*. https://www-cs.stanford.edu/~acoates/papers/coatesng_nntot2012.pdf. Springer, 2012.

[54] *CodaLab*. https://competitions.codalab.org/competitions/.

[55] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. "Certified Adversarial Robustness via Randomized Smoothing". In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1902.02918. 2019.

[56] H.S.D. Cole. *Models of Doom: A Critique of the Limits to Growth*. Universe Books, 1975. URL: https://books.google.com/books?id=rQRm5AAACAAJ.

[57] David Roxbee Cox. *Planning of experiments*. Wiley, 1958.

[58] Ekin D Cubuk et al. "RandAugment: Practical automated data augmentation with a reduced search space". In: *Advances in Neural Information Processing Systems (NeurIPS)*. https://papers.nips.cc/paper/2020/hash/d85b63ef0ccb114d0a3bb7b7d808028f-Abstract.html. 2020.

[59] Alexander D'Amour et al. *Underspecification presents challenges for credibility in modern machine learning*. https://arxiv.org/abs/2011.03395. 2020.

[60] Mohammad Zalbagi Darestani, Akshay Chaudhari, and Reinhard Heckel. "Measuring robustness in deep learning based compressive sensing". In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/2102.06103. 2021.

[61] Luke N Darlow et al. *CINIC-10 is not ImageNet or CIFAR-10*. https://arxiv.org/abs/1810.03505. 2018.

[62] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. http://www.image-net.org/papers/imagenet_cvpr09.pdf. 2009.

[63] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. http://www.image-net.org/papers/imagenet_cvpr09.pdf. 2009.

[64] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: (2018). http://arxiv.org/abs/1810.04805.

[65] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[66] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. http://www.szit.bme.hu/~gyorfi/pbook.pdf. Springer, 1996.

[67] Josip Djolonga et al. "On Robustness and Transferability of Convolutional Neural Networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/2007.08558. 2021.

[68] Jeff Donahue et al. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition". In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1310.1531. 2014.

[69] Vincent Dorie et al. "Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition". In: *Statistical Science* 34.1 (2019), pp. 43–68.

[70] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2010.11929. 2021.

[71] Matthew Dunn et al. "Searchqa: A new q&a dataset augmented with context from a search engine". In: *arXiv preprint arXiv:1704.05179* (2017).

[72] Cynthia Dwork et al. "Preserving statistical validity in adaptive data analysis". In: *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. 2015, pp. 117–126.

[73] Joshua M Epstein. "Modeling civil violence: An agent-based computational approach". In: *Proceedings of the National Academy of Sciences* 99.suppl 3 (2002), pp. 7243–7250.

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

[75] Mark Everingham et al. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

[76] Vitaly Feldman, Roy Frostig, and Moritz Hardt. "The advantages of multiple classes for reducing overfitting from test set reuse". In: *arXiv preprint arXiv:1905.10360* (2019).

[77] Vitaly Feldman, Roy Frostig, and Moritz Hardt. "The advantages of multiple classes for reducing overfitting from test set reuse". In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1905.10360. 2019.

[78] Adam Fisch et al. "MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension". In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1–13. DOI: 10.18653/v1/D19-5801.

[79] Sarah Flood et al. *Integrated Public Use Microdata Series, Current Population Survey: Version 8.0 [dataset]*. Minneapolis, MN: IPUMS, https://doi.org/10.18128/D030.V8.0. 2020.

[80] Jay W Forrester. *Urban Dynamics*. MIT Press, 1969.

[81] Jay W Forrester. *World Dynamics*. Wright-Allen Press, 1971.

[82] D.D. Gatti et al. *Agent-Based Models in Economics: A Toolkit*. Cambridge University Press, 2018. ISBN: 9781108243988. URL: https://books.google.com/books?id=Vq1XDwAAQBAJ.

[83] Timnit Gebru et al. "Datasheets for datasets". In: *arXiv:1803.09010* (2018).

[84] Pascal Germain et al. "A new PAC-Bayesian perspective on domain adaptation". In: *International conference on machine learning (ICML)*. https://arxiv.org/abs/1506.04573. 2016.

[85] Pascal Germain et al. "A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers". In: *International conference on machine learning (ICML)*. http://proceedings.mlr.press/v28/germain13.html. 2013.

