WILDS: A Benchmark of in-the-Wild Distribution Shifts

Pang Wei Koh* and **Shiori Sagawa*** {pangwei, ssagawa}@cs.stanford.edu
Henrik Marklund marklund@stanford.edu
Sang Michael Xie xie@cs.stanford.edu
Marvin Zhang marvin@eecs.berkeley.edu
Akshay Balsubramani abalsubr@stanford.edu
Weihua Hu weihuahu@stanford.edu
Michihiro Yasunaga myasu@stanford.edu
Richard Lanas Phillips richard@cs.cornell.edu
Sara Beery sbeery@caltech.edu
Jure Leskovec jure@cs.stanford.edu
Anshul Kundaje akundaje@stanford.edu
Emma Pierson epierson@microsoft.com
Sergey Levine svlevine@eecs.berkeley.edu
Chelsea Finn cbfinn@cs.stanford.edu
Percy Liang pliang@cs.stanford.edu

Correspondence to: wilds@cs.stanford.edu

Abstract

Distribution shifts can cause significant degradation in machine learning (ML) systems deployed in the wild, and they manifest as a practical challenge across a broad range of real-world applications. However, many widely-used datasets in the ML community today were not designed for evaluating distribution shifts. These datasets typically have training and test sets drawn from the same distribution, and prior work on retrofitting them with distribution shifts has generally relied on artificially introducing shifts that can be unrealistic. In this paper, we present WILDS, a benchmark of in-the-wild distribution shifts spanning diverse data modalities and applications, from tumor identification to wildlife monitoring to poverty mapping. WILDS builds on top of recent data collection efforts by domain experts in these applications and provides a unified collection of datasets with evaluation metrics and train/test splits that are representative of real-world distribution shifts. These datasets reflect distribution shifts arising from training and testing on different hospitals, camera locations, countries, time periods, demographics, molecular scaffords, etc., all of which cause substantial performance drops in our baseline models. Finally, we survey other application areas that would be promising additions to the benchmark but for which we did not manage to find appropriate datasets; for these applications, we discuss their associated challenges as well as detail datasets and shifts where we did not see an appreciable performance drop. By unifying datasets from a variety of application areas and making them accessible to the ML community, we hope to encourage the development of general-purpose methods that are anchored to real-world distribution shifts and that work well across different application areas and problem settings. Data loaders, default models, and leaderboards are available at https://wilds.stanford.edu.

∗. These authors contributed equally to this work.

. NeurIPS 2020 Workshop: ML Retrospectives, Surveys & Meta-Analyses (ML-RSA).

Contents

1. Overview

Distribution shifts—mismatches in data distributions between training and test time—pose significant challenges for machine learning (ML) systems deployed in the wild. These shifts arise naturally in many real-world scenarios. In this work, we study two common distribution shift settings: *domain generalization* and *subpopulation shift* (Figure 1). In the domain generalization setting, the training and test distributions comprise data from related but distinct domains; prior work has shown that model performance can degrade substantially on, e.g., patients from different hospitals (Zech et al., 2018; Beede et al., 2020; DeGrave et al., 2020), images taken by different cameras (Beery et al., 2018; D'Amour et al., 2020), biological assays from different cell types (Li et al., 2019a), or satellite images from different countries (Jean et al., 2016) and time periods (Christie et al., 2018). In the subpopulation shift setting, the test distribution is a subpopulation of the training distribution; e.g., models can systematically fail on people from minority subpopulations (Buolamwini and Gebru, 2018; Borkan et al., 2019b; Koenecke et al., 2020), raising issues of equity and generalization.

Despite the ubiquity of distribution shifts in real-world applications, many of the widely-used datasets in the ML community today were not designed for evaluating models under distribution shifts. These datasets typically have train and test sets drawn from the same distribution, and prior work on retrofitting them with distribution shifts has generally relied on artificially introducing shifts that need not represent the kinds of shifts encountered in the wild. For example, a substantial amount of recent work in the ML community has focused on object recognition tasks with distribution shifts induced by synthetic transformations, using datasets such as Colored MNIST (Arjovsky et al., 2019), which changes the colors of MNIST digits; ImageNet-C (Hendrycks and Dietterich, 2019), which corrupts ImageNet images with noise; and the Backgrounds Challenge (Xiao et al., 2020) and Waterbirds (Sagawa et al., 2020), which swap out image backgrounds. It is also common to split the data to induce a shift that is more extreme than one might typically encounter, e.g., in PACS (Li et al., 2017a), which includes classifying photos solely from cartoons and other stylized representations; DeepFashion Remixed (Hendrycks et al., 2020b), which involves classifying objects at different scales solely from objects at a single scale; or BREEDS (Santurkar et al., 2020), which uses a training set that has disjoint subclasses from the test set. These datasets are important for method development; they encode distribution shifts that are controllable, targeted, and convenient to work with. However, they are not sufficient to ensure that these methods can translate to real-world shifts.

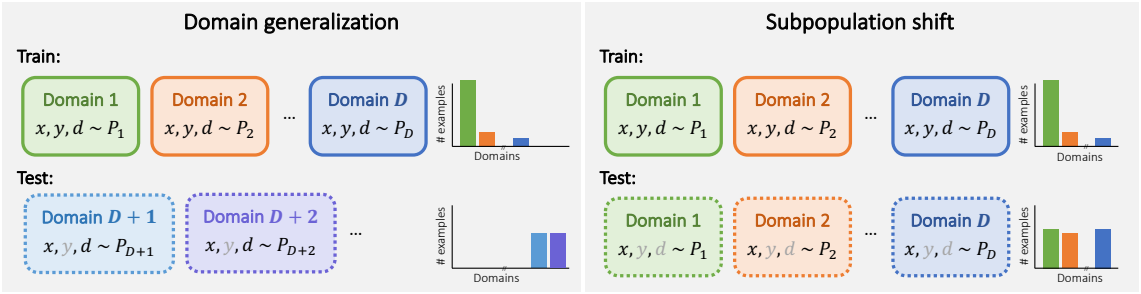

Figure 1: In a WILDS dataset, each data point comes from a domain d, e.g., patients from different hospitals or images from different cameras. The training and test distributions comprise different mixtures of domains. We focus on two particular distribution shift settings. **Left:** In domain generalization, the training and test distributions comprise related but distinct domains, e.g., the training set might be from one set of hospitals and the test set might be from a different set of hospitals. **Right:** In subpopulation shift, the training and test domains overlap, but their relative proportions differ. For example, we might be interested in test performance on a particular minority subpopulation (domain). In this setting, we generally assess models by their worst performance over all subpopulations of interest, and we assume that domain information is unavailable at test time.

Dataset	Input (x)	Prediction target (y)	Examples	Domains	Domain count	Train/test domain overlap
FMoW (Christie et al., 2018)	satellite images	land use	523,846	time regions	16 5	✗ ✓
PovertyMap (Yeh et al., 2020)	satellite images	asset wealth	19,669	countries urban/rural	23 2	✗ ✓
iWildCam2020 (Beery et al., 2020a)	camera trap photos	animal species	217,609	trap locations	291	✗
Camelyon17 (Bandi et al., 2018)	tissue slides	tumor	455,954	hospitals	5	✗
OGB-MolPCBA (Hu et al., 2020b)	molecular graphs	bioassays	437,929	molecular scaffolds	120,084	✗
Amazon (Ni et al., 2019)	product reviews	sentiment	1,400,382	users	7,642	✗
CivilComments (Borkan et al., 2019b)	online comments	toxicity	448,000	demographics	16	✓

Table 1: The Wilds benchmark contains 7 datasets across a diverse set of application areas and data modalities. Each dataset comprises data from different domains, and the benchmark is set up to evaluate models on distribution shifts across these domains.

On the other hand, earlier (pre-deep-learning) work in the ML research community has studied datasets with more realistic distribution shifts, but these are not as widely used today as they tend to be much smaller than modern datasets, e.g., in object recognition (Saenko et al., 2010; Gong et al., 2012), sentiment analysis (Blitzer et al., 2007; Pan et al., 2010), part-of-speech tagging (Marcus et al., 1993; Daumé III, 2007), flow cytometry (Blanchard et al., 2011; Muandet et al., 2013), or land cover classification (Bruzzone and Marconcini, 2009).

In contrast to general-purpose ML research, domain experts applying ML in their respective areas are often forced to grapple with distribution shifts in order to make progress on real-world problems. As a result, these application areas are rich sources of datasets with distribution shifts that arise in the wild, e.g., in medicine (Chen et al., 2020), computational biology (Leek et al., 2010), wildlife conservation (Beery et al., 2018), satellite imagery (Jean et al., 2016), and so on. Unfortunately, these datasets can be less accessible and convenient for ML researchers who are not domain experts, and existing models and methods for mitigating these shifts can be highly domain-specific, e.g., specialized color-normalizing techniques to handle variation in histopathological staining procedures across hospitals (Tellez et al., 2019). Our goal is to bridge this gap.

In this paper, we present Wilds, a curated collection of benchmark datasets with evaluation metrics and train/test splits that we believe are representative of the kinds of distribution shift challenges that ML algorithms face when deployed in applications in the wild (Table 1). Wilds datasets span a diverse array of societally-important applications with natural distribution shifts: poverty mapping (Yeh et al., 2020), building and land use classification (Christie et al., 2018), animal species categorization (Beery et al., 2020a), predicting text toxicity (Borkan et al., 2019b), sentiment analysis (Ni et al., 2019), tumor identification (Bandi et al., 2018), and bioassay prediction (Hu et al., 2020b). At present, there are 7 datasets in Wilds (Table 1), reflecting distribution shifts arising from different demographics, users, hospitals, camera locations, countries, time periods, molecular scaffolds, and cell types.

WILDS builds on top of extensive data-collection efforts by domain experts. To design WILDS, we worked with domain experts and researchers studying distribution shifts to identify and modify datasets to fulfill the following criteria:

1. **Distribution shifts with corresponding performance drops.** The train/test splits reflect distribution shifts that cause model performance to substantially degrade, i.e., with a large gap between in-distribution and out-of-distribution performance.

2. **Real-world relevance.** The training/test splits and evaluation metrics are motivated by real-world scenarios, and chosen in conjunction with domain experts to be consistent with prior work in their corresponding applications.

3. **Potential leverage.** Distribution shift benchmarks must strike a fine balance between being non-trivial but also being possible to solve, as it is unreasonable to expect a model to generalize to arbitrary distribution shifts. We approach this by ensuring that each WILDS dataset comprises training data from multiple domains, with domain annotations and other metadata available at training time. This information can then be used to learn models that perform well under distribution shift: e.g., for domain generalization, one could use these domain annotations to learn models that do not rely on domain-specific features, while for subpopulation shift, one could learn models that perform uniformly well across each domain.

Most of the WILDS datasets have been substantially modified to make them more user-friendly and consistent, as well as to have train/test splits that reflect the distribution shift of interest. By including datasets from a variety of application areas and making them accessible to the ML community through careful preprocessing, standardized evaluations, and shared infrastructure, we hope to encourage the development of *general-purpose* methods that are anchored to real-world distribution shifts and that can work well across different application areas.

To contextualize the WILDS benchmark, we discuss each dataset's broader context and relation to other tasks and distribution shifts in the same application area. In Section 6, we further survey other application areas—algorithmic fairness and policing, medicine and healthcare, natural language and speech processing, code, education, and robotics—that would be promising sources for future additions to the benchmark, but for which we did not manage to find appropriate datasets. We discuss the challenges associated with these areas as well as some datasets and shifts in those areas for which we did not see an appreciable performance drop. In Section 7, we discuss potential extensions to other problem settings, such as test-time adaptation and selective classification, that can also be tested on WILDS. Finally, in Section 8, we discuss some empirical trends that we observed across the WILDS datasets, e.g., underspecification and the effect of distribution shifts across multiple axes.

WILDS is available as an open-source Python package. It has two main components:

1. **Data loaders.** Dataset downloading, preprocessing, and splitting are fully automated by the WILDS package. For each dataset, we provide a standardized PyTorch data loader that returns user-customizable minibatches of (x, y, m), where the metadata m contains information like the domain identity (e.g., which hospital that patient's data came from).

2. **Dataset evaluators.** To allow for consistent comparisons between models and training algorithms, we provide dataset-dependent evaluation functions that measure the performance of a model according to the metric chosen for that dataset.

To use WILDS, one needs to specify a **model architecture** (e.g., BERT) and a **training algorithm** (e.g., empirical risk minimization, a distributionally robust learning algorithm, etc.). To facilitate general-purpose algorithm development, we also provide default models and hyperparameters that we used for the baselines described in this paper. All baseline results can be easily replicated with the WILDS package. We discuss package usage and provide illustrative code snippets in Section 9.

To track the state-of-the-art on the WILDS datasets, we are hosting public leaderboards for each dataset. We encourage the community to develop and submit general-purpose, model-agnostic training algorithms that can work across multiple datasets; the code is set up to make it convenient for users to write a new training algorithm and then run it across any WILDS dataset using its default model. We also welcome modeling improvements, e.g., new model architectures that are more robust to domain shifts. We aim for WILDS to be extensible and community-driven, and we welcome other researchers to contribute distribution shift datasets in their area of work that fit the criteria above. Code, leaderboards, and updates are available at https://wilds.stanford.edu.

2. Comparison with existing ML benchmarks

Developing models that are robust to distribution shifts is an active area of research in the ML research community, and many benchmarks have been proposed. WILDS focuses on realistic distribution shifts; to situate our work among other ML benchmarks, we first discuss what we mean by the realism of a benchmark.

Realism is subtle to pin down and highly contextual, and assessing realism often requires consulting with domain experts and practitioners. As a general framework, we can view a benchmark dataset as comprising the data, a task and associated evaluation metric, and a train/test split that potentially reflects a distribution shift. Each of these components can independently be more or less realistic:

1. The **data** is realistic if it accurately reflects what would plausibly be collected (and available for a model to use) in a real application, and it includes not just the inputs x but also any associated metadata (e.g., the domain that each data point came from). The realism of data also depends on the application context; for example, using medical images captured with state-of-the-art equipment might be realistic for well-equipped hospitals, but not necessarily for clinics that use older generations of the technology, or vice versa. Extreme examples of unrealistic data include the Gaussian distributions that are often used to cleanly illustrate the theoretical properties of various algorithms.

2. The **task and evaluation metric** is realistic if the task is relevant to a real application and if the metric measures how successful a model would be in that application. Here and with the other components, realism is a spectrum. For example, in a wildlife conservation application where the input is images from camera traps, the real task might be to estimate species populations Parham et al. (2017), i.e., the number of distinct individual animals of each species seen in the overall collection of images; a task that is less realistic but still relevant and useful for ecologists might be to classify what species of animal is seen in each image Tabak et al. (2019). The choice of evaluation metric is also important. In the wildlife example, conservationists might care more about rare species than common species, so measuring average classification accuracy would be less realistic than a metric that prioritizes getting the rare species correct.

3. The **distribution shift (train/test split)** is realistic if it reflects training and test distributions that might arise in deployment for that dataset and task. For example, if a medical algorithm is trained on data from a few hospitals and then expected to be deployed more widely, then it would be realistic to test it on hospitals that are not in the training set. On the other hand, an example of a less realistic shift is to, for instance, train a pedestrian classifier entirely on daytime photos and then test it only on nighttime photos; in practice, any reasonable dataset for pedestrian detection that is used in a real application would include photos from both daytime and nighttime.

Recent public benchmarks in the ML research community[1] have tended to prioritize distribution shifts that are more controllable and accessible, at the cost of realism and diversity.[2] Specifically, they have largely focused on visual object recognition tasks, and on shifts induced by synthetic transformations, unrealistic data splits, or dataset combinations.

We stress that these existing benchmarks are still useful testbeds for method development. However, it is also important to ensure that methodological progress translates to real-world settings. The extent to which model robustness transfers from one type of shift to another is still an open question (Taori et al., 2020; Hendrycks et al., 2020b); for example, a method that improves robustness on a standard vision dataset can consistently harm robustness on real-world satellite imagery datasets (Xie et al., 2020). This underscores the importance of evaluating directly on distribution shifts encountered in the wild.

With WILDS, we seek to complement existing ML benchmarks by focusing on datasets with realistic distribution shifts across diverse data modalities and important application areas. We now discuss some of these existing benchmarks, categorizing them by how they induce their respective distribution shifts.