[86] Sven Gowal et al. "Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples". https://arxiv.org/abs/2010.03593. 2020.

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

[88] Susan Gruber, Mark van der Laan, et al. "tmle: An R Package for Targeted Maximum Likelihood Estimation". In: *Journal of Statistical Software* 51.i13 (2012).

[89] Ishaan Gulrajani and David Lopez-Paz. "In Search of Lost Domain Generalization". In: *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2007.01434. 2021.

[90] Moritz Hardt, Eric Price, and Nati Srebro. "Equality of Opportunity in Supervised Learning". In: *Proc. 29th NIPS*. 2016, pp. 3315–3323.

[91] Moritz Hardt and Benjamin Recht. *Patterns, predictions, and actions: A story about machine learning*. https://mlstory.org, 2021. arXiv: 2102.05242 [cs.LG].

[92] Moritz Hardt and Benjamin Recht. *Patterns, predictions, and actions: Foundations of machine learning*. Princeton University Press, 2022.

[93] Kunimatsu Hashimoto et al. "KOSNet: A Unified Keypoint, Orientation and Scale Network for Probabilistic 6D Pose Estimation". http://groups.csail.mit.edu/robotics-center/public_papers/Hashimoto20.pdf. 2020.

[94] Trevor Hastie et al. "Multi-class AdaBoost". In: *Statistics and its Interface* (2009). http://ww.web.stanford.edu/~hastie/Papers/SII-2-3-A8-Zhu.pdf.

[95] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[96] Kaiming He et al. "Deep residual learning for image recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1512.03385. 2016.

[97] Kaiming He et al. "Identity mappings in deep residual networks". In: *European Conference on Computer Vision (ECCV)*. https://arxiv.org/abs/1603.05027. 2016.

[98] Dan Hendrycks and Thomas Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1903.12261. 2018.

[99] Dan Hendrycks et al. "Natural Adversarial Examples". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1907.07174. 2021.

[100] Dan Hendrycks et al. *The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization*. https://arxiv.org/abs/2006.16241. 2020.

[101] MA Hernán and JM Robins. *Causal Inference*. Boca Raton: Chapman & Hall/CRC, forthcoming, 2019.

[102] Miguel A Hernán et al. "Comment: Spherical Cows in a Vacuum: Data Analysis Competitions for Causal Inference". In: *Statistical Science* 34.1 (2019), pp. 69–71.

[103] Jennifer L Hill. "Bayesian nonparametric modeling for causal inference". In: *Journal of Computational and Graphical Statistics* 20.1 (2011), pp. 217–240.

[104] Stefan Hinterstoisser et al. "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes". In: *Asian Conference on Computer Vision*. https://link.springer.com/chapter/10.1007/978-3-642-37331-2_42. 2012.

[105] Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. "Algorithms and theory for multiple-source adaptation". In: *International Conference on Neural Information Processing Systems (ICML)*. https://arxiv.org/abs/1805.08727. 2018.

[106] Matthew Honnibal and Ines Montani. "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing". To appear. 2017.

[107] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation networks". In: *Conference on Computer Vsion and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1709.01507. 2018.

[108] Gao Huang et al. "Densely connected convolutional networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1608.06993. 2017.

[109] Forrest N. Iandola et al. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡0.5MB model size*. https://arxiv.org/abs/1602.07360. 2016.

[110] Kosuke Imai, Luke Keele, and Teppei Yamamoto. "Identification, Inference and Sensitivity Analysis for Causal Mediation Effects". In: *Statist. Sci.* 25.1 (Feb. 2010), pp. 51–71. DOI: 10.1214/10-STS321. URL: https://doi.org/10.1214/10-STS321.

[111] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[112] Díaz Iván and van der Laan Mark J. "Sensitivity Analysis for Causal Inference under Unmeasured Confounding and Measurement Error Problems". In: *The International Journal of Biostatistics* 9.2 (Nov. 2013), pp. 149–160. URL: https://ideas.repec.org/a/bpj/ijbist/v9y2013i2p149-160n8.html.

[113] Robin Jia and Percy Liang. "Adversarial Examples for Evaluating Reading Comprehension Systems". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 2021–2031.

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

[115] Mandar Joshi et al. "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 1601–1611.

[116] Alex Kendall, Matthew Grimes, and Roberto Cipolla. "PoseNet: A convolutional network for real-time 6-DOF camera relocalization". In: *International Conference on Computer Vision (ICCV)*. `https://arxiv.org/abs/1505.07427`. 2015.

[117] Nikita Kitaev and Dan Klein. "Constituency Parsing with a Self-Attentive Encoder". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018.

[118] Michael Knaus, Michael Lechner, and Anthony Strittmatter. "Machine learning estimation of heterogeneous causal effects: Empirical monte carlo evidence". In: (2018).

[119] Allison Koenecke et al. "Racial disparities in automated speech recognition". In: *Proceedings of the National Academy of Sciences* 117.14 (2020), pp. 7684–7689.

[120] Pang Wei Koh et al. "WILDS: A Benchmark of in-the-Wild Distribution Shifts". In: *International Conference on Machine Learning (ICML)*. `https://arxiv.org/abs/2012.07421`. 2020.

[121] Ronny Kohavi and Barry Becker. "UCI Adult Data Set". In: *UCI Meachine Learning Repository* 5 (1996).

[122] Simon Kornblith, Jonathon Shlens, and Quoc V Le. "Do better ImageNet models transfer better?" In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. `https://arxiv.org/abs/1805.08974`. 2019.

[123] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`. 2009.

[124] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. `http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf`. 2009.

[125] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. `https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks`. 2012.

[126] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NIPS)*. `https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks`. 2012.

[127] Tom Kwiatkowski et al. "Natural questions: a benchmark for question answering research". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 453–466.

[128] Robert J LaLonde. "Evaluating the econometric evaluations of training programs with experimental data". In: *The American economic review* (1986), pp. 604–620.

[129] John Langford. "Quantitatively Tight Sample Complexity Bounds". `http://hunch.net/~jl/projects/prediction_bounds/thesis/thesis.pdf`. PhD thesis. Carnegie Mellon University, 2002.

[130] Pat Langley. *The changing science of machine learning*. 2011.

[131] Yann LeCun, Corinna Cortes, and Christopher Burges. "MNIST handwritten digit database". `http://yann.lecun.com/exdb/mnist/`. 1998.

[132] Seanie Lee, Donggyu Kim, and Jangwon Park. "Domain-agnostic question-answering with adversarial training". In: *arXiv preprint arXiv:1910.09342* (2019).

[133] Vincent Lepetit and Pascal Fua. "Monocular Model-Based 3D Tracking of Rigid Objects: A Survey". In: *Foundations and Trends in Computer Graphics and Vision* (2005). `https://ieeexplore.ieee.org/document/8187270`.

[134] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. "EPnP: An accurate o(n) solution to the PnP problem". In: *International Journal of Computer Vision* (2009). `https://link.springer.com/article/10.1007/s11263-008-0152-6`.

[135] Amanda Levendowski. "How copyright law can fix artificial intelligence's implicit bias problem". In: *Wash. L. Rev.* 93 (2018), p. 579.

[136] Jianquan Li, Jie Lou, and Meizhi Lou. "Some discrete SI and SIS epidemic models". In: *Applied Mathematics and Mechanics* 29.1 (Jan. 2008), pp. 113–119. ISSN: 1573-2754. DOI: `10.1007/s10483-008-0113-y`. URL: `https://doi.org/10.1007/s10483-008-0113-y`.

[137] Liam Li and Ameet Talwalkar. "Random search and reproducibility for neural architecture search". In: *Conference on Uncertainty in Artificial Intelligence (UAI)*. `https://arxiv.org/abs/1902.07638`. 2019.

[138] Xiao Li and Jeff Bilmes. "A bayesian divergence prior for classiffier adaptation". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. `http://proceedings.mlr.press/v2/li07a.html`. 2007.

[139] Mark Liberman. "Obituary: Fred Jelinek". In: *Computational Linguistics* 36.4 (2010), pp. 595–599.

[140] Jori Liesenborgs et al. "SimpactCyan 1.0: An Open-source Simulator for Individual-Based Models in HIV Epidemiology with R and Python Interfaces". In: *bioRxiv* (2018), p. 440834.

[141] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[142] Chenxi Liu et al. "Progressive neural architecture search". In: *European Conference on Computer Vision (ECCV)*. https://arxiv.org/abs/1712.00559. 2018.

[143] Hanxiao Liu, Karen Simonyan, and Yiming Yang. "Darts: Differentiable architecture search". In: *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1806.09055. 2019.