2.1 Prior work on ML benchmarks for distribution shifts

Distribution shifts from transformations. Some of the most widely-adopted benchmarks induce distribution shifts by synthetically transforming the data. Examples include rotated and translated versions of MNIST and CIFAR (Worrall et al., 2017; Gulrajani and Lopez-Paz, 2020); surface variations such as texture, color, and corruptions like blur in Colored MNIST (Gulrajani and Lopez-Paz, 2020), Stylized ImageNet (Geirhos et al., 2018), and ImageNet-C (Hendrycks and Dietterich, 2019); and datasets that crop out objects and replace their backgrounds, as in the Backgrounds Challenge (Xiao et al., 2020) and other similar datasets (Sagawa et al., 2020; Koh et al., 2020). Benchmarks for adversarial robustness also fall in this category of distribution shifts from transformations (Goodfellow et al., 2015; Croce et al., 2020). Though adversarial robustness is not a focus of this work, we note that recent work on temporal perturbations with the ImageNet-Vid-Robust and YTBB-Robust datasets (Shankar et al., 2019) represents a different form of distribution shift that also impacts real-world applications. Outside of visual object recognition, other work has used synthetic datasets and transformations to explore compositional generalization, e.g., SCAN (Lake and Baroni, 2018). We discuss this more in Section 6.

Synthetic-to-real transfers. Fully synthetic datasets such as SYNTHIA (Ros et al., 2016) and StreetHazards (Hendrycks et al., 2020a) have been adopted for out-of-distribution detection as well as domain adaptation and generalization, e.g., by testing robustness to transformations in the seasons, weather, time, or architectural style (Hoffman et al., 2018; Volpi et al., 2018). While the data is synthetic, it can still look realistic if a high-fidelity simulator is used. In particular, synthetic benchmarks that study transfers from synthetic to real data (Ganin and Lempitsky, 2015; Richter et al., 2016; Peng et al., 2018) can be important tools for tackling real-world problems: even though the data is synthesized and by definition, not real, the synthetic-to-real distribution shift can still be realistic in contexts where real data is much harder to acquire than synthetic data (Bellemare et al., 2020). In this work, we do not study these types of synthetic distribution shifts; instead, we focus on distribution shifts that occur in the wild between data distributions that are not synthetically generated.

1. It is difficult to draw a clear line between work in the "ML research community" and in adjacent communities; for example, there is also a lot of work on distribution shifts in the natural language processing (NLP) research community, among others. In this section, we mostly focus on work that has appeared in ML conferences and journals; we discuss related work from other communities in Sections 5 and 6.
2. Others have studied proprietary datasets with realistic distribution shifts, such as the StreetView StoreFronts dataset (Hendrycks et al., 2020b) or diabetic retinopathy datasets (D'Amour et al., 2020). However, these datasets are not public for privacy and other reasons.

Distribution shifts from constrained splits. Other benchmarks do not rely on transformations but instead split the data in a way that induces particular distribution shifts. These benchmarks have realistic data, e.g., the data points are derived from real-world photos, though their distribution shifts do not necessarily reflect shifts that would arise in the wild. For example, BREEDS (Santurkar et al., 2020) tests generalization to unseen subclasses by holding out subclasses as specified by several controllable parameters; similarly, NICO (He et al., 2020) considers subclasses that are defined by their context, such as dogs at home versus dogs on the beach; DeepFashion-Remixed (Hendrycks et al., 2020b) constrains the training set to include only photos from a single camera viewpoint and tests generalization to unseen camera viewpoints; BDD-Anomaly (Hendrycks et al., 2020a) uses a driving dataset but with all motorcycles, trains, and bicycles removed from the training set only; and ObjectNet (Barbu et al., 2019) comprises images taken from a few pre-specified viewpoints, allowing for systematic evaluation for robustness to camera angle changes but deviating from natural camera angles.

Distribution shifts across datasets. A well-studied special case of the above category is the class of distribution shifts obtained by combining several disparate datasets (Torralba and Efros, 2011), training on one or more of them and then testing on the remaining datasets. Many of these distribution shifts are constructed to be more drastic than might arise in the wild. For example, standard domain adaptation benchmarks include transfers across digit classification datasets such as MNIST and SVHN (LeCun et al., 1998; Yuval et al., 2011; Tzeng et al., 2017; Hoffman et al., 2018), as well as transfers across datasets containing different renditions (e.g., photos, clipart, sketches) in DomainNet (Peng et al., 2019) and the Office-Home dataset (Venkateswara et al., 2017).

The main difference between domain adaptation and domain generalization is that in the latter, we do not assume access to unlabeled data from the test distribution. This makes it straightforward to use domain adaptation benchmarks for domain generalization, e.g., in DomainBed (Gulrajani and Lopez-Paz, 2020); we focus on domain generalization in this work, but further discuss unsupervised domain adaptation in Section 7. Other similar benchmarks that have been proposed for domain generalization include VLCS (Fang et al., 2013), which tests generalization across similar visual object recognition datasets; PACS (Li et al., 2017a), which (like DomainNet) tests generalization across datasets with different renditions; and ImageNet-R (Hendrycks et al., 2020b), which also tests generalization across different renditions by collecting a separate dataset from Flickr.

3. Problem settings

WILDS presents a suite of benchmark datasets with distribution shifts, each of which can be cast as a domain generalization problem, a subpopulation shift problem, or a mixture of both. In this section, we formalize these two problem settings. We first discuss *domain shifts*, a broad class of distribution shifts addressed in both the domain generalization and subpopulation shift settings. We then define the two settings in terms of the nature of the domain shifts as well as the information available at training and test time.

Domain shifts. In domain shifts, we assume that the overall data distribution is a mixture of domains. Each domain $d \in \mathcal{D}$ corresponds to a fixed data distribution P_d, and $\mathcal{D} = \{1, \ldots, D\}$ is the set of all domains observed at training or test time. For example, in the CAMELYON17-WILDS dataset, the task is tumor classification, the domains are hospitals, and each hospital has a particular distribution over patient samples collected in that hospital. Concretely, we consider data distributions over (x, y, d) where x is the input, y is the label, and d is the domain; all points sampled from P_d have domain d.

As is standard, x is observed at both training and test time, and y is only observed at training time. Whether d is observed at training and/or test time depends on the problem setting; in this work, we focus on settings where the domain d is also observed at training time, though it may or may not be observed at test time.

The training distribution is a mixture of domains, with mixture weights q_d^{train} for each domain $d \in \mathcal{D}$,

$$P^{\mathsf{train}} = \sum_{d \in \mathcal{D}} q_d^{\mathsf{train}} P_d. \tag{1}$$

Analogously, the test distribution is a different mixture of domains with weights q_d^{test},

$$P^{\mathsf{test}} = \sum_{d \in \mathcal{D}} q_d^{\mathsf{test}} P_d. \tag{2}$$

For convenience, we define the set of training domains as $\mathcal{D}^{\mathsf{train}} = \{d \in \mathcal{D} \mid q_d^{\mathsf{train}} > 0\}$, and likewise, the set of test domains as $\mathcal{D}^{\mathsf{test}} = \{d \in \mathcal{D} \mid q_d^{\mathsf{test}} > 0\}$.

Domain generalization and subpopulation shift are both problem settings that involve domain shift; they primarily differ in the overlap between the training domains $\mathcal{D}^{\mathsf{train}}$ and the test domains $\mathcal{D}^{\mathsf{test}}$. We discuss other potential problem settings involving domain shift in Section 7.

Domain generalization. In the domain generalization setting, we seek to generalize to new domains that have not been seen at training time. Accordingly, we consider disjoint domains at training and test time, with $\mathcal{D}^{\mathsf{train}} \cap \mathcal{D}^{\mathsf{test}} = \emptyset$. While the specific evaluation metric varies from dataset to dataset, a typical goal is to train a model θ that achieves low average loss on the test distribution. This can be written as a weighted average of losses over the test domains $\mathcal{D}^{\mathsf{test}}$,

$$\mathbb{E}_{P^{\mathsf{test}}}\left[\ell(\theta; (x, y))\right] = \sum_{d \in \mathcal{D}^{\mathsf{test}}} q_d^{\mathsf{test}} \mathbb{E}_{P_d}\left[\ell(\theta; (x, y))\right], \tag{3}$$

where $\ell(\theta; (x, y))$ is the loss incurred by the model θ on the point (x, y).

At training time, we observe the input x, label y, and the domain d drawn from P^{train} (Equation (1)). For each test point drawn from P^{test} (Equation (1)), we observe its input x as well as its domain d.[3]

For example, in the CAMELYON17-WILDS dataset mentioned above, the goal is to generalize to unseen hospitals: we train on patient data from one set of hospitals, and then test on patient data from a different hospital that is not in the training set.

Subpopulation shifts. In the subpopulation shift setting, the goal is to perform well across a wide range of domains seen during training time. All test domains are seen during training time, with $\mathcal{D}^{\mathsf{test}} \subseteq \mathcal{D}^{\mathsf{test}}$, but the frequencies of the domains change, with $q^{\mathsf{train}} \neq q^{\mathsf{test}}$.[4] While subpopulation shift settings can target specific test distributions (e.g., a particular minority domain), we often consider a set of potential test distributions (e.g., any domain) instead, with the goal to perform well across all such distributions. In the latter case, a typical goal is to perform well on the worst-case test distribution, for example on the worst-case domain, and more precisely to minimize the worst-case loss,

$$\max_{d \in \mathcal{D}^{\mathsf{test}}} \mathbb{E}_{P_d}\left[\ell(\theta; (x, y))\right]. \tag{4}$$

We observe (x, y, d) at training time, but unlike the domain generalization setting, we observe only x at test time.

3. As the test domains $\mathcal{D}^{\mathsf{test}}$ are disjoint from the training domains $\mathcal{D}^{\mathsf{train}}$, seeing the domain information at test time (e.g., a single integer with the ID of the domain) is generally not helpful. In the domain generalization setting, we can equivalently assume that the test domain d is unobserved. Whether the test domain d is observed matters more in other problem settings where we might assume access to unlabeled test data at training time; see Section 7 for a discussion.
4. The term subpopulation shift sometimes also includes the setting where the training and test distributions comprise completely distinct subpopulations, e.g., in Santurkar et al. (2020). However, to distinguish the subpopulation shift and domain generalization settings, we focus here on subpopulation shifts where there is non-zero overlap between the training and test distributions.

For example, in the CivilComments-wilds dataset, the domains correspond to particular subpopulations defined by demographics, some of which are a minority in the training set, and we seek to have high performance on each of these subpopulations without observing their demographic identity d at test time.

Discussion. We associate each dataset in Wilds with the problem setting that we believe best reflects the real-world challenges in the corresponding application area. For example, domain generalization is a realistic setting for the Camelyon17-wilds dataset as medical models are typically trained on data collected from a handful of hospitals but with the goal of general deployment across different hospitals, including unseen ones. On the other hand, subpopulation shift is appropriate for the CivilComments-wilds dataset, as the real-world challenge is that some demographic subpopulations (domains) are underrepresented, rather than completely unseen, in the training data; this leads to poor model performance on those subpopulations.

Which problem setting is appropriate depends on many dataset-specific dactors, but some common considerations include:

- **Domain type.** Certain types of domains are generally more appropriate for a particular setting. For example, if the domains represent time, as in FMoW-wilds, then domain generalization is suitable as a common challenge is to generalize from past data to future data. On the other hand, if the domains represent demographics and the goal is to improve performance on minority subpopulations, as in CivilComments-wilds, then subpopulation shift is typically more appropriate.

- **Data collection challenges.** When collecting data from a new domain is expensive, domain generalization is often appropriate, as we might want to train on data from a limited number of domains but still generalize to unseen domains. For example, it is difficult to collect patient data from multiple hospitals, as in Camelyon17-wilds, or survey data from new countries, as in PovertyMap-wilds.

- **Continuous addition of new domains.** A special case of the above is when new domains are continuously created. For example, in Amazon-wilds, where domains correspond to users, new users are constantly signing up for the platform; and in iWildCam2020-wilds, where domains correspond to camera traps, new cameras are constantly being deployed. These are natural domain generalization settings.

The categories of domain generalization or subpopulation shift provide a general framework for thinking about domain shifts. In practice, it is not always possible to clearly define a problem as one or the other. For example, if a domain d is present in the training set but $q_d^{\mathsf{train}}/q_d^{\mathsf{test}}$ is very small, it might be technically subpopulation shift, but practically closer to domain generalization. Furthermore, some settings are a combination of domain generalization and subpopulation shift. For example, in the FMoW-wilds dataset, the input is satellite images, and the domains correspond to the year and the geographical region that a satellite image was taken. We study a distribution shift where there is domain generalization across time (i.e., the training set comprises images taken before a certain year, and the test set comprises images taken after a certain year) but a subpopulation shift across geographical regions (i.e., there are images from the same geographical regions in the training and test sets, but at different frequencies).

4. Baselines and experimental setup

We take the following approach for reporting baseline performance on each dataset. First, to see if there is a substantial performance gap between the in-distribution and out-of-distribution settings, we measure the performances of models trained using the standard approach of empirical risk minimization (i.e., minimizing average training loss). Second, we benchmark other training

algorithms for the appropriate problem setting (domain generalization or subpopulation shift), in order to see if the performance gaps are eliminated by those algorithms or if there is still room for improvement.

4.1 Empirical risk minimization (ERM)

Empirical risk minimization seeks models that minimize the average training loss

$$\mathcal{R}_{\mathsf{ERM}}(\theta) := \hat{\mathbb{E}}_{P^{\mathsf{train}}}\left[\ell(\theta;(x,y))\right]. \tag{5}$$

These models do not make use of any additional metadata (e.g., domain annotations) when training, although these metadata may be used for model selection. ERM is the de facto standard for training (state-of-the-art) ML models. However, as we note in Section 1, prior work has shown that the performance of such models can degrade substantially under distribution shift; for example, its focus on minimizing average loss can lead to high loss on minority subpopulations of the data (Blodgett et al., 2016; Tatman, 2017; Hashimoto et al., 2018).

4.2 Domain generalization baselines

Numerous methods have been proposed for domain generalization, including domain-invariant learning (Ganin et al., 2016; Sun and Saenko, 2016), invariant risk minimization (IRM) (Arjovsky et al., 2019), and meta-learning-based methods (Li et al., 2017b; Dou et al., 2019). Recently, Gulrajani and Lopez-Paz (2020) benchmarked many of these methods on standard domain generalization datasets and found that they all performed comparably to, and no better, than ERM. These methods all utilize training domain information. A common approach is to add a penalty term to the ERM objective that encourages some form of invariance across domains. We include two such penalty-based methods as representative baselines:

- **DeepCORAL** (Sun and Saenko, 2016), which penalizes differences in the means and covariances of the feature distributions (i.e., the distribution of the activations of the last layer in a neural network) for each domain. This method was originally proposed in the context of domain adaptation, where it was shown to substantially improve performance on standard domain adaptation benchmarks, and it has been subsequently adapted for domain generalization (Gulrajani and Lopez-Paz, 2020).
- **Invariant risk minimization (IRM)** (Arjovsky et al., 2019), which penalizes feature distributions that have different optimal linear classifiers for each domain.

Both of these methods are designed for models with featurizers, which first map each input to a feature representation and then predict based on the representation. We adapted their implementations from Gulrajani and Lopez-Paz (2020), with the following change. To estimate the feature distribution for a domain, these algorithms need to see a sufficient number of examples from that domain in a minibatch. However, some of our datasets have large numbers of domains, making it infeasible for each minibatch to contain examples from all domains. For these algorithms, our data loaders form a minibatch by first sampling a few domains, and then sampling examples from those domains.[5]

4.3 Subpopulation shift baselines

To train models robust to subpopulation shifts, we test the following approaches, both of which utilize training domain annotations:

5. The batch size and the number of domains per batch can be customized. For consistency in our experiments, we used the same total batch size for these algorithms and for ERM, with a default of 8 examples per domain in each minibatch (e.g., if the batch size was 32, then in each minibatch we would have 8 examples $\times$ 4 domains).

- **Reweighting** (Shimodaira, 2000). A simple but strong baseline is to train models with a reweighted objective that effectively upweights minority domains by treating each domain equally,

$$\mathcal{R}_{\mathsf{RW}}(\theta) := \frac{1}{|\mathcal{D}^{\mathsf{train}}|} \sum_{d \in \mathcal{D}^{\mathsf{train}}} \hat{\mathbb{E}}_{P_d} \left[\ell(\theta; (x, y)) \right]. \tag{6}$$

  This is a heuristic for achieving good performance on minority domains.

- **Group DRO** (Hu et al., 2018; Sagawa et al., 2020) uses distributionally robust optimization (DRO) to explicitly minimize the loss on the worst-case domain,

$$\mathcal{R}_{\mathsf{DRO}}(\theta) := \max_{d \in \mathcal{D}^{\mathsf{train}}} \hat{\mathbb{E}}_{P_d} \left[\ell(\theta; (x, y)) \right]. \tag{7}$$

Other methods for subpopulation shifts include more sophisticated reweighting methods (Cui et al., 2019); other DRO algorithms that do not make use of explicit domain information, instead relying on unsupervised clustering (Oren et al., 2019; Sohoni et al., 2020) or worst-case subpopulations across all domains (Duchi et al., 2019); adaptive Lipschitz regularization (Cao et al., 2020); slice-based learning (Chen et al., 2019b; Ré et al., 2019); and style transfer across domains (Goel et al., 2020).