[144] Nelson F Liu et al. *Can Small and Synthetic Benchmarks Drive Modeling Innovation? A Retrospective Study of Question Answering Modeling Approaches*. https://arxiv.org/abs/2102.01065. 2021.

[145] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1411.4038. 2015.

[146] Shayne Longpre et al. "An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering". In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 220–227.

[147] Shangyun Lu et al. "Harder or Different? A Closer Look at Distribution Shift in Dataset Reproduction". In: *ICML Workshop on Uncertainty and Robustness in Deep Learning*. http://www.gatsby.ucl.ac.uk/~balaji/udl2020/accepted-papers/UDL2020-paper-101.pdf. 2020.

[148] Ningning Ma et al. "ShuffleNet V2: Practical guidelines for efficient CNN architecture design". In: *European Conference on Computer Vision (ECCV)*. https://arxiv.org/abs/1807.11164. 2018.

[149] Marloes H. Maathuis and Diego Colombo. "A Generalized Back-Door Criterion". In: *The Annals of Statistics* 43.3 (2015), pp. 1060–1088. ISSN: 00905364.

[150] Aleksander Madry et al. "Towards deep learning models resistant to adversarial attacks". In: *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1706.06083. 2018.

[151] Jeffrey Mahler et al. "Learning ambidextrous robot grasping policies". In: *Science Robotics* 4.26 (2019), eaau4984.

[152] Horia Mania and Suvrit Sra. *Why do classifier accuracies show linear trends under distribution shift?* https://arxiv.org/abs/2012.15483. 2020.

[153] Horia Mania et al. "Model similarity mitigates test set overuse". In: *Advances in Neural Information Processing Systems*. 2019, pp. 9993–10002.

[154] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. "Domain adaptation with multiple sources". In: *Advances in neural information processing systems (NeurIPS)* (2008). https://papers.nips.cc/paper/2008/hash/0e65972dce68dad4d52d063967f0a705-Abstract.html.

[155] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. "Domain adaptation: Learning bounds and algorithms". In: *Conference on Learning Theory (COLT)*. `https://arxiv.org/abs/0902.3430`. 2009.

[156] Sandra Mathison. *Encyclopedia of evaluation*. Sage publications, 2004.

[157] Julian McAuley et al. "Image-based recommendations on styles and substitutes". In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2015, pp. 43–52.

[158] R. T. McCoy, J. Min, and T. Linzen. "BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance". In: *BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. `https://arxiv.org/abs/1911.02969`. 2019.

[159] Dennis Meadows and Jorgan Randers. *The limits to growth: the 30-year update*. Routledge, 2012.

[160] Donella H Meadows et al. "The limits to growth: a report to the club of Rome (1972)". In: *Google Scholar* (1972).

[161] John Miller et al. "The Effect of Natural Distribution Shift on Question Answering Models". In: *International Conference on Machine Learning (ICML)*. `https://arxiv.org/abs/2004.14444`. 2020.

[162] Mimi Onuoha. "The Point of Collection". In: *Data & Society: Points* (2016).

[163] Reham Osama, Nagwa El-Makky, and Marwan Torki. "Question Answering Using Hierarchical Attention on Top of BERT Features". In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Nov. 2019.

[164] Boyuan Pan et al. "Memen: Multi-layer embedding with memory networks for machine comprehension". In: *arXiv preprint arXiv:1707.09098* (2017).

[165] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* (2010). `https://ieeexplore.ieee.org/document/5288526`.

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

[167] Amandalynne Paullada et al. "Data and its (dis) contents: A survey of dataset development and use in machine learning research". In: *arXiv preprint arXiv:2012.05345* (2020).

[168] Georgios Pavlakos et al. "6-DoF object pose from semantic keypoints". In: *International Conference on Robotics and Automation (ICRA)*. `https://arxiv.org/abs/1703.04670`. 2017.

[169] Judea Pearl. "Causal diagrams for empirical research". In: *Biometrika* 82.4 (Dec. 1995), pp. 669–688.

[170] Judea Pearl. *Causality: Models, Reasoning and Inference*. 2nd. New York, NY, USA: Cambridge University Press, 2009. ISBN: 052189560X, 9780521895606.