4.4 Model selection

We run the baseline algorithms above on standard model architectures, e.g., ResNets and DenseNets for image classification datasets (He et al., 2016; Huang et al., 2017) or BERT for text datasets (Devlin et al., 2019).

In our settings, our goal is to select models that have high out-of-distribution (OOD) performance. To do so, we form a separate OOD validation set that exhibits a similar distribution shift to the tes set. For example, in the iWildCam2020-wilds dataset, where the domains are camera traps, we form the training set, OOD validation set, and test set from distinct sets of camera traps. We then grid search for model hyperparameters (e.g., weight decay or dropout) and algorithm hyperparameters (i.e., penalty weights for DeepCORAL and IRM) using the OOD validation set.

We provide further detail on model selection in Appendix A, and discuss the choice of in-distribution vs. out-of-distribution validation sets in Section 8.

5. Wilds datasets

We now discuss the 7 datasets in the Wilds benchmark, summarized in Table 1. For each dataset, we first describe the task, the distribution shift, and the evaluation criteria. We then present baseline results, demonstrating significant performance drops due to distribution shifts and benchmarking existing robust training algorithms. Finally, we discuss the real-world relevance of the problem setting as well as its connections to other distribution shifts reported in the literature. Because we modify the dataset from the original version in terms of the evaluation, splits, and data through substantial pre-processing, we use a -wilds suffix to avoid confusion between our modified datasets and their original sources.

5.1 FMoW-wilds: Building and land classification across different regions and years

Machine learning techniques on satellite imagery and other remotely sensed data can enable global-scale monitoring of sustainability and economic challenges, particularly in data-poor regions where gathering data on the ground is prohibitively expensive. Closing this data gap improves research and decision-making with respect to policy and humanitarian efforts in applications such as tracking deforestation (Hansen et al., 2013), population density mapping (Tiecke et al., 2017), poverty

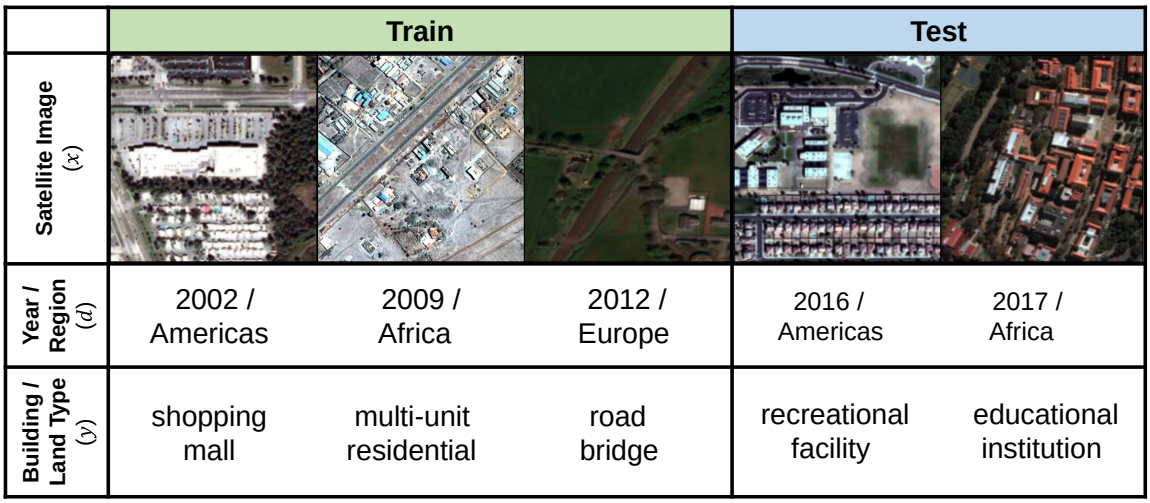

Figure 2: Examples from the FMoW-WILDS dataset. The training and test sets are split by time. We aim to generalize to satellite imagery taken in the future, which may be shifted due to infrastructure development across time, and to do equally well across geographic regions.

mapping (Abelson et al., 2014; Jean et al., 2016), crop yield prediction (Wang et al., 2020b), and other economic tracking applications (Katona et al., 2018). As human activity and environmental processes change the natural environment and human-made infrastructure, ML models on satellite imagery must be robust to distribution shifts over time. Disparities in data available between regions can also lead to performance disparities over spatial locations.

We study this problem through a variant of the Functional Map of the World dataset (Christie et al., 2018). Additional dataset and model details are in Appendix B.1.

5.1.1 Setup

Problem setting. The input x is an RGB satellite image and the label y is one of 62 building or land use categories. The domain d is defined on time and geographical regions. We aim to solve both a domain generalization problem across time and improve subpopulation performance across regions.

Data. FMoW-WILDS is based on the Functional Map of the World dataset (Christie et al., 2018), which collects and categorizes over 1 million high-resolution satellite images from over 200 countries based on the functional purpose of the buildings or land in the image, over the years 2002–2018 (see Figure 2). We split the data into three time range domains, 2002–2013, 2013–2016, and 2016–2018, as well as five geographical regions as subpopulations (Africa, Americas, Oceania, Asia, and Europe). For each example, we also provide the timestamp and location coordinate, though our baseline models only use the coarse time ranges and geographical regions instead of these additional metadata.

We use the following data splits:

1. **Training:** 76,863 images from the years 2002–2013.
2. **Validation (OOD):** 19,915 images from the years from 2013–2016.
3. **Test (OOD):** 22,108 images from the years from 2016–2018.
4. **Validation (ID):** 11,483 images from the years from 2002–2013.
5. **Test (ID):** 11,327 images from the years from 2002–2013.

	Validation (ID)	Validation (OOD)	Test (ID)	Test (OOD)
Average				
ERM	61.6 (0.21)	59.7 (0.14)	59.9 (0.27)	53.1 (0.25)
DeepCORAL	58.7 (0.44)	56.7 (0.06)	57.1 (0.15)	50.5 (0.30)
IRM	59.2 (0.31)	57.2 (0.01)	58.0 (0.25)	50.9 (0.32)
Worst-region				
ERM	59.3 (0.12)	48.2 (2.05)	57.7 (0.46)	31.7 (1.01)
DeepCORAL	57.0 (0.26)	46.8 (1.18)	55.4 (0.26)	30.5 (0.70)
IRM	57.0 (0.40)	47.4 (2.36)	56.6 (0.78)	31.0 (1.15)

Table 2: Time shift and worst-region accuracies (%) for models trained on data before 2013 and tested on held-out locations from in-distribution (ID) or out-of-distribution (OOD) test sets in FMoW-wilds. The models are early-stopped with respect to OOD validation accuracy. Standard deviations over 3 trials are in parentheses.

		Test (ID)		Test (OOD)		
	Algorithm	Average	Worst-region	Average	Last year	Worst-region
Standard split (ID examples)	ERM	59.7 (0.14)	57.7 (0.46)	53.1 (0.25)	48.3 (0.60)	31.7 (1.01)
Mixed split (ID + OOD examples)	ERM	59.3 (0.26)	57.2 (0.10)	57.6 (0.44)	54.4 (0.91)	48.3 (0.40)

Table 3: Performance drops for ERM models on FMoW-wilds. In the standard split, we train on data from 2002–2013, whereas in the mixed split, we train on the same amount of data but half from 2002–2013 and half from 2013–2018. In both cases, we test on data from 2016–2018. Models trained on the standard split degrade in performance under the time shift, especially on the last year (2017) of the test data, and also fare poorly on the subpopulation shift, with low worst-region accuracy. Models trained on the mixed split have higher OOD average and last year accuracy and much higher OOD worst-region accuracy.

The in-distribution (ID) splits roughly follow the train, validation, and test splits from the original dataset, which did not consider distribution shifts; see Appendix B.1.

The data splits contain images from disjoint location coordinates, and all splits contain data from all 5 geographic regions. There is a disparity in the number of examples in each region, with Africa and Oceania having the least examples (Figure 3); this could be due to bias in sampling and/or a lack of infrastructure and land data in certain regions.

Evaluation. We evaluate models by their average and worst-region OOD accuracies. The former measures the ability of the model to generalize across time, while the latter additionally measures how well models do across different regions/subpopulations under a time shift.

	Asia	Europe	Africa	Americas	Oceania	Worst region
OOD Test						
ERM	55.5 (0.13)	55.3 (0.11)	31.7 (1.01)	56.3 (0.36)	60.0 (1.32)	31.7
DeepCORAL	52.3 (0.40)	53.0 (0.79)	30.5 (0.70)	53.4 (0.70)	57.6 (1.63)	30.5
IRM	53.3 (0.64)	54.1 (0.40)	31.0 (1.15)	53.1 (0.40)	55.9 (2.13)	31.0
ID Test						
ERM	59.1 (0.57)	57.7 (0.46)	67.4 (0.71)	62.7 (0.76)	71.8 (2.38)	57.7
DeepCORAL	56.5 (0.53)	55.4 (0.26)	68.0 (3.13)	58.9 (0.42)	70.0 (3.24)	55.4
IRM	56.6 (0.71)	56.6 (0.78)	70.2 (0.23)	60.1 (1.22)	69.1 (0.72)	56.6

Table 4: Region shift results (accuracy, %) for models trained on data before 2013 and tested on held-out locations from ID (< 2013) or OOD (≥ 2016) test sets in FMoW-wilds. One STD shown in parentheses.

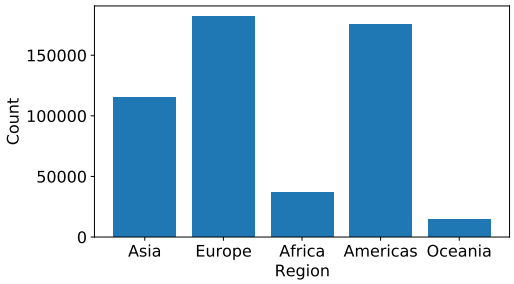

Figure 3: Number of examples from each region of the world in FMoW-wilds. There is much less data from Africa and Oceania than other regions.

Potential leverage. FMoW-wilds considers both domain generalization across time and subpopulation shift across regions. We may leverage structure across both space and time improve robustness to these shifts. One hypothesis is that infrastructure development occurs smoothly over time. Utilizing this gradual shift structure with the timestamp metadata may enable adaptation across longer time periods (Kumar et al., 2020).

As with time, the distribution may shift smoothly over spatial locations, and enforcing some consistency with respect to spatial structure may improve predictions (Rolf et al., 2020; Jean et al., 2018). On a region level, developing countries with less labeled data may present greater robustness challenges. One potential source of leverage for mitigating this is to use knowledge of other developing countries with similar economies from which we can transfer some knowledge. The location coordinate metadata allows for transfer learning across similar locations at any spatial scale.

5.1.2 Baseline results

ERM results and performance drops. Following Christie et al. (2018), we train a Densenet-121 model pretrained on ImageNet to minimize the cross entropy loss on data from the training split (2002–2013). Table 3 shows that accuracy drops almost 7% when evaluated on OOD test set ($\geq$ 2016) vs. the ID test set, and that the accuracy drop is especially large (11%) on images from the last year of the dataset (2017), furthest in the future from the training set. In addition, there is a substantial 26% drop in worst-region accuracy, with the model performing much worse in Africa than other regions (Table 4).

We ran an additional experiment where we mixed in some data from the OOD period (2013–2018) into the training set, while keeping the overall training set size constant. A model trained on this mixed split had a much smaller drop in performance under the time and region shifts (Table 3). This comparison implies that the performance drop between the ID and OOD test sets is largely due to the distribution shift across time and region.

Additional baseline methods. We compare against two domain generalization methods, DeepCORAL and IRM, using examples from different years as distinct domains. Table 2 shows that both of the methods actually make ID and OOD test performance worse than ERM, and that the worst-region accuracy is comparable to ERM. These existing domain generalization methods do not effectively improve robustness to shift across time.

Discussion. Intriguingly, a large subpopulation shift across regions only occurs with a combination of time and region shift. This is corroborated by the oracle region shift results (Table 4), which do not have time shift between training and test and do not display a large drop in performance across regions. As we show in Appendix B.1, we find that there is a large label distribution shift between non-African regions and Africa, suggesting that the drop in performance may be in some part due to label shift.

Despite having the smallest number of training examples (Figure 3), the baseline models do not suffer a drop in performance in Oceania on validation or test sets (Table 4). We hypothesize that infrastructure in Oceania is more similar to regions with a large amount of data than Africa. In contrast, Africa may be more distinct and may have changed more drastically over 2002-2018, the time extent of the dataset. This suggests that the subpopulation shift is not merely a function of the number of training examples.

5.1.3 Broader context

Recognizing infrastructure and land features is crucial to many remote sensing applications. For example, in crop land prediction Wang et al. (2020b), recognizing gridded plot lines, plot circles, farm houses, and other visible features are important in recognizing crop fields. However, farming practices and equipment evolve over time and vary widely across the world, requiring both robust object recognition and synthesis of their different usage patterns.

Although the data is typically limited, we desire generalization on a global scale without requiring frequent large-scale efforts to gather more ground-truth data. It is natural to have labeled data with limited temporal or spatial extent since ground truth generally must be verified on the ground or requires manual annotations from domain experts (i.e. often hard to be crowdsourced). A number of existing remote sensing datasets have limited spatial or temporal scope, including the UC Merced Land Use Dataset (Yang and Newsam, 2010), TorontoCity (Wang et al., 2017), and SpaceNet (DigitalGlobe and Works, 2016). However, works based on these datasets generally do not systematically study shifts in time or location.

5.2 PovertyMap-wilds: Poverty mapping across different countries

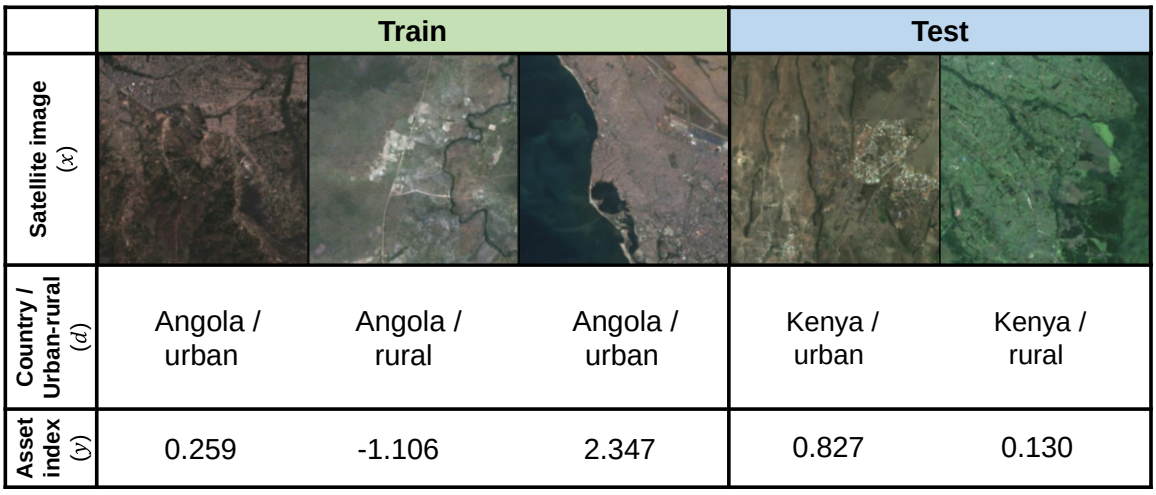

Figure 4: Examples from the PovertyMap-wilds dataset. Training and test splits are split by countries, where we aim to generalize to unseen countries. There may be significant economic and cultural differences across country borders that contribute to the spatial distribution shift.

High-resolution predictions of poverty measures are essential for targeted humanitarian efforts and directing policy decisions in developing countries (Espey et al., 2015; Abelson et al., 2014). However, ground truth measurements of poverty are lacking for much of the developing world, since gathering them requires conducting expensive surveys in the field (Blumenstock et al., 2015; Xie et al., 2016; Jean et al., 2016; Yeh et al., 2020). At least 4 years pass between nationally representative consumption or asset wealth surveys in the majority of African countries, with seven countries that

had either never conducted a survey or had gaps of over a decade between surveys (Yeh et al., 2020). The lack of labels in certain countries creates a natural scenario where we desire model generalization to unseen countries. In addition, we consider the effect of shift across countries on model performance on rural vs. urban subpopulations. Improving performance within the rural subpopulation is especially important in developing African countries, where most of the poorest villages are rural.

We study this problem through a variant of poverty mapping dataset collected by Yeh et al. (2020). Additional dataset and model details are in Appendix B.2.

5.2.1 Setup

Problem setting. The input x is a multispectral LandSat satellite image and the output y is a real-valued asset wealth index computed from survey data (Demographic and Health Surveys (DHS)). The domain d is defined on countries (geographic borders) and urban/rural areas. We aim to solve both a domain generalization problem across country borders and improve subpopulation performance across urban and rural areas.

Data. PovertyMap-wilds is based on a dataset collected by Yeh et al. (2020), which assembles satellite imagery and survey data at 19,669 villages from 23 African countries between 2009 and 2016 (Figure 4). There are 23 domains (countries) and every location is classified as either urban or rural. Each example comes with location coordinates, the survey year, and urban/rural classification. On top of LandSat images, we follow Yeh et al. (2020) and append an additional eighth image channel for nighttime light intensity from a separate satellite, which correlates with poverty measures (Noor et al., 2008; Elvidge et al., 2009). We use 5 folds of the dataset, where each fold defines a different set of OOD countries. In each fold, we use the following splits of the data (the number of countries and images in each split varies slightly from fold to fold):

1. **Training:** ∼10000 images from 13–14 countries.
2. **Validation (OOD):** ∼4000 images from 4–5 different countries (distinct from training and test (OOD) countries).
3. **Test (OOD):** ∼4000 images from 4–5 different countries (distinct from training and validation (OOD) countries).
4. **Validation (ID):** ∼1000 images from the same 13–14 countries in the training set.
5. **Test (ID):** ∼1000 images from the same 13–14 countries in the training set.

All splits contain images of both urban and rural locations.

Evaluation. As is standard in the literature (Jean et al., 2016; Yeh et al., 2020), the models are evaluated on squared Pearson correlation (r^2) on the OOD test set for shift across countries and also the r^2 on the rural subpopulation in the OOD test set for shift across urban/rural areas. We report the latter as previous works on poverty prediction from satellite imagery have noted that a significant part of model performance relies on distinguishing urban vs. rural areas, and improving performance within these subpopulations is an ongoing challenge (Jean et al., 2016; Yeh et al., 2020).

Potential leverage. Given large shifts across countries, is it possible to generalize across borders? We note that some indicators of wealth are known to be robust and are able to be seen from space. For example, roof type (e.g. thatched or metal roofing) has been shown to be a reliable proxy for wealth (Abelson et al., 2014), and context factors such as nearby cropland health, presence of paved roads, and connections to urban areas are plausibly reliable signals for measuring poverty. Poverty measures are known to be highly correlated across space, meaning nearby villages will likely have similar poverty measures, and methods can utilize this spatial structure (using the provided location coordinate metadata) to improve predictions (Jean et al., 2018; Rolf et al., 2020).

	Validation (ID)	Validation (OOD)	Test (ID)	Test (OOD)
Overall				
ERM	0.67 (0.02)	0.65 (0.06)	0.68 (0.03)	0.62 (0.05)
DeepCORAL	0.65 (0.04)	0.66 (0.07)	0.68 (0.03)	0.61 (0.07)
IRM	0.67 (0.03)	0.65 (0.07)	0.68 (0.05)	0.61 (0.04)
Rural subpopulation				
ERM	0.32 (0.05)	0.26 (0.06)	0.34 (0.08)	0.22 (0.07)
DeepCORAL	0.28 (0.07)	0.27 (0.10)	0.29 (0.10)	0.20 (0.08)
IRM	0.31 (0.07)	0.29 (0.05)	0.32 (0.07)	0.24 (0.06)

Table 5: Squared Pearson correlation r^2 (higher is better) on in-distribution and out-of-distribution (unseen countries) held-out sets in PovertyMap-wilds, including results on rural or urban subpopulations. All results are averaged over 5 different OOD country folds taken from Yeh et al. (2020), with standard deviations across different folds in parentheses. All models are early-stopped with respect to OOD validation MSE.

	Test (ID)			Test (OOD)		
	Overall r^2	Rural r^2	Urban r^2	Overall r^2	Rural r^2	Urban r^2
Standard split (ID examples)	0.68 (0.03)	0.34 (0.08)	0.42 (0.05)	0.62 (0.05)	0.22 (0.07)	0.35 (0.10)
Mixed split (ID + OOD examples)	0.69 (0.01)	0.35 (0.07)	0.43 (0.03)	0.69 (0.06)	0.35 (0.07)	0.43 (0.07)

Table 6: Performance drops for ERM models on PovertyMap-wilds. In the standard split, we train on data from one set of countries, and then test on a different set of countries. In the mixed split, we train on the same amount of data but sampled uniformly from all countries. Models trained on the standard split degrade in performance, especially on rural subpopulations, while models trained on the mixed split do not.

ERM results and performance drops. Following Yeh et al. (2020), we trained a ResNet-18 to minimize squared error. When shifting across country borders, Table 6 shows that ERM suffers a 0.06 drop in r^2 on OOD test examples compared to ID test examples (MSE results are in Appendix B.2).[6] Moreover, the drop in performance is exacerbated when looking at urban and rural subpopulations, even though all splits contain urban and rural examples. Table 6 shows that the difference between ID and OOD test r^2 in the ERM model doubles from 0.06 to 0.12 when considering the overall vs. rural r^2. The spatial shift also exacerbates the gap between urban and rural performances, which grows from 0.08 to 0.13 between ID and OOD test in the ERM model.

We ran an additional experiment where we considered an alternative training set with data that was uniformly sampled from all countries, while keeping the overall training set size constant (i.e., compared to the standard training set, it has fewer examples from each country, but data from more countries). A model trained on this mixed split had a much smaller drop in performance between the ID and OOD test sets (Table 6), as the mixed training set contains examples from the countries in both the ID and OOD test sets. This comparison implies that the performance drop between the ID and OOD test sets is largely due to the distribution shift from seen to unseen countries.

Additional baseline methods. We trained models with DeepCORAL and IRM, taking examples from different countries as coming from distinct domains. Table 5 shows that these baselines are comparable to ERM. These existing domain generalization methods do not effectively improve robustness to shift across countries and urban/rural areas.

Discussion. These results corroborate performance drops seen in previous out-of-country generalization tests for poverty prediction from satellite imagery (Jean et al., 2016). In general, differences

6. Unlike other Wilds datasets, the standard deviations here are computed with models trained on 5 different data folds rather than only over different model initializations, resulting in larger standard deviations due to inherent differences between folds.

in infrastructure, economic development, agricultural practices, and even cultural differences can cause large shifts across country borders. Differences between urban and rural subpopulations have also been well-documented (Jean et al., 2016; Yeh et al., 2020). Models based on nighttime light information could suffer more in rural areas where nighttime light intensity is uniformly low or even zero.

Since survey years are also available, we could also investigate the robustness of the model over time. This would enable the models to be used for a longer time before needing more updated survey data, and we leave this to future work. Yeh et al. (2020) investigated predicting the change in asset wealth for individual villages in the World Bank Living Standards Measurement Surveys (LSMS), which is a longitudinal study containing multiple samples from the same village, finding that the village level task is relatively difficult ($r^2 = 0.35$) while aggregate predictions at the district level are more promising ($r^2 = 0.51$). Instead of predicting a time-series at each village, we can also consider shifts across years for cross-sectional samples such as in DHS using PovertyMap-wilds, which we leave to future work.

5.2.2 Broader context

Computational sustainability applications in the developing world includes not only poverty mapping but also tracking child mortality (Osgood-Zimmerman et al., 2018; Reiner et al., 2018; Burke et al., 2016), educational attainment (Graetz et al., 2018), food security and crop yield prediction (Wang et al., 2020b; You et al., 2017; Xie et al., 2020). Remote sensing data and satellite imagery has the potential to enable high-resolution maps of many of these sustainability challenges, but similarly to poverty, ground truth labels in these applications come from expensive surveys or observations from human workers in the field. Some prior works consider using spatial structure (Rolf et al., 2020; Jean et al., 2018), unlabeled data (Xie et al., 2016; Jean et al., 2018; Xie et al., 2020), weak sources of supervision (Wang et al., 2020b) to improve models globally despite the lack of ground-truth data. We hope that PovertyMap-wilds can be used to improve the robustness of machine learning techniques on satellite data, providing an avenue for cheaper and faster measurements that can be used to make progress on a general set of computational sustainability challenges.

5.3 iWildCam2020-wilds: Species classification across different camera traps

In the 2020 Living Planet Report (Grooten et al., 2020), the WWF found that animal populations have declined 68% on average since 1970. In the current climate crisis, understanding the connection between climate change, human impact, and wildlife biodiversity loss is more important than ever. Camera traps, heat or motion activated static cameras placed in the wild, are one of the primary methods for monitoring wildlife in the ecology community Wearn and Glover-Kapfer (2017). Camera traps collect data much faster than experts can process it, and so the ecologists have turned to computer vision as an efficient solution (Ahumada et al., 2020; Weinstein, 2018; Norouzzadeh et al., 2019; Tabak et al., 2019; Beery et al., 2019). However, camera traps, and in fact all static sensors, capture signals that are correlated in time and space. This correlation results in overfitting and poor generalization to new sensor deployments, reducing the scalability of computer vision solutions (Beery et al., 2018).

We study this problem using a variant of the iWildCam 2020 Competition Dataset (Beery et al., 2020a). Additional dataset and model details are in Appendix B.3.

5.3.1 Setup

Problem setting. We consider the domain generalization setting, where the domains are camera traps, and our goal is to learn models that generalize to photos taken from new camera deployment locations (Figure 5). The task is multi-class species classification. Concretely, the input x is a photo

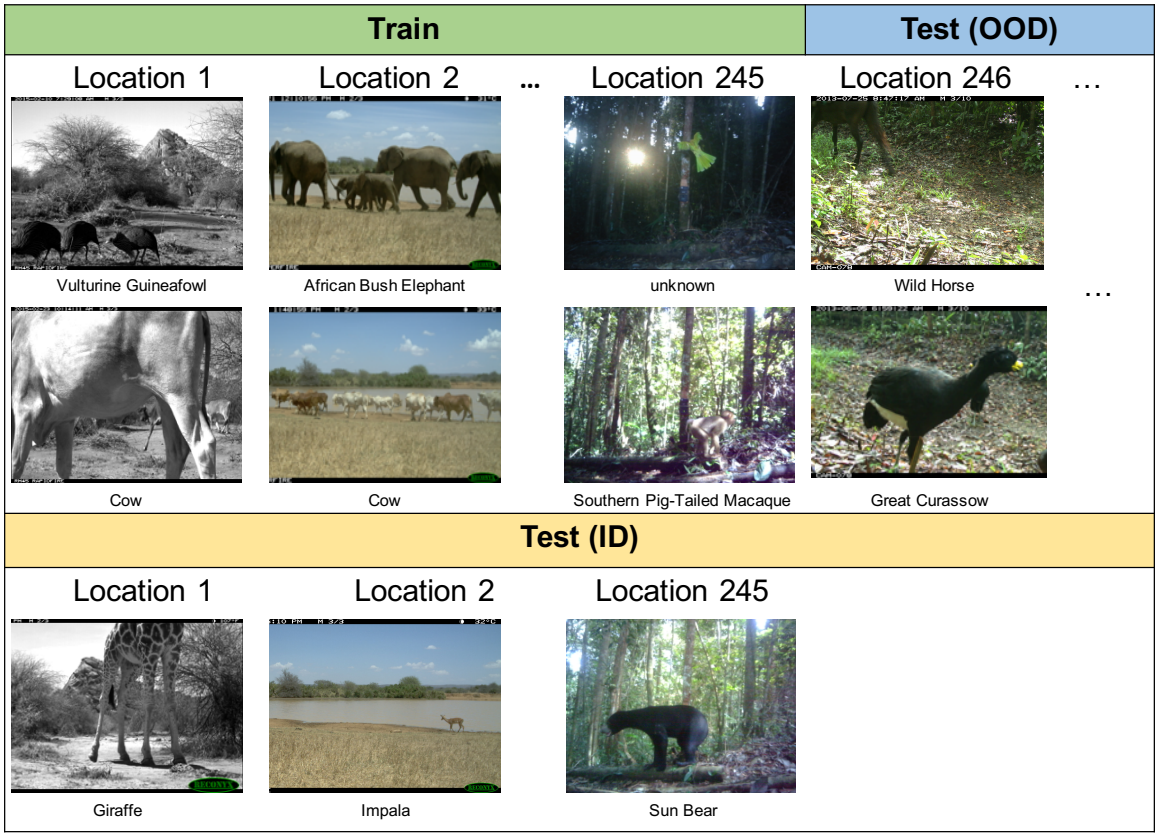

Figure 5: The iWildCam2020-wilds dataset comprises photos of wildlife taken by a variety of camera traps.

taken by a camera trap, the label y is one of 186 different classes, corresponding to animal species, and the domain d is an integer that identifies the camera trap that took the photo.

The training set contains 217, 959 images from 441 locations, and the test set contains 62, 894 images from 111 locations. These 552 locations are spread across 12 countries in different parts of the world. Each image is associated with a location ID so that images from the same location can be linked. As is typical for camera traps, approximately 50% of the total number of images are empty (this varies per location)

Data. The dataset comprises 217,609 images from 324 different camera traps spread across 12 countries in different parts of the world. The original camera trap data comes from the Wildlife Conservation Society.[7] Approximately half of the images do not contain any animal species; this corresponds to one of the 186 class labels.

We split the dataset by randomly partitioning the data by camera traps:

1. **Training:** 142,202 images taken by 245 camera traps.
2. **Validation (OOD):** 20,784 images taken by 32 different camera traps.
3. **Test (OOD):** 38,943 images taken by 47 different camera traps.

7. http://lila.science/datasets/wcscameratraps

Algorithm	Test (ID) Macro F1	Test (ID) Average accuracy	Test (OOD) Macro F1	Test (OOD) Average accuracy
ERM	**82.5** (1.3)	**96.5** (0.2)	**27.8** (1.3)	**62.9** (0.5)
DeepCORAL	68.3 (8.8)	93.1 (2.2)	26.3 (1.4)	62.5 (1.7))

Table 7: Baseline results on iWildCam2020-wilds.

4. **Validation (ID):** 7,819 images taken by the same camera traps as the training set, but on different dates.

5. **Test (ID):** 7,861 images taken by the same camera traps as the training set, but on different dates.

The camera traps were randomly distributed across the training, validation (OOD), and test (OOD) sets.

The original iWildCam 2020 Kaggle competition similarly split the dataset by camera trap, though the competition focused on average accuracy. We consider a smaller subset of the data here; see Appendix B.3.

Evaluation. We evaluate models by their macro F1 score (i.e., we compute the F1 score for each class separately, and then average those scores). We also report the average accuracy of each model across all test images, but primarily use the macro F1 score to better capture model performance on rare species. In the natural world, protected and endangered species are rare by definition, and are often the most important to accurately monitor. However, common species are much more likely to be captured in camera trap images, leading to a vast imbalance in data representation between rare and common species; this imbalance can make metrics like average accuracy an inaccurate picture of model effectiveness.

Potential leverage. Though the problem is challenging for existing ML algorithms, adapting to photos from different camera traps adaptation is simple and intuitive for humans. Repeated backgrounds and habitual animals, which cause each sensor to have a unique class distribution, provide a strong implicit signal across data from any one location. We anticipate that approaches that utilize the provided camera trap annotations can learn to factor out these common features and avoid learning spurious correlations between particular backgrounds and animal species.

5.3.2 Baseline results

ERM results and performance drops. We trained a ResNet-50 (He et al., 2016) that was pretrained on Imagenet. Model performance dropped substantially and consistently going from in-distribution (ID) to out-of-distribution (OOD) camera traps (Table 7), with a macro F1 score of 82.5 on the ID test set but only 27.8 on the OOD test set. Similarly, the model obtained an average accuracy of 96.5% on the ID test set but only 62.9% on the OOD test set. The large discrepancy between ID and OOD model performance suggests that there is significant room for improvement.

Additional baseline methods. We trained a DeepCORAL model, treating each camera trap as a domain. However, it did not improve upon the ERM baseline, with a macro F1 score of 26.3. We observed that as the F1 score on the OOD validation set tends to converge more quickly than on the ID validation set, early stopping on OOD validation sometimes leads to selecting an early epoch, which in turn leads to low scores on the ID validation and test sets. This occurred for one random seed out of three in the DeepCORAL models, which partially contributes to its lower ID accuracy and F1 scores.

Discussion. Even though there is significant label imbalance, the overall label distribution is approximately the same in the ID and OOD split, suggesting that it is not primarily label shift that

accounts for the performance drop. Across locations, there is drastic variation in illumination, camera angle, and background, vegetation, and color. This variation, coupled with considerable differences in the distribution of animals between camera traps, likely encourages the model to overfit to specific animal species appearing in specific locations, which may account for the performance drop.

The original iWildCam 2020 competition allows users to use MegaDetector (Beery et al., 2019), which is an animal detector trained on a large set of data beyond what is provided in the training set. To facilitate more controlled experiments, we intentionally do not use MegaDetector in iWildCam2020-wilds.