[171] Judea Pearl. "Comment: Graphical Models, Causality and Intervention". In: *Statist. Sci.* Bayesian Analysis in Expert Systems 8.3 (Aug. 1993), pp. 266–269.

[172] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* (2011). https://www.jmlr.org/papers/v12/pedregosa11a.html.

[173] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[174] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. "Discrimination-aware Data Mining". In: *Proc. 14th SIGKDD*. ACM, 2008.

[175] Matthew E Peters et al. "Deep contextualized word representations". In: *Proceedings of NAACL-HLT*. 2018, pp. 2227–2237.

[176] Matthew E. Peters et al. "Deep Contextualized Word Representations". In: *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. https://arxiv.org/abs/1802.05365. 2018.

[177] Vinay Uday Prabhu and Abeba Birhane. "Large image datasets: A pyrrhic win for computer vision?" In: *arXiv preprint arXiv:2006.16923* (2020).

[178] 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/.

[179] Riyi Qiu. Personal Communication. 2020.

[180] Joaquin Quionero-Candela et al. *Dataset Shift in Machine Learning*. https://mitpress.mit.edu/books/dataset-shift-machine-learning. The MIT Press, 2009.

[181] Mahdi Rad and Vincent Lepetit. "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth". In: *International Conference on Computer Vision (ICCV)*. https://arxiv.org/abs/1703.10896. 2017.

[182] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/2103.00020. 2021.

[183] Ilija Radosavovic et al. "Designing network design spaces". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/2003.13678. 2020.

[184] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. "Certified Defenses against Adversarial Examples". In: *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1801.09344. 2018.

[185] Pranav Rajpurkar. Personal Communication. 2019.

[186] Pranav Rajpurkar, Robin Jia, and Percy Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2018, pp. 784–789.

[187] Pranav Rajpurkar et al. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://arxiv.org/abs/1606.05250. 2016.

[188] Ali Sharif Razavian et al. "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. https://arxiv.org/abs/1403.6382. 2014.

[189] Benjamin Recht et al. "Do CIFAR-10 Classifiers Generalize to CIFAR-10?" https://arxiv.org/abs/1806.00451. 2018.

[190] Benjamin Recht et al. "Do ImageNet Classifiers Generalize to ImageNet?" In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1902.10811. 2019.

[191] Ievgen Redko et al. *A survey on domain adaptation theory: learning bounds and theoretical guarantees*. https://arxiv.org/abs/2004.11829. 2020.

[192] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Semantically Equivalent Adversarial Rules for Debugging NLP Models". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 856–865.

[193] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. "Mctest: A challenge dataset for the open-domain machine comprehension of text". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 193–203.

[194] Alexander Robey, George J Pappas, and Hamed Hassani. *Model-Based Domain Generalization*. https://arxiv.org/abs/2102.11436. 2021.

[195] Rebecca Roelofs et al. "A Meta-Analysis of Overfitting in Machine Learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 9175–9185.

[196] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. https://arxiv.org/abs/1505.04597. 2015.

[197] Paul R Rosenbaum. *Design of observational studies*. Vol. 10. Springer, 2010.

[198] D.B. Rubin. "Estimating causal effects of treatments in randomized and nonrandomized studies". In: *Journal of Educational Psychology* 66.5 (1974), pp. 688–701.

[199] Donald B Rubin. "Randomization analysis of experimental data: The Fisher randomization test comment". In: *Journal of the American Statistical Association* 75.371 (1980), pp. 591–593.

[200] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* (2015). https://arxiv.org/abs/1409.0575.

[201] Olga Russakovsky et al. "ImageNet large scale visual recognition challenge". In: *International Journal of Computer Vision* (2015). https://arxiv.org/abs/1409.0575.

[202] Mark Sandler et al. "MobileNetV2: Inverted residuals and linear bottlenecks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1801.04381. 2018.

[203] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. "BREEDS: Benchmarks for Subpopulation Shift". In: *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2008.04859. 2021.

[204] Thomas C Schelling. "Dynamic models of segregation". In: *Journal of mathematical sociology* 1.2 (1971), pp. 143–186.

[205] Ludwig Schmidt et al. "Adversarially Robust Generalization Requires More Data". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2018). https://arxiv.org/abs/1804.11285.

[206] Matthew Schwall et al. "Waymo public road safety performance data". In: *arXiv preprint arXiv:2011.00038* (2020).