5.3.3 Broader context

Differences across data distributions at different sensor locations is a common challenge in automated wildlife monitoring applications, including using audio sensors to monitor animals that are easier heard than seen such as primates, birds, and marine mammals (Crunchant et al., 2020; Stowell et al., 2019; Shiu et al., 2020), and using static sonar to count fish underwater to help maintain sustainable fishing industries (Pipal et al., 2012; Vatnehol et al., 2018; Schneider and Zhuang, 2020). As with camera traps, each static audio sensor has a specific species distribution as well as a sensor specific background noise signature, making generalization to new sensors challenging. Similarly, static sonar used to measure fish escapement have sensor-specific background reflectance based on the shape of the river bottom. Moreover, since species are distributed in a non-uniform and long-tailed fashion across the globe, it is incredibly challenging to collect sufficient samples for rare species to escape the low-data regime. Implicitly representing camera-specific distributions and background features in per-camera memory banks and extracting relevant information from these via attention has been shown to help overcome some of these challenges for static cameras Beery et al. (2020b).

More broadly, shifts in background, image illumination and viewpoint have been studied in computer vision research. First, several works have shown that object classifiers often rely on the background rather than the object to make its classification (Rosenfeld et al., 2018; Shetty et al., 2019; Xiao et al., 2020). Second, common perturbations such as blurriness or shifts in illumination, tend to reduce performance (Dodge and Karam, 2017; Temel et al., 2018; Hendrycks and Dietterich, 2019). Finally, shifts in rotation and viewpoint of the object has been shown to degrade performance (Barbu et al., 2019).

5.4 Camelyon17-wilds: Tumor identification across different hospitals

Models for medical applications are often trained on data from a small number of hospitals, but with the goal of being deployed more generally to other hospitals. The challenge is that variation in data collection and processing across different hospitals can degrade model accuracy on data from hospitals not included in the training set (e.g., Zech et al. (2018); AlBadawy et al. (2018)). In histopathology applications—studying tissue slides under a microscope—this variation can arise from sources like differences in the patient population or in slide staining and image acquisition (Veta et al., 2016; Komura and Ishikawa, 2018; Tellez et al., 2019).

We study this distribution shift by constructing a patch-based variant of the Camelyon17 dataset (Bandi et al., 2018). Additional dataset and model details are in Appendix B.4.

5.4.1 Setup

Problem setting. We consider the domain generalization setting, where the domains are hospitals, and our goal is to learn models that generalize to data from a hospital that is not in the training set (Figure 6). The task is to predict if a given region of tissue contains any tumor tissue, which we model as binary classification. Concretely, the input x is a 96x96 histopathological image, the label y is a binary indicator of whether the central 32x32 region contains any tumor tissue, and the domain d is an integer that identifies the hospital that the patch was taken from.

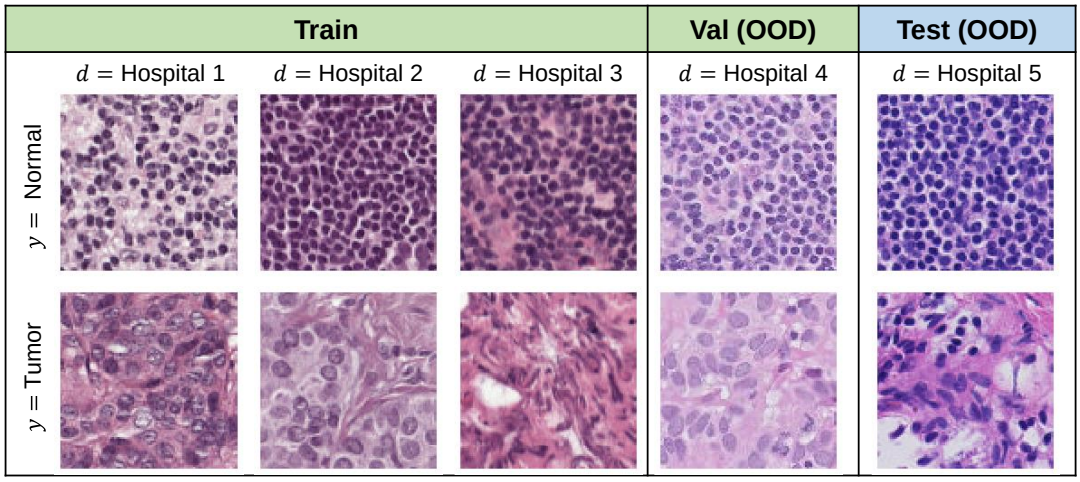

Figure 6: Sample patches from each hospital in CAMELYON17-WILDS. Each column contains two patches, one of normal tissue and the other of tumor tissue, from the same slide.

Data. The dataset comprises 450,000 patches extracted from 50 whole-slide images (WSIs) of breast cancer metastases in lymph node sections, with 10 WSIs from each of 5 hospitals in the Netherlands. Each WSI was manually annotated with tumor regions by pathologists, and the resulting segmentation masks were used to determine the labels for each patch. We also provide metadata on which slide (WSI) each patch was taken from, though our baseline algorithms do not use this metadata.

We split the dataset by domain (i.e., which hospital the patches were taken from):

1. **Training:** 302,436 patches taken from 30 WSIs, with 10 WSIs from each of the 3 hospitals in the training set.
2. **Validation (OOD):** 34,904 patches taken from 10 WSIs from a 4th hospital. These WSIs are distinct from those in the other splits.
3. **Test (OOD):** 85,054 patches taken from 10 WSIs from the 5th hospital, which was chosen because its patches were the most visually distinctive. These WSIs are also distinct from those in the other splits.
4. **Validation (ID):** 33,560 patches taken from the same 30 WSIs from the training hospitals.

The original dataset (Bandi et al., 2018) did not consider distribution shifts; the original training and test splits had all 5 hospitals, and the task was also different (see Appendix B.4).

Evaluation. We evaluate models by their average test accuracy across patches. Histopathology datasets can be unwieldy for ML models, as individual images can be several gigabytes large; extracting patches involves many design choices; the classes are typically very unbalanced; and evaluation often relies on more complex slide-level measures such as the free-response receiver operating characteristic (FROC) (Gurcan et al., 2009). To improve accessibility, we pre-process the slides into patches and balance the dataset so that each split has a 50/50 class balance, making average accuracy is a reasonable measure of performance (Veeling et al., 2018; Tellez et al., 2019).

Potential leverage. Prior work has shown that differences in staining between hospitals are the primary source of variation in this dataset, and that specialized stain augmentation methods can close the in- and out-of-distribution accuracy gap on a variant of the dataset based on the same

Algorithm	Validation (ID) accuracy	Validation (OOD) accuracy	Test (OOD) accuracy
ERM	97.8 (0.4)	84.3 (2.1)	**73.3** (9.9)
DeepCORAL	97.1 (0.8)	86.3 (2.2)	59.2 (15.1)
IRM	97.6 (0.4)	86.2 (2.2)	60.9 (15.3)

Table 8: Baseline results on Camelyon17-wilds. Parentheses show standard deviation across 10 replicates.

	Algorithm	Test accuracy
Standard split	ERM	74.1 (10.0)
Mixed split (oracle)	ERM	**90.4** (3.4)

Table 9: Performance drops for ERM models on Camelyon17-wilds. In the standard split, we train on data from three hospitals and evaluate on a different test hospital, whereas in the mixed split, we add data from one extra slide from the test hospital to the training set. The original test set has data from 10 slides; here, we report performance for both splits on 9 slides (without the slide that was moved to the training set). This makes the numbers (74.1 vs. 73.3) for the standard split slightly different from Table 8. Parentheses show standard deviation across 10 replicates.

underlying slides (Tellez et al., 2019). However, the general task of learning histopathological models that are robust to variation across hospitals (from staining and other sources) is still an open research question. In this way, the Camelyon17-wilds dataset is a controlled testbed for general-purpose methods that can learn to be robust to stain variation between hospitals, given a training set from multiple hospitals.

5.4.2 Baseline results

ERM results and performance drops. We trained a standard DenseNet-121 (Huang et al., 2017) from scratch with ERM, following prior work (Veeling et al., 2018). Across different hyperparameters (learning rate, L_2 regularization, and random seeds), Table 8 shows that this model was consistently accurate on the in-distribution (ID) validation set and to a lesser extent on the out-of-distribution (OOD) validation set, which was from a held-out hospital. However, it was wildly inconsistent on the test set, which was from a different held-out hospital, with a standard deviation of 9.9% in accuracies across 10 random seeds. There is a large gap between ID validation and OOD validation accuracy, and between OOD validation and OOD test accuracy. Nevertheless, we found that using the OOD validation set gave better results than using the ID validation set; see Appendix B.4 for more discussion.

We ran an additional experiment where we moved 1 of the 10 slides[8] from the test hospital to the training set and tested on the patches from the remaining 9 slides. The resulting oracle model consistently gets much higher accuracy on the reduced test set (Table 9), further suggesting that the observed performance drop is due to the distribution shift, as opposed to the intrinsic difficulty of the examples from the test hospital.

Additional baseline methods. We trained DeepCORAL and IRM models, treating each hospital as a domain. However, they did not improve upon the ERM baseline, with the our grid search

8. This slide was randomly chosen and corresponded to about 6% of the test patches; some slides contribute more patches than others because they contain larger tumor regions.

selecting the lowest values of their penalty weights (0.1 and 1, respectively) based on OOD validation accuracy.

Discussion. These results demonstrate a subtle failure mode when considering out-of-distribution accuracy: there are models (i.e., choices of hyperparameters and random seeds) that do well both in- and out-of-distribution, but we cannot reliably choose these models from just the training/validation set. We discuss this problem of underspecification (D'Amour et al., 2020) and generalization instability further in Section 8.

Due to the substantial variability in test accuracy on Camelyon17-wilds, we ask researchers to submit leaderboard submissions with results from 10 random seeds, instead of the 3 random seeds required for other datasets. Furthermore, to benchmark general-purpose robust learning algorithms and avoid specialized methods for dealing with stain variation, we ask that researchers do not use color-specific techniques (e.g., color augmentation) and also train their models from scratch, instead of fine-tuning models that are pre-trained from ImageNet or other datasets.

5.4.3 Broader context

Beyond histopathology applications, variation between different hospitals and deployment sites has also been shown to degrade model accuracy in other medical applications such as diabetic retinopathy (Beede et al., 2020) and chest radiographs (Zech et al., 2018; Phillips et al., 2020), including recent work on COVID-19 detection (DeGrave et al., 2020). Even within the same hospital, process variables like which scanner/technician took the image can significantly change model predictions (Badgeley et al., 2019).

In these medical applications, the gold standard is to evaluate models on an independent test set collected from a different hospital (e.g.,Beck et al. (2011); Liu et al. (2017); Courtiol et al. (2019); McKinney et al. (2020)) or at least with a different scanner within the same hospital (e.g., Campanella et al. (2019)). However, this practice has not been ubiquitous due to the difficulty of obtaining data spanning multiple hospitals (Esteva et al., 2017; Bejnordi et al., 2017; Codella et al., 2019; Veta et al., 2019). The baseline results reported above show that even evaluating on a single different hospital might be insufficient, as results can vary widely between different hospitals (e.g., between the validation and test OOD datasets). We hope that the Camelyon17-wilds dataset, which has multiple hospitals in the training set and independent hospitals in the validation and test sets, will be useful for developing models that can generalize reliably to new hospitals and contexts (Chen et al., 2020).

5.5 OGB-MolPCBA: Molecular property prediction across different scaffolds

Drug discovery is a time-consuming procedure; the entire process typically takes more than 10 years, during which many expensive wet lab experiments need to be performed to find a potent molecule (Hughes et al., 2011). Accurate and generalizable molecular property predictor is useful for virtual screening (Shoichet, 2004), i.e., computer-aided search over a huge collection of small molecules in order to identify those structures which are most likely to bind to a drug target. Once the candidate molecules are identified, those molecules are further scrutinized via lab experiments (Hughes et al., 2011). Accurate virtual screening can significantly reduce redundant experiments and thereby accelerate the drug discovery process. The challenge is that molecular properties need to be predicted over diverse sets of molecules curated in the large chemical database (e.g., ZINC (Sterling and Irwin, 2015)). It is, therefore, critical for models to generalize to out-of-distribution molecules that are structurally very different from training ones.

We study this issue through the OGB-MolPCBA dataset, which is directly adopted from the Open Graph Benchmark (Hu et al., 2020b) and originally curated by the MoleculeNet (Wu et al., 2018). Additional dataset and model details are in Appendix B.5.

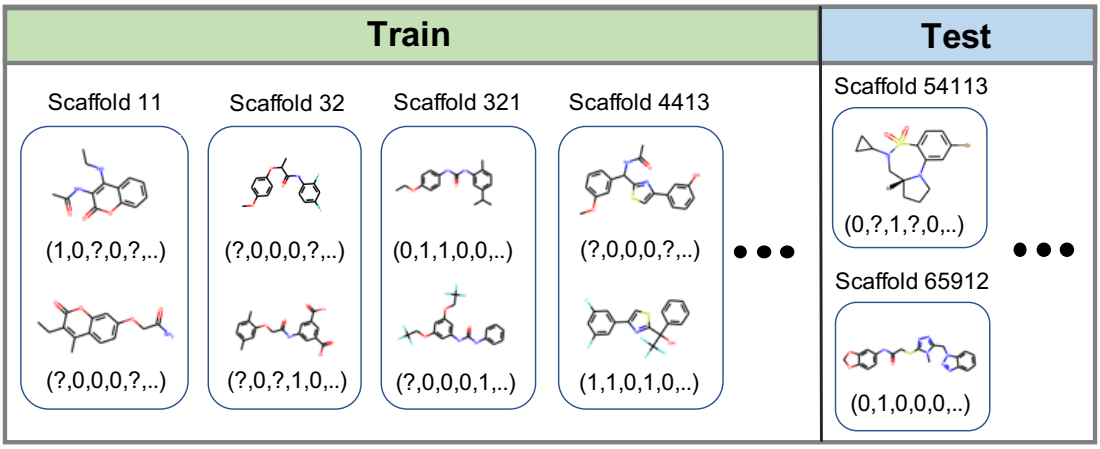

Figure 7: Sample molecules from each scaffold in OGB-MolPCBA, together with target labels: each molecule is associated with 128 binary labels and '?' indicates that the label is not provided for the molecule.

5.5.1 Setup

Problem setting. We consider the domain generalization setting, where the domains are molecular scaffolds, and our goal is to learn models that generalize to structurally-distinct molecules with scaffolds that are not in the training set (Figure 7). This is a multi-task classification problem: for each molecule, we predict the presence or absence of 128 different kinds of biological activity, such as binding to a particular enzyme. In addition, we group the molecules according to their two-dimensional scaffold structure, and annotate each molecule with the scaffold group that it belongs to. Concretely, the input x is a graph representation of a molecule, the target y is a binary vector of length 128 corresponding to various types of biological activity, and the domain d is the scaffold group that the molecule belongs to. Not all biological activities are measured for each molecule, so y can have missing values.

Data. OGB-MolPCBA contains more than 400K small molecules with 128 kinds of prediction labels. Each small molecule is represented as a graph, where the nodes are atoms and the edges are chemical bonds. The molecules are pre-processed using RDKit (Landrum et al., 2006). Input node features are 9-dimensional, including atomic number, chirality, whether the atom is in the ring. Input edge features are 3-dimensional, including bond type and bond stereochemistry.

We split the dataset by scaffold structure. This *scaffold split* (Wu et al., 2018) is also used in the Open Graph Benchmark (Hu et al., 2020b). By attempting to separate structurally different molecules into different subsets, it provides a realistic estimate of model performance in prospective experimental settings. We assign the largest scaffolds to the training set to make it easier for algorithms to leverage scaffold information, and the smallest scaffolds to the test set to ensure that it is maximally diverse in scaffold structure:

1. **Training:** The largest 44,930 scaffolds, with an average of 7.8 molecules per scaffold.
2. **Validation (OOD):** The next largest 31,361 scaffolds, with an average of 1.4 molecules per scaffold.
3. **Test (OOD):** The smallest 43,793 scaffolds, which are all singletons.

Evaluation. We evaluate models by their average Average Precision (AP) across tasks (i.e., we compute the average precision for each task separately, and then average those scores), following Hu

Algorithm	Validation AP (%)	Test AP (%)
ERM	**28.0** (0.2)	**26.8** (0.5)
DeepCORAL	19.3 (0.6)	18.3 (0.4)
IRM	18.6 (0.5)	17.8 (0.6)

Table 10: Baseline results on OGB-MolPCBA. Parentheses show standard deviation across 3 replicates.

	Algorithm	Test AP (%)
Out-of-distribution (standard split)	ERM	26.8 (0.5)
In-distribution (random split)	ERM	**34.4** (0.9)

Table 11: Out-of-distribution vs. in-distribution performance for ERM models on OGB-MolPCBA. In the standard split, we train on molecules from some scaffolds and evaluate on molecules from different scaffolds, whereas in the random split, we randomly divide molecules into training and test sets without using scaffold information.

et al. (2020b). This accounts for the extremely skewed class balance in OGB-MolPCBA (only 1.4% of data is positive). Not all labels are available for each molecule; when calculating the AP for each task, we only consider the labeled molecules for the task.

Potential leverage. We provide the scaffold grouping of molecules for training algorithms to leverage. Finding generalizable representations of molecules across different scaffold groups is useful for models to make accurate extrapolation on unseen scaffold groups. In fact, very recent work (Jin et al., 2020) has leveraged scaffold information of molecules to improve the extrapolation performance of molecular property predictors.

One notable characteristic of the scaffold group is that the size of each group is rather small; on the training split, each scaffold contains only 7.8 molecules on average. This also results in many scaffold groups: 44,930 groups in the training split. In Appendix B.5, we perform some analyses and show that these scaffold groups are well-behaved in the sense that different groups contain a similar ratio of positive labels as well as missing labels.

5.5.2 Baseline results

ERM results and performance drops. For the model, we use the Graph Isomorphism Network (GIN) (Xu et al., 2018) combined with the virtual node (Gilmer et al., 2017)—the best model benchmarked in the Open Graph Benchmark (Hu et al., 2020b). We first compare the generalization performance of ERM on the scaffold split against the conventional random split, in which the entire molecules are randomly split into train/validation/test sets with the same split ratio as the scaffold split (i.e., 80/10/10). The results are provided in Table 10. The test performance of ERM drops by 7.6 points AP when the scaffold split is used, suggesting that the scaffold split is indeed harder than the random split.

Additional baseline methods. We see that ERM performs better than the specialized domain generalization methods, indicating that they are not effective in this particular dataset.

For the hyper-parameters of DeepCORAL and IRM, we find the smaller penalty giving better generalization performance, as the large penalty term makes the training insufficient. We use the 0.1 penalty for DeepCORAL and $\lambda = 1$ for IRM.

The primary issue with the existing methods is that they make the model significantly underfit the training data even when the dropout is turned off. In fact, the training AP of DeepCORAL and IRM is 29.0% and 26.9%, respectively, which are both lower than 36.3%—that of ERM with 0.5 dropout. Also, these methods are primary designed for the case when each group contains decent number of examples, which is not the case for the OGB-MolPCBA dataset. It is fruitful future work to develop effective domain generalization methods that resolve the above issues.

5.5.3 Broader context

Because of the very nature of discovering *new* molecules, out-of-distribution prediction is prevalent in nearly all applications of machine learning to chemistry domains. A variety of tasks and associated datasets have been proposed for molecules of different sizes.

For small organic molecules, the scaffold split has been widely adopted to stress-test models' capability for out-of-distribution generalization. While OGB-MolPCBA primarily focuses on predicting biophysical activity (e.g., protein binding), other datasets in the MoleculeNet include prediction of quantum mechanical properties (e.g., HOMO/LUMO), physical chemistry properties (e.g., water solubility), and physiological properties (e.g., toxicity).

Besides the small molecules, it is also of interest to apply machine learning over larger molecules such as catalysts and proteins. In the domain of catalysis, using machine learning to approximate expensive quantum chemistry simulation has gotten attention. The OC20 dataset has been recently introduced, containing 200+ million samples from quantum chemistry simulations relevant to the discovery of new catalysts for renewable energy storage and other energy applications (Becke, 2014; Chanussot et al., 2020; Zitnick et al., 2020). The OC20 dataset explicitly provides test sets with qualitatively different materials. In the domain of proteins, the recent trend is to use machine learning to predict 3D structure of proteins given their the amino-acid sequence information. This is known as the protein folding problem, and has sometimes been referred to as the Holy Grail of structural biology (Dill and MacCallum, 2012). CASP is a bi-annual competition to benchmark the progress of protein folding (Moult et al., 1995), and it evaluates predictions made on proteins whose 3D structures are identified very recently, presenting a natural temporal distribution shift. Recently, the AlphaFold2 deep learning model obtained breakthrough performance on the CASP challenge (Jumper et al., 2020), demonstrating exciting avenues of machine learning for structural biology.

5.6 Amazon-wilds: Sentiment classification across different users

Models are often trained on data collected on a set of users and then deployed as a general-purpose model across a wide range of users. Yet, they can exhibit large performance disparities across individuals (Li et al., 2019b; Caldas et al., 2018; Geva et al., 2019; Tatman, 2017; Koenecke et al., 2020). These performance gaps are practical limitations in applications that call for good performance across a wide range of users (e.g., user-facing models). In addition, they can be indicative of unfairness of models (Li et al., 2019b; Dwork et al., 2012) as well as their failure to learn the actual task in a generalizable fashion, with models learning the biases specific to individuals instead (Geva et al., 2019).

We study this issue of inter-individual performance disparities in the sentiment classification task on the Amazon-wilds dataset (Ni et al., 2019), in which our goal is to train models with consistently high performance across reviewers. Dataset details and supplemental results are in Appendix B.6.

5.6.1 Setup

Problem setting. We consider the domain generalization setting, where the domains are reviewers, and our goal is to learn models that generalize to new reviewers that are not represented in the training set. We wish to not only perform well on these unseen reviewers on average, but also

	Reviewer ID (d)	Review Text (x)	Stars (y)
Train	Reviewer 1	They are decent shoes. Material quality is good but the color fades very quickly. Not as black in person as shown.	5
		Super easy to put together. Very well built.	5
	Reviewer 2	This works well and was easy to install. The only thing I don't like is that it tilts forward a little bit and I can't figure out how to stop it.	4
		Perfect for the trail camera	5
		...	
	Reviewer 10,000	I am disappointed in the quality of these. They have significantly deteriorated in just a few uses. I am going to stick with using foil.	1
		Very sturdy especially at this price point. I have a memory foam mattress on it with nothing underneath and the slats perform well.	5
Test	Reviewer 10,001	Solidly built plug in. I have had 4 devices plugged in and all charge just fine.	5
		Works perfectly on the wall to hang our wreath without having to do any permanent damage.	5
		...	

Figure 8: Amazon-wilds dataset.

consistently well on a wide range of such reviewers. The task is multi-class sentiment classification. Concretely, the input x is the text of a review, the label y is a corresponding star rating from 1 to 5, and the domain d is the identifier of the user that wrote the review.

Data. The dataset is a modified version of the Amazon Reviews dataset (Ni et al., 2019) and comprises 1.4 million customer reviews on Amazon. To measure generalization to unseen reviewers, we train on reviews written by a set of reviewers and consider reviews written by *unseen* reviewers at test time. Specifically, we consider the following random split across reviewers:

1. **Training:** 1,000,124 reviews from 5,008 reviewers.
2. **Validation (OOD):** 100,050 reviews from another set of 1,334 reviewers, distinct from training and test (OOD).
3. **Test (OOD):** 100,050 reviews from another set 1,334 reviewers, distinct from training and validation (OOD).
4. **Validation (ID):** 100,050 reviews from 1,334 of the 4,974 reviewers in the training set.
5. **Test (ID):** 100,050 reviews from 1,334 of the 4,974 reviewers in the training set.

While we primarily evaluate model performance on the above OOD test set, we also provide in-distribution validation and test sets for potential use in hyperparameter tuning and additional reporting. These in-distribution splits comprise reviews written by reviewers in the training set.

Evaluation. To assess whether models perform consistently well across reviewers, we evaluate models by their accuracy on the reviewer at the 10th percentile; this follows the federated learning literature (Li et al., 2019b).

Potential leverage. We include thousands of reviewers in the training set, capturing variations across a wide range of reviewers. In addition, we provide reviewer ID annotations for all reviews in the dataset. These annotations could be used to directly mitigate performance disparities across reviewers seen during training time, which is a first step toward mitigating the performance drop.

Algorithm	Validation (OOD) 10th percentile	Validation (OOD) Average	Test (OOD) 10th percentile	Test (OOD) Average	Validation (ID) 10th percentile	Validation (ID) Average
ERM	57.3 (0.0)	74.3 (0.0)	56.0 (0.0)	73.5 (0.1)	58.7 (0.0)	75.5 (0.0)
DeepCORAL	56.4 (0.8)	74.2 (0.2)	55.1 (0.8)	73.3 (0.2)	57.3 (0.0)	75.0 (0.2)
IRM	56.9 (0.8)	73.6 (0.2)	56.0 (0.0)	72.8 (0.2)	58.2 (0.8)	74.9 (0.1)

Table 12: Baseline results on Amazon-wilds. We report the accuracy of models trained using three baseline algorithms: ERM, DeepCORAL, and IRM. In addition to the average accuracy across all reviews, we compute the accuracy for each reviewer and report the performance for the reviewer in the 10th percentile.

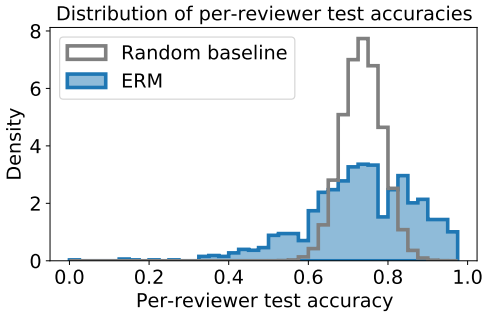

Figure 9: Distribution of per-reviewer accuracy on the test set for the ERM model (blue). The corresponding random baseline would have per-reviewer accuracy distribution in grey.

5.6.2 Baseline results

ERM results and performance drops. First, we present results of a standard model on the out-of-distribution test set, showing that there are significant performance disparities across unseen reviewers. A BERT-base-uncased model trained with the standard ERM objective performs well on average, but their performance vary widely across reviewers (Figure 9, Table 12). Despite the high average accuracy of 73.5%, per-reviewer accuracies vary widely between 100.0% and 0.0%, with accuracy at the tenth percentile of 56.0%. The above variation is larger than expected from randomness; a random binomial baseline with equal average accuracy would have a tenth percentile accuracy of 67.0%.

To demonstrate that there is a significant performance drop due to the distribution shift, we now compare the above results with the performance of in-distribution baseline models; these models are evaluated on the same test set as the above, but are oracle models trained instead on reviews from the target distribution, thus experiencing no distribution shift upon evaluation. Concretely, we train the oracle baseline models specific to each of those reviewers by fine-tuning on reviews written by each reviewer,[9] and show that these models achieve high accuracy. We consider reviewers at the tenth percentile or below in terms of accuracy of the standard ERM model, and in particular those with the highest number of reviews. Despite being trained on data that are orders of magnitude smaller (less than a thousand reviews per user, compared to the full training set of 1 million reviews), the oracle baseline models achieve 61.4% accuracy on average, outperforming the standard models by 11.8% on those reviewers.

9. The training reviews are disjoint from the test reviews.

	Accuracy on reviewers in the bottom 10%
OOD baseline (ERM)	49.6 (0.8)
ID baseline (oracle)	61.4 (1.3)

Table 13: Performance drops due to distribution shift on Amazon-wilds. To demonstrate that the poor out-of-distribution performance of the ERM model (Table 12) stems from the distribution shift, we compare with in-distribution (ID) baseline models, which are oracle models finetuned on each reviewer. We report the average accuracy on a fixed set of 10 reviewers, which are in the 10th percentile or below for the ERM model. Despite being trained on data that are orders of magnitude smaller (less than a thousand reviews per user, compared to the full training set of 1 million reviews), the oracle baseline models achieve outperform the ERM models, demonstrating that the low 10th percentile of the ERM model stems from the distribution shift.

Additional baseline methods. Above, we observed that the model trained via ERM performs poorly on the out-of-distribution test set. We now consider models trained by existing robust training algorithms, and show that they also perform poorly in the out-of-distribution test set, failing to mitigate the performance drop (Table 12). We observe that resampling to achieve uniform class balance fails to improve the 10th percentile accuracy, showing that variation across users cannot be solved simply by accounting for label imbalance. In addition, we show that deep CORAL and IRM fails to improve performance on unseen users.

Discussion. While we focus on unseen reviewers in Amazon-wilds, we observe that performance disparity across users is similarly a problem in our in-distribution evaluation set (Table 12). This suggests that mitigating performance disparities across in-distribution (i.e., seen at training time) reviewers is a promising first step for improving robustness.

5.6.3 Broader context

Performance disparities across individuals have been observed in a wide range of tasks and applications, including in natural language processing (Geva et al., 2019), automatic speech recognition (Koenecke et al., 2020; Tatman, 2017), federated learning (Li et al., 2019b; Caldas et al., 2018), and medical imaging (Badgeley et al., 2019). These performance gaps are practical limitations in applications that call for good performance across a wide range of users, including many user-facing applications such as speech recognition (Koenecke et al., 2020; Tatman, 2017) and personalized recommender systems (Patro et al., 2020), tools used for analysis of individuals such as sentiment classification in computational social science (West et al., 2014) and user analytics (Lau et al., 2014), and applications in federated learning. These performance disparities have also been studied in the context of algorithmic fairness, including in the federated learning literature, in which uniform performance across individuals is cast as a goal toward fairness (Li et al., 2019b; Dwork et al., 2012). Lastly, these performance disparities can also highlight models' failures to learn the actual task in a generalizable manner; instead, some models have been shown learn the biases specific to individuals. Prior work has shown that individuals—technicians for medical imaging in this case—can not only be identified from data, but also are predictive of the diagnosis, highlighting the risk of learning to classify technicians rather than the medical condition (Badgeley et al., 2019). More directly, across a few natural language processing tasks where examples are annotated by crowdworkers, models have been observed to perform well on annotators that are commonly seen at training time, but fail to generalize to unseen annotators, suggesting that models are merely learning annotator-specific patterns and not the task (Geva et al., 2019).

Toxic	Comment Text	Male	Female	LGBTQ	White	Black	...	Christian
0	I applaud your father. He was a good man! We need more like him.	1	0	0	0	0	...	0
0	As a Christian, I will not be patronizing any of those businesses.	0	0	0	0	0	...	1
0	What do Black and LGBT people have to do with bicycle licensing?	0	0	1	0	1	...	0
0	Government agencies track down foreign baddies and protect law-abiding white citizens. How many shows does that describe?	0	0	0	1	0	...	0
1	Maybe you should learn to write a coherent sentence so we can understand WTF your point is.	0	0	0	0	0	...	0

Figure 10: Example comments from CivilComments-wilds.

5.7 CivilComments-wilds: Toxicity classification across demographic identities

Automatic review of user-generated text—e.g., detecting if an online comment is toxic—is an important tool for moderating the sheer volume of text being written every day on the Internet. Unfortunately, prior work has documented biases in automatic moderation tools; for example, toxicity classifiers have been shown to spuriously associate the mention of certain demographic groups with toxicity (Park et al., 2018; Dixon et al., 2018).

We study this issue through a modified variant of the CivilComments dataset (Borkan et al., 2019b), a large collection of comments on online articles taken from the Civil Comments platform and annotated for toxicity and demographic mentions by multiple crowdworkers. Additional dataset and model details are in Appendix B.7.

5.7.1 Setup

Problem setting. We cast CivilComments-wilds as a subpopulation shift problem, where the subpopulations correspond to different demographic identities, and our goal is to do well on all subpopulations (and not just on average across these subpopulations). Specifically, we focus on mitigating biases with respect to comments that mention particular demographic identities, and not comments written by members of those demographic identities; we discuss this distinction in the broader context section below.

The task is a binary classification task of determining if a comment is toxic. Concretely, the input x is a comment on an online article (comprising one or more sentences of text) and the label y is whether it is rated toxic or not. In CivilComments-wilds, unlike in most of the other datasets we consider, the domain annotation d is a multi-dimensional binary vector, with the 8 dimensions corresponding to whether the comment mentions each of the 8 demographic identities *male*, *female*, *LGBTQ*, *Christian*, *Muslim*, *other religions*, *Black*, and *White*.

Data. CivilComments-wilds comprises 450,000 comments, each annotated for toxicity and demographic mentions by multiple crowdworkers. We model toxicity classification as a binary task. Toxicity labels were obtained in the original dataset via crowdsourcing and majority vote, with each comment being reviewed by at least 10 crowdworkers. Annotations of demographic mentions were similarly obtained through crowdsourcing and majority vote.

Each comment was originally made on some online article. We randomly partitioned these articles into disjoint training, validation, and test splits, and then formed the corresponding datasets by taking all comments on the articles in those splits. This gives the following splits:

1. **Training:** 269,038 comments.
2. **Validation:** 45,180 comments.
3. **Test:** 133,782 comments.

Evaluation. We evaluate a model by its worst-group accuracy, i.e., its lowest accuracy over groups of the test data that we define below.

As mentioned at the start of this section, toxicity classifiers can spuriously latch onto mentions of particular demographic identities, resulting in a biased tendency to flag comments that innocuously mention certain demographic groups as toxic (Park et al., 2018; Dixon et al., 2018). To measure the extent of this bias, we define subpopulations based on whether they mention a particular demographic identity, compute the sensitivity (a.k.a. recall, or true positive rate) and specificity (a.k.a. true negative rate) of the classifier on each subpopulation, and then report the worst of these two metrics over all subpopulations of interest. This is equivalent to further dividing each subpopulation into two groups according to the label, and then computing the accuracy on each of these two groups; see Appendix B.7 for more discussion.