[207] Jasjeet S Sekhon. "Multivariate and propensity score matching software with automated balance optimization: the matching package for R". In: *Journal of Statistical Software, Forthcoming* (2008).

[208] Minjoon Seo et al. "Bidirectional attention flow for machine comprehension". In: *arXiv preprint arXiv:1611.01603* (2016).

[209] Vaishaal Shankar et al. *Do Image Classifiers Generalize Across Time?* https://arxiv.org/abs/1906.02168. 2019.

[210] Vaishaal Shankar et al. "Evaluating Machine Accuracy on ImageNet". In: *International Conference on Machine Learning (ICML)*. http://proceedings.mlr.press/v119/shankar20c.html. 2020.

[211] Vaishaal Shankar et al. "Evaluating machine accuracy on ImageNet". In: *International Conference on Machine Learning (ICML)*. http://proceedings.mlr.press/v119/shankar20c.html. 2020.

[212] Ilya Shpitser, Tyler VanderWeele, and James M. Robins. "On the validity of covariate adjustment for estimating causal effects". English (US). In: *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, UAI 2010*. Dec. 2010, pp. 527–536. ISBN: 9780974903965.

[213] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: (2015). https://arxiv.org/abs/1409.1556.

[214] V. Smil. *Energy at the Crossroads: Global Perspectives and Uncertainties*. Mit Press. MIT Press, 2005. ISBN: 9780262693240. URL: https://books.google.com/books?id=2UM6KSEMoLUC.

[215] Adam Smith. *Information, privacy and stability in adaptive data analysis*. https://arxiv.org/abs/1706.00820. 2017.

[216] Michael E Sobel. "What do randomized studies of housing mobility demonstrate? Causal inference in the face of interference". In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1398–1407.

[217] Oleg Sofrygin, Mark J van der Laan, and Romain Neugebauer. "Simcausal R package: conducting transparent and reproducible simulation studies of causal effect estimation with complex longitudinal data". In: *Journal of statistical software* 81 (2017).

[218] Jerzy Splawa-Neyman, Dorota M Dabrowska, and TP Speed. "On the application of probability theory to agricultural experiments. Essay on principles. Section 9." In: *Statistical Science* (1990), pp. 465–472.

[219] Chris Stokel-Walker. *Why Zillow couldn't make Algorithmic House pricing work*. Nov. 2021. URL: https://www.wired.com/story/zillow-ibuyer-real-estate/.

[220] Xiao Sun et al. "Integral human pose regression". In: *European Conference on Computer Vision (ECCV)*. https://arxiv.org/abs/1711.08229. 2018.

[221] Christian Szegedy et al. "Going deeper with convolutions". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1409.4842v1. 2015.

[222] Christian Szegedy et al. "Going deeper with convolutions". In: *Conference on Computer vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1409.4842. 2015.

[223] Christian Szegedy et al. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *AAAI Conference on Artificial Intelligence (AAAI)*. https://arxiv.org/abs/1602.07261. 2017.

[224] Christian Szegedy et al. "Intriguing properties of neural networks". In: *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1312.6199. 2014.

[225] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1512.00567. 2016.

[226] Alon Talmor and Jonathan Berant. "MultiQA: An Empirical Investigation of Generalization and Transfer in Reading Comprehension". In: *arXiv preprint arXiv:1905.13453* (2019).

[227] Mingxing Tan and Quoc Le. "EfficientNet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1905.11946. 2019.

[228] Rohan Taori et al. "Measuring robustness to natural distribution shifts in image classification". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2020). https://arxiv.org/abs/2007.00644.

[229] David Tellez et al. "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology". In: *Medical image analysis* (2019). https://arxiv.org/abs/1902.06543.

[230] David Tellez et al. "Whole-slide mitosis detection in H&E breast histology using PHH3 as a reference to train distilled stain-invariant convolutional networks". In: *IEEE Transactions on Medical Imaging* (2018). https://arxiv.org/abs/1808.05896.

[231] Josh Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 23–30.

[232] Emanuel Todorov, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control". In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2012, pp. 5026–5033.

[233] Antonio Torralba and Alexei A Efros. "Unbiased look at dataset bias". In: *CVPR 2011*. IEEE. 2011, pp. 1521–1528.