Specifically, for each of the 8 identities we study (e.g., "male"), we form 2 groups based on the toxicity label (e.g., one group of comments that mention the male gender and are toxic, and another group that mentions the male gender and are not toxic), for a total of 16 groups. These groups overlap (a comment might mention multiple identities) and are not a complete partition (a comment might not mention any identity).

We then measure a model's performance by its worst-group accuracy, i.e., its lowest accuracy over these 16 groups. A high worst-group accuracy (relative to average accuracy) implies that the model is not spuriously associating a demographic identity with toxicity. We can view this subpopulation shift problem as testing on multiple test distributions (corresponding to different subsets of the test set, based on demographic identities and the label) and reporting the worst performance over these different test distributions.

Potential leverage. Since demographic identity annotations are provided at training time, we have an i.i.d. dataset available at training time for each of the test distributions of interest (corresponding to each group). Moreover, even though demographic identity annotations are unavailable at test time, they are relatively straightforward to predict.

5.7.2 Baseline results

ERM results and performance drops. We fine-tuned a standard BERT-base-uncased model (Devlin et al., 2019) via ERM and found that it does well on average, with 92.2% average accuracy, but poorly on the worst group, with 58.0% accuracy on toxic comments that mention Christianity (Table 14). Overall, accuracy on toxic comments (which are a minority of the dataset) was lower than accuracy on non-toxic comments, so we also trained a reweighted model (Section 4) that balanced toxic and non-toxic comments. This reweighted model had a slightly worse average accuracy of 89.9% and a better worst-group accuracy of 68.2%, on non-toxic comments that mention Black people (Table 14, Reweighted (label)). These results match the spurious correlations reported in the literature, and the large disparity between average and worst-group accuracy on these models show that subpopulation shift can substantially degrade performance.

Additional baseline methods. Given a specified set of groups, the group DRO algorithm (Hu et al., 2018; Sagawa et al., 2020) adaptively finds a reweighting of those groups that minimizes worst-group training loss. We trained group DRO models by forming two groups based on the labels; this corresponds to reweighting positive and negative examples to balance their training losses. These models seem to slightly outperform standard reweighted models, though the difference is not statistically significant given three replicates (Table 14, Group DRO (label)). There remains a large gap between average and worst-group accuracies.

In Table 15, we show the results of this group DRO model on each demographic subpopulation. As with the standard reweighted model, the worst-performing group comprises non-toxic comments that mention Black people. We attempted to address this by training a group DRO model as well as a standard reweighted model using four groups, corresponding to all combinations of label and

Algorithm	Avg val acc	Worst-group val acc	Avg test acc	Worst-group test acc
ERM	92.3 (0.6)	53.6 (0.7)	**92.2** (0.6)	58.0 (1.2)
Reweighted (label)	90.1 (0.1)	67.7 (0.9)	89.9 (0.0)	68.2 (0.5)
Reweighted (label × Black)	89.1 (0.3)	67.9 (1.2)	88.9 (0.3)	67.3 (0.1)
Group DRO (label)	90.3 (0.1)	66.6 (1.2)	90.1 (0.2)	69.2 (1.3)
Group DRO (label × Black)	89.6 (0.3)	68.7 (1.0)	89.4 (0.3)	**70.4** (2.1)

Table 14: Baseline results on CivilComments-wilds. The reweighted (label) algorithm samples equally from the positive and negative class; the group DRO (label) algorithm additionally weights these classes so as to minimize the maximum of the average positive training loss and average negative training loss. Similarly, the reweighted (label × Black) and group DRO (label × Black) algorithms sample equally from the four groups corresponding to all combinations of class and whether there is a mention of Black identity. We show standard deviation across random seeds in parentheses.

Demographic	Test accuracy on non-toxic comments	Test accuracy on toxic comments
Male	87.3 (0.7)	76.6 (1.3)
Female	89.0 (0.6)	75.2 (1.6)
LGBTQ	74.6 (0.5)	76.5 (1.0)
Christian	92.1 (0.2)	72.6 (0.8)
Muslim	80.9 (1.0)	73.4 (1.4)
Other religions	86.1 (0.1)	74.6 (1.9)
Black	**69.2** (1.3)	82.2 (1.6)
White	71.2 (1.4)	80.5 (1.9)

Table 15: Accuracies on each subpopulation in CivilComments-wilds, averaged over models trained by group DRO (label).

mention of Black identity. These models performed marginally but not statistically significantly better (Table 14, label × Black), which slight improvements on the Black groups but poorer accuracy on some other groups, specifically, non-toxic LBGTQ comments; see Appendix B.7 for more discussion.

Adapting the reweighting and group DRO methods to handle multiple overlapping groups, which were not studied in their original settings, could be a potential approach to improving accuracy on this task. Another potential approach is using baselining to account for different groups having different intrinsic levels of difficulty (Oren et al., 2019).

5.7.3 Broader context

The CivilComments-wilds dataset does not assume that user demographics are available; instead, it uses mentions of different demographic identities in the actual comment text. For example, we want models that do not associate comments that mention being Black with being toxic, regardless of whether a Black or non-Black person wrote the comment. This setting is particularly relevant when user demographics are unavailable, e.g., when considering anonymous online comments.

A related and important setting is subpopulation shifts with respect to user demographics (e.g., the demographics of the author of the comment, regardless of the content of the comment). Such demographic disparities have been widely documented in natural language and speech processing tasks (Hovy and Spruit, 2016), among other areas. For example, NLP models have been shown to obtain worse performance on African-American Vernacular English compared to Standard American English on part-of-speech tagging (Jørgensen et al., 2015), dependency parsing (Blodgett et al., 2016), language identification (Blodgett and O'Connor, 2017), and auto-correct systems (Hashimoto et al.,

2018). Similar disparities exist in speech, with state-of-the-art commerical systems obtaining higher word error rates on particular races (Koenecke et al., 2020) and genders and dialects (Tatman, 2017).

These disparities are present not just in academic models, but in large-scale commercial systems that are already widely deployed, e.g., in speech-to-text systems from Amazon, Apple, Google, IBM, and Microsoft (Tatman, 2017; Koenecke et al., 2020) or language identification systems from IBM, Microsoft, and Twitter (Blodgett and O'Connor, 2017). Indeed, the original CivilComments dataset was developed by Google's Conversation AI team, which is also behind a public toxicity classifier (Perspective API) that was developed in partnership with The New York Times (NYTimes, 2016).

6. Potential extensions to other application areas

Beyond the datasets currently included in WILDS, there are many other applications where it is critical for models to be robust to distribution shifts. Here, we discuss some of these applications and the challenges of finding appropriate benchmark datasets in those areas. These all represent important avenues of future work, and we highly welcome community contributions of benchmark datasets in these areas.

6.1 Algorithmic fairness

Distribution shifts which worsen algorithmic performance in historically disadvantaged or minority populations have been frequently discussed in the algorithmic fairness literature and represent an important area for research. Geographic inequities are one concern (Shankar et al., 2017; Atwood et al., 2020): for example, publicly available image datasets overrepresent images from the United States and Europe, producing drops in performance on images from the developing world (Shankar et al., 2017) and prompting creation of more geographically diverse image datasets (Atwood et al., 2020). Racial disparities are a second concern. Commercial gender classifiers are more likely to misclassify the gender of darker-skinned women, likely in part because training datasets overrepresent lighter-skinned subjects (Buolamwini and Gebru, 2018). Algorithmic pedestrian detection systems achieve poorer performance when recognizing darker-skinned pedestrians (Wilson et al., 2019). As discussed in Section 5.7, NLP models also show racial bias.

Publicly available algorithmic fairness benchmarks (Mehrabi et al., 2019)—e.g., the COMPAS recidivism dataset (Larson et al., 2016)—often suffer from several limitations; ameliorating these represents a promising direction for future work. First, the datasets are often quite small by the standards of modern machine learning: the COMPAS dataset has only a few thousand rows (Larson et al., 2016). Second, the datasets sometimes have relatively few features, allowing even simple algorithms to achieve state-of-the-art performance and limiting the benefit of more sophisticated approaches. On the COMPAS dataset, logistic regression on a small number of features performs comparably to a black-box commercial algorithm (Jung et al., 2020; Dressel and Farid, 2018). Relatedly, disparities in performance across subgroups are not always large, again limiting opportunity for improvement from more sophisticated algorithms (Larrazabal et al., 2020). Third, the datasets sometimes represent "toy" problems which, while useful for illustrating the theoretical properties of an approach, do not represent real-world problems of interest. For example, the UCI Adult Income dataset (Asuncion and Newman, 2007) is widely used as a fairness benchmark, but the classification task—predicting whether a person will have an income above $50,000—does not represent a real-world application. Finally, because many of the domains in which algorithmic fairness is of most concern—for example, criminal justice and healthcare—are high-stakes and disparities are politically sensitive, it can be difficult to make datasets publicly available.

Creating algorithmic fairness benchmarks which do not suffer from these limitations represents a promising direction for future work. In particular, such datasets would ideally have: 1) information about a sensitive attribute like race or gender; 2) a prediction task which is of immediate real-world interest; 3) enough samples, a rich enough feature set, and large enough disparities in

group performance that more sophisticated machine learning approaches would plausibly produce improvement over naive approaches.

6.2 Medicine and healthcare

Substantial evidence indicates the potential for distribution shifts in medical settings. One concern is *demographic* distribution shifts (e.g., across race, gender, or socioeconomic status) (Chen et al., 2020), similar to the algorithmic fairness concerns discussed above. Historically disadvantaged populations are underrepresented in many medical datasets, potentially producing inferior algorithmic performance on these groups. A second source of distribution shifts is heterogeneity *across hospitals*; this might include differences in imaging equipment and protocol, as discussed in Section 5.4, and also differences in other operational protocols such as lab tests (D'Amour et al., 2020; Subbaswamy et al., 2020). Finally, changes *over time* in care settings can also produce distribution shifts and drops in algorithmic performance: Nestor et al. (2019) shows that switching between two electronic health record (EHR) systems produced a drop in algorithmic performance. Similarly, temporal shifts in conditions affecting patient populations could cause distribution shifts: for example, the COVID-19 epidemic has affected the distribution of chest radiographs (Wong et al., 2020).

Creating medical distribution shift benchmarks thus represents a promising direction for future work, if several challenges can be overcome. First, while there are large demographic disparities in healthcare outcomes (e.g., by race, or socioeconomic status), many of them are not due to distribution shifts, but to disparities in non-algorithmic factors (e.g., access to care or prevalence of comorbidities (Chen et al., 2020)) or to algorithmic problems unrelated to distribution shift (e.g., choice of a biased outcome variable (Obermeyer et al., 2019)). Several previous investigations have found relatively small disparities in algorithmic performance across demographic groups (Chen et al., 2019a; Larrazabal et al., 2020); Seyyed-Kalantari et al. (2020) finds larger disparities in TPR across demographic groups, and future work should investigate whether there are disparities in other performance metrics.

A second challenge to overcome is data availability, in part because stringent medical privacy laws often preclude data sharing (Price and Cohen, 2019). For example, EHR datasets are fundamental to medical decision-making, but there are few widely adopted EHR benchmarks (the MIMIC database representing one example (Johnson et al., 2016)) and relatively little progress in predictive performance has been made on them (Bellamy et al., 2020).

6.3 Natural language and speech processing

As mentioned in Section 5.7, recent work found that automated speech recognition (ASR) systems have higher error rates for black speakers than for white speakers (Koenecke et al., 2020) and for women and speakers of some dialects (Tatman, 2017). This is a natural setting for developing methods that are robust to subpopulation shifts, like in CivilComments-wilds. The aforementioned papers use commercial ASR systems to demonstrate these disparities. However, there are many public speech datasets with speaker metadata that could potentially be used to construct a benchmark, e.g., LibriSpeech (Panayotov et al., 2015), the Speech Accent Archive (Weinberger, 2015), VoxCeleb2 (Chung et al., 2018), the Spoken Wikipedia Corpus (Baumann et al., 2019), and Common Voice (Ardila et al., 2020).

In natural language processing (NLP), a current focus is on challenge datasets that are carefully crafted to test particular aspects of models, e.g., HANS (McCoy et al., 2019b), PAWS (Zhang et al., 2019), counterfactually-augmented datasets (Kaushik et al., 2019), or CheckList (Ribeiro et al., 2020). These challenge datasets represent a form of distribution shift, as their test distributions are often (deliberately) quite different from the data distributions that the models were originally trained on. However, one issue with using these datasets as benchmark datasets is that they represent just a few of the many potential shifts in NLP applications, so hill-climbing on the specific types of shifts that

these datasets represent might not translate to progress on general robust NLP models. Similarly, there are several synthetic datasets designed to test compositional generalization, such as CLEVR (Johnson et al., 2017), SCAN (Lake and Baroni, 2018), and COGS (Kim and Linzen, 2020). The test sets in these datasets are chosen such that models need to generalize to novel combinations of, e.g., familiar primitives and grammatical roles (Kim and Linzen, 2020).

All of the NLP examples we have discussed so far deal with English-language models; other languages typically have fewer and smaller datasets available for training and benchmarking models. Multi-lingual models and benchmarks (e.g., Conneau et al. (2018); Conneau and Lample (2019); Hu et al. (2020a); Clark et al. (2020)) represent another source of subpopulation shifts with corresponding disparities in model performance: training sets might contain fewer examples in low-resource languages (Nekoto et al., 2020), but we would still hope for high model performance on these minority groups. A challenge here is that multi-lingual models are often more complex, requiring larger datasets, than their mono-lingual counterparts.

6.4 Code

Machine learning can aid programming and software engineering in various ways: automatic code completion (Raychev et al., 2014; Svyatkovskiy et al., 2019), program synthesis (Bunel et al., 2018; Kulal et al., 2019), program repair (Vasic et al., 2019; Yasunaga and Liang, 2020), code search (Husain et al., 2019), and code summarization (Allamanis et al., 2015). However, deploying these systems in practice faces several forms of distribution shifts. One major challenge is the shifts across code bases, where systems need to adapt to the project content, coding convention, and library or API usage, etc. in each code base (Nita and Notkin, 2010; Allamanis and Brockschmidt, 2017). A second source of shifts is programming languages, which includes adaption across different domain-specific languages (DSLs) (e.g., for robotic environments) (Shin et al., 2019) and different versions of languages (e.g., Python 2 and 3) (Malloy and Power, 2017). Another critical challenge is the subpopulation shift from training time to real usage: for instance, Hellendoorn et al. (2019) show that existing code completion systems, typically trained as language models on source code, perform poorly on real completion instances that developers use the most in IDEs, such as intra-project API calls. We hope to include a dataset about code in the future.

6.5 Education

ML models can help in educational settings in a variety of ways: e.g., assisting in grading (Piech et al., 2013; Shermis, 2014; Kulkarni et al., 2014; Taghipour and Ng, 2016), estimating student knowledge and ability (Desmarais and Baker, 2012; Wu et al., 2020), identifying students who need assistance (Ahadi et al., 2015), or automatically generating explanations for student submissions (Williams et al., 2016; Wu et al., 2019a). However, there are substantial distribution shift issues to deal with in these settings as well. For example, automatic essay scoring has been found to be affected by rater bias (Amorim et al., 2018) and spurious correlations like essay length (Perelman, 2014), leading to problems with subpopulation shift; and these systems would also ideally generalize across different educational contexts, e.g., a model for scoring grammar should work well across multiple different essay prompts. Recent attempts at predicting grades algorithmically (BBC, 2020; Broussard, 2020) have also been found to be biased against certain subpopulations.

Unfortunately, finding a suitable education benchmark is difficult due to a general lack of standardized datasets, in part due to student privacy concerns and the proprietary nature of large-scale standardized tests. Datasets from massive open online courses are a potential source of large-scale data (Kulkarni et al., 2015). In general, dataset construction for ML in education is an active area—e.g., the NeurIPS 2020 workshop on Machine Learning for Education[10] has a segment devoted to finding "ImageNets for education"—and we hope to be able to include one in the future.

10. https://www.ml4ed.org/

6.6 Robotics

Robot learning has emerged as a strong paradigm for automatically acquiring complex and skilled behaviors such as locomotion (Yang et al., 2019; Peng et al., 2020), navigation (Mirowski et al., 2017; Kahn et al., 2020), and manipulation (Gu et al., 2017; et al, 2019). However, the advent of learning-based techniques for robotics has not convincingly addressed, and has perhaps even exasperated, problems stemming from distribution shift. These problems have manifested in many ways, including shifts induced by weather and lighting changes (Wulfmeier et al., 2018), location changes (Gupta et al., 2018), and the simulation-to-real-world gap (Sadeghi and Levine, 2017; Tobin et al., 2017). Dealing with these challenging scenarios is critical to deploying robots in the real world, especially in high-stakes decision-making scenarios.