[234] Antonio Torralba, Rob Fergus, and William T. Freeman. "80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008). https://people.csail.mit.edu/torralba/publications/80millionImages.pdf.

[235] Jonathan Tremblay et al. "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects". In: *Conference on Robot Learning (CoRL)*. https://arxiv.org/abs/1809.10790. 2018.

[236] Adam Trischler et al. "NewsQA: A Machine Comprehension Dataset". In: *Proceedings of the 2nd Workshop on Representation Learning for NLP*. 2017, pp. 191–200.

[237] Stefan Wager and Susan Athey. "Estimation and inference of heterogeneous treatment effects using random forests". In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1228–1242.

[238] Willem M van der Wal, Ronald B Geskus, et al. "Ipw: an R package for inverse probability weighting". In: *J Stat Softw* 43.13 (2011), pp. 1–23.

[239] Alex Wang et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding". In: *arXiv preprint arXiv:1804.07461* (2018).

[240] Alex Wang et al. "Superglue: A stickier benchmark for general-purpose language understanding systems". In: *Advances in neural information processing systems* 32 (2019).

[241] Mei Wang and Weihong Deng. "Deep visual domain adaptation: A survey". In: *Neurocomputing* (2018). https://arxiv.org/abs/1802.03601.

[242] Adina Williams, Nikita Nangia, and Samuel R Bowman. "The multi-genre nli corpus". In: (2018).

[243] Andrew Wong et al. "External validation of a widely implemented proprietary sepsis prediction model in hospitalized patients". In: *JAMA Internal Medicine* 181.8 (2021), pp. 1065–1070.

[244] Eric Wong and Zico Kolter. "Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope". In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1711.00851. 2018.

[245] Yu Xiang et al. "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes". In: *Robotics: Science and Systems (RSS)*. https://arxiv.org/abs/1711.00199. 2017.

[246] Saining Xie et al. "Aggregated residual transformations for deep neural networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1611.05431. 2017.

[247] Chhavi Yadav and Léon Bottou. "Cold Case: The Lost MNIST Digits". In: (2019). https://arxiv.org/abs/1905.10498.

[248] Chhavi Yadav and Léon Bottou. "Cold Case: The Lost MNIST Digits". In: *Advances in Neural Information Processing Systems (NeurIPS)*. http://arxiv.org/abs/1905.10498. 2019.

[249] Chhavi Yadav and Léon Bottou. "Cold case: The lost mnist digits". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13443–13452.

[250] Kaiyu Yang et al. "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*. 2020, pp. 547–558.

[251] Zhilin Yang et al. "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 2369–2380.

[252] Zhilin Yang et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *arXiv preprint arXiv:1906.08237* (2019).

[253] Dani Yogatama et al. "Learning and Evaluating General Linguistic Intelligence". In: *arXiv preprint arXiv:1901.11373* (2019).

[254] Bin Yu. "Stability". In: *Bernoulli* (2013). https://projecteuclid.org/journals/bernoulli/volume-19/issue-4/Stability/10.3150/13-BEJSP14.full.

[255] Jiahui Yu et al. "Coca: Contrastive captioners are image-text foundation models". In: *arXiv preprint arXiv:2205.01917* (2022).

[256] John R Zech et al. "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study". In: *PLoS medicine* 15.11 (2018), e1002683.

[257] Richard Zemel et al. "Learning fair representations". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning*. 2013, pp. III–325.

[258] Xiangyu Zhang et al. "ShuffleNet: An extremely efficient convolutional neural network for mobile devices". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1707.01083. 2018.

[259] Xingcheng Zhang et al. "Polynet: A pursuit of structural diversity in very deep networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1611.05725. 2016.

[260] Hengshuang Zhao et al. "Pyramid scene parsing network". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1612.01105. 2017.

[261] X. Zhou et al. *The Curse of Performance Instability in Analysis Datasets: Consequences, Source, and Suggestions*. https://arxiv.org/abs/2004.13606. 2020.

[262] Zongwei Zhou et al. "UNet++: A nested U-Net architecture for medical image segmentation". In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. https://arxiv.org/abs/1807.10165. 2018.

[263] Barret Zoph et al. "Learning Transferable Architectures for Scalable Image Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1707.07012. 2018.

[264] Tijana Zrnic and Moritz Hardt. "Natural Analysts in Adaptive Data Analysis". In: *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1901.11143. 2019.