For example, to safely deploy autonomous driving vehicles, it is critical that these systems work reliably and robustly across the huge variety of conditions that exist in the real world, such as locations, lighting and weather conditions, and sensor intrinsics. This is a challenging requirement, as many of these conditions may be underrepresented, or not represented at all, by the available training data. Indeed, prior work has shown that naively trained models can suffer at segmenting nighttime driving scenes (Dai and Van Gool, 2018), detecting relevant objects in new or challenging locations and settings (Yu et al., 2020; Sun et al., 2020a), and, as discussed earlier, detecting pedestrians with darker skin tones (Wilson et al., 2019).

Creating a benchmark for distribution shifts in robotics applications, such as autonomous driving, represents a promising direction for future work. Here, we briefly summarize our initial findings on distribution shifts in the BDD100K driving dataset (Yu et al., 2020), which is publicly available and widely used, including in some of the works listed above.

BDD100K. We investigated the task of multi-label binary classification of the presence of each object category in each image. In general, we found no substantial performance drops across a wide range of different test scenarios, including user shifts, weather and time shifts, and location shifts. We provide additional details in Section C.1.

Our findings contrast with previous findings that other tasks, such as object detection and segmentation, can suffer under the same types of shifts on the same dataset (Yu et al., 2020; Dai and Van Gool, 2018). Currently, WILDS consists of datasets involving classification and regression tasks. However, most tasks of interest in autonomous driving, and robotics in general, are difficult to formulate as classification or regression. For example, autonomous driving applications may require models for object detection or lane and scene segmentation. These tasks are often more challenging than classification tasks, and we speculate that they may suffer more severely from distribution shift.

7. Potential extensions to other problem settings

Thus far, we have focused on two problem settings involving domain shifts: domain generalization and subpopulation shifts. In this section, we discuss additional problem settings within the framework of domain shifts that can also apply to WILDS datasets. Extending WILDS to benchmark existing algorithms for these settings represents an important avenue for future work, and we welcome community contributions towards this effort.

7.1 Problem settings in domain shifts

Within the general framework of domain shifts, specific problem settings can differ along the following axes of variation:

1. **Seen versus unseen test domains.** Test domains may be seen during training time ($\mathcal{D}^{\mathsf{test}} \subseteq \mathcal{D}^{\mathsf{train}}$), as in subpopulation shift, or unseen ($\mathcal{D}^{\mathsf{train}} \cap \mathcal{D}^{\mathsf{test}} = \emptyset$), as in domain generalization. Thus far, we have focused on this axis of variation.

2. **Train-time domain annotations.** The domain identity d may be observed for none, some, or all of the training examples. Train-time domain annotations are straightforward to obtain in some settings, e.g., we should know which patients in the training sets came from which hospitals, but can be harder to obtain in some settings, e.g., we might only have demographic information on a subset of training users. In our domain generalization and subpopulation shift settings, d is always observed at training time.

3. **Test-time domain annotations.** The domain identity d may be observed for none, some, or all of the test examples. Test-time domain annotations allow models to be domain-specific, e.g., by treating domain identity as a feature if the train and test domains overlap. For example, if the domains correspond to continents and the data to satellite images from a continent, we would presumably know what continent each image was taken from. On the other hand, if the domains correspond to demographic information, this might be hard to obtain at test time (as well as training time, as mentioned above). In domain generalization, d is observed at test time, but it is not particularly helpful by itself as all of the test domains are unseen at training time. In subpopulation shift, d is unobserved at test time.

4. **Test-time unlabeled data**. Varying amounts of unlabeled test data—samples of x drawn from the test distribution P^{test}—may be available, from none to a small batch to a large pool. This affects the degree to which models can adapt to test distributions. For example, if the domains correspond to locations and the data points to photos taken at those locations, we might assume access to some unlabeled photos taken at the test locations.

Each combination of the above four factors corresponds to a problem setting, each with a specific set of applicable methods. In the current version of the WILDS benchmark, we focus on domain generalization and subpopulation shifts, which represent specific configurations of these factors. We briefly discuss a few other problem settings, as summarized in Table 16.

Problem setting	Unseen test domains	Train domain annotations	Test domain annotations	Test unlabeled data
Domain generalization	✓	✓	✗	✗
Subpopulation shift	✗	✓	✓	✗
Test-time adaptation	✓	✓	✓	limited
Domain adaptation	✓	✓	✓	full

Table 16: WILDS focuses on two specific settings of domain shift: domain generalization and subpopulation shift. These two settings vary only in whether the test domains are seen or unseen. Other problem settings that can apply to WILDS datasets include test-time adaptation and unsupervised domain adaptation.

7.2 Unsupervised domain adaptation

In the presence of distribution shift, a potential source of leverage is observing unlabeled test points from the test distribution. In the unsupervised domain adaptation setting, we assume that at training time, we have access to a large amount of unlabeled data from each test distribution of interest, as well as the resources to train a separate model for each test distribution. For example, in a satellite imagery setting like FMOW, it might be appropriate to assume that we have access to a large set of unlabeled recent satellite images from each continent and the wherewithal to train a separate model for each continent.

Many of the methods for domain generalization discussed in Section 4 were originally methods for domain adaptation, since methods for both settings share the common goal of learning models that can transfer between domains; for example, methods that learn features that have similar distributions across domains are equally applicable to both settings (e.g., Ben-David et al. (2006); Long et al. (2015); Sun et al. (2016); Ganin et al. (2016); Tzeng et al. (2017); Shen et al. (2018); Wu et al. (2019b)). Other methods rely on knowing the test distribution and are thus specific to domain adaptation, e.g., learning to map data points from source to target domains (Hoffman et al., 2018), or estimating the test label distribution from unlabeled test data (Saerens et al., 2002; Zhang et al., 2013; Lipton et al., 2018; Azizzadenesheli et al., 2019; Alexandari et al., 2020; Garg et al., 2020).

7.3 Test-time adaptation

A closely-related setting to unsupervised domain adaptation is test-time adaptation, which also assumes the availability of unlabeled test data. For datasets where there are many potential test domains (e.g., in iWildCam2020-wilds, we want a model that can ideally generalize to any camera trap), it might be infeasible to train a separate model for each test domain, as unsupervised domain adaptation would require. In the test-time adaptation setting, we assume that a model is allowed to adapt to a small amount of unlabeled test data in a way that is computationally much less intensive than typical domain adaptation methods. This is a difference of degree and not of kind, but it can have significant practical implications. For example, domain adaptation approaches typically require access to the training set and a large unlabeled test set at the same time, whereas test-time adaptation methods typically only require the learned model (which can be much smaller than the original training set) as well as a smaller amount of unlabeled test data.

A number of test-time adaptation methods have been recently proposed (Li et al., 2017c; Sun et al., 2020b; Wang et al., 2020a). For example, adaptive risk minimization (ARM) is a meta-learning approach that adapts models to each batch of test examples under the assumption that all data points in a batch come from the same domain (Zhang et al., 2020). Many datasets in Wilds are suitable for the test-time adaptation setting. For example, in iWildCam2020-wilds, images from the same domain are highly similar, sharing the same location, background, and camera angle, and prior work has shown inferring these shared features can improve performance considerably (Beery et al., 2020b).

7.4 Selective prediction

A different problem setting that is orthogonal to the settings described above is selective prediction. In the selective prediction setting, models are allowed to abstain on points where their confidence is below a certain threshold. This is appropriate when, for example, abstentions can be handled by backing off to human experts, such as pathologists for Camelyon17, content moderators for CivilComments, wildlife experts for iWildCam2020, etc. Many methods for selective prediction have been developed, from simply using softmax probabilities as a proxy for confidence (Cordella et al., 1995; Geifman and El-Yaniv, 2017), to methods involving ensembles of models (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Geifman et al., 2018) or jointly learning to abstain and classify (Bartlett and Wegkamp, 2008; Geifman and El-Yaniv, 2019; Feng et al., 2019).

Intuitively, even if a model is not robust to a distribution shift, it might at least be able to maintain high accuracies on some subset of points that are close to the training distribution, while abstaining on the other points. Indeed, prior work has shown that selective prediction can improve model accuracy under distribution shifts (Pimentel et al., 2014; Hendrycks and Gimpel, 2017; Liang et al., 2018; Ovadia et al., 2019; Feng et al., 2019; Kamath et al., 2020). However, distribution shifts still pose a problem to selective prediction methods; for instance, it is difficult to maintain desired abstention rates under distribution shifts (Kompa et al., 2020), and confidence estimates have been found to drift over time (e.g., Davis et al. (2017)).

8. Discussion

8.1 Underspecification

Prior work has shown that there is often insufficient information at training time to distinguish models that would generalize well under a particular distribution shift; many models that perform similarly in-distribution (ID) can vary substantially out-of-distribution (OOD). This instability in OOD performance has been reported in natural language processing settings (McCoy et al., 2019a; Kim and Linzen, 2020) and also in vision and healthcare applications (D'Amour et al., 2020). In Wilds, we see this effect most dramatically in the Camelyon17-wilds dataset, where the variability in OOD accuracy is on the order of ±9.9% while the variability in ID accuracy is on the order of ±0.4%. Pathology datasets like Camelyon17-wilds tend to comprise a small number of slides (though a large number of patches extracted from these slides), and there is a lot of correlation across patches from the same slides and hospitals, which can exacerbate this instability (Zhou et al., 2020). However, the other Wilds datasets do not show significantly higher OOD performance variance compared to ID performance variance. In Wilds, we attempt to mitigate this problem of underspecification (D'Amour et al., 2020) by ensuring that all of the datasets comprise multiple training domains. We speculate that the relatively similar ID and OOD variances in our other datasets could be in part because of this, and in part also because we select models based on their OOD validation performance (as opposed to the standard practice of ID validation), but further investigation is required.

On a related note, Gulrajani and Lopez-Paz (2020) showed that model selection can have a large effect on OOD performance. In particularly, they show that in their experiments on the DomainBed dataset, using ID validation accuracy instead of OOD validation accuracy leads to higher OOD accuracy. This contrasts with our approach of using OOD validation sets, which we find to generally provide a good estimate of OOD test performance. One reason could be that the datasets in DomainBed are quite different from the Wilds datasets (e.g., the former have a small number of domains); when to use different model selection procedures and validation sets is an important question for future work to resolve.

8.2 Shifts across multiple axes

In the PovertyMap-wilds and FMoW-wilds datasets, we observed the simultaneous effect of unseen domains (OOD shift) and shifts across subpopulations, finding that the OOD shift exacerbates the gap in subpopulation performance (and vice versa). In PovertyMap-wilds, there was a drop in the overall correlation (r^2) of model predictions when tested on unseen countries. Further investigation into urban and rural subpopulations showed that shifting to unseen countries particularly widens the performance drop in rural areas, where the gap between ID and OOD r^2 doubles from 0.06 to 0.12. In FMoW-wilds, we found that accuracy on the worst region dropped precipitously (by 26 percentage points) between ID and OOD test, which contains time-shifted examples from 3-4 years in the future. Accuracy on images from Africa, which has a much smaller number of images in the dataset than other regions, was particularly affected (dropping by 36 percentage points). In FMoW-wilds, the difference in subpopulation performance (across regions) is not even manifested until also considering shift in another axis (time). In both PovertyMap-wilds and FMoW-wilds, distribution shifts across multiple axes have a compounding effect on model degradation.

8.3 Dataset specificity

Performance drops due to distribution shifts are highly specific to the task and the dataset; we consistently found that a type of shift might lead to a substantial performance drop in one dataset but not in others. For example, while we observed substantial performance gaps due to time shifts in FMoW-wilds, we observe no performance gap on time shifts in the Amazon dataset. Moreover,

when we considered the same sentiment classification task as in Amazon on a similar review dataset from Yelp, we saw modest but consistent performance drops due to time shifts, unlike in Amazon. Similarly, we saw a large variation in model performance across users in Amazon-wilds but not on the Yelp dataset. Finally, location shifts affected model performance differently across the two satellite imagery datasets, FMoW-wilds and PovertyMap-wilds; while we saw substantial performance gaps due to location shifts in PovertyMap-wilds when we simply split by location, we only saw such gaps in FMoW-wilds after splitting by time.

8.4 Diversity of training domains

Finally, we observed that domain diversity in the training data helps models generalize to unseen domains. On category shifts on the Amazon dataset, increasing the number of training categories from one to four (Books to Books, Electronics, Movies, and Home) yielded significant improvement in generalization to unseen categories. In fact, with respect to in-distribution baseline models that are trained on each target category, there were substantial performance gaps for a model trained on a single source category, but almost no gap for a model trained on four categories. This observation underscores the importance of including as many domains as possible in the training data to the extent it is realistic, since distribution shifts that might cause performance drops when there are few training domains might no longer be a problem with more diverse training sets.

9. Using the Wilds package

To facilitate algorithm development on the Wilds benchmark, we provide an open-source PyTorch-based package that exposes a simple interface to our datasets and automatically handles data downloads, allowing users to get started on a Wilds dataset in just a few lines of code. In addition, the package provides various data loaders and utilities surrounding domain annotations and other metadata, meeting the diverse requirements of the baseline algorithms. Finally, we provide standardized evaluations for each of our datasets.

Datasets and data loading. The Wilds package provides a simple, standardized interface for all datasets in the benchmark as well as their data loaders, as summarized in Figure 11. This short code snippet covers all of the steps of getting started with a Wilds dataset, including dataset download and initialization, accessing various splits, and initializing the data loader. We also provide multiple data loaders in order to accommodate a wide array of algorithms, which often require specific data loading schemes.

```
>>> from wilds.datasets.iwildcam_dataset import IWildCamDataset
>>> from wilds.common.data_loaders import get_train_loader
# Load the full dataset
>>> dataset = IWildCamDataset()
# Get the training set
>>> train_data = dataset.get_subset("train")
# Prepare the "standard" data loader
>>> train_loader = get_train_loader("standard", train_data,
...                                 batch_size=16)
# Train loop
>>> for x, y_true, metadata in train_loader:
...     ...
```

Figure 11: Dataset initialization and data loading.

Domain information. To allow algorithms to leverage domain annotations as well as other groupings over the available metadata, the Wilds package provides `Grouper` objects. `Grouper`

objects (e.g., `grouper` in Figure 12) extract group annotations from metadata, allowing users to specify the grouping scheme in a flexible fashion.

```
>>> from wilds.common.grouper import CombinatorialGrouper
# Initialize grouper, which extracts domain (location) information
>>> grouper = CombinatorialGrouper(dataset, ["location"])
# Train loop
>>> for x, y_true, metadata in train_loader:
...     z = grouper.metadata_to_group(metadata)
...     ...
```

Figure 12: Accessing domain and other group information via a Grouper object.

Evaluation. Finally, the WILDS package standardizes and automates the evaluation for each dataset. As summarized in Figure 13, invoking the `eval` method of each dataset yields all metrics reported in the paper and on the leaderboard.

```
>>> from wilds.common.data_loaders import get_eval_loader
# Get the test set
>>> test_data = dataset.get_subset("test")
# Prepare the data loader
>>> test_loader = get_eval_loader("standard", test_data,
...                               batch_size=16)
# Get predictions for the full test set
>>> for x, y_true, metadata in test_loader:
...     y_pred = model(x)
...     [ accumulate y_true, y_pred, metadata]
# Evaluate
>>> dataset.eval(all_y_pred, all_y_true, all_metadata)
{"macro_recall": 0.66, ...}
```

Figure 13: Evaluation.

Acknowledgements

Many people generously volunteered their time and expertise to advise us on WILDS. We are grateful for all of the helpful suggestions and constructive feedback from: Aditya Khosla, Andreas Schlueter, Annie Chen, Alexander D'Amour, Allison Koenecke, Alyssa Lees, Ananya Kumar, Andrew Beck, Behzad Haghgoo, Charles Sutton, Christopher Yeh, Cody Coleman, Dan Jurafsky, Daniel Levy, Daphne Koller, David Tellez, Erik Jones, Evan Liu, Fisher Yu, Georgi Marinov, Irena Gao, Irene Chen, Jacky Kang, Jacob Schreiber, Jacob Steinhardt, Jared Dunnmon, Jean Feng, Jeffrey Sorensen, Jianmo Ni, John Hewitt, Kate Saenko, Kelly Cochran, Kensen Shi, Kyle Loh, Li Jiang, Lucy Vasserman, Ludwig Schmidt, Luke Oakden-Rayner, Marco Tulio Ribeiro, Matthew Lungren, Nimit Sohoni, Pranav Rajpurkar, Robin Jia, Rohan Taori, Sarah Bird, Sharad Goel, Sherrie Wang, Stefano Ermon, Steve Yadlowsky, Tatsunori Hashimoto, Vincent Hellendoorn, Yair Carmon, Zachary Lipton, and Zhenghao Chen.

The design of the WILDS benchmark was inspired by the Open Graph Benchmark (Hu et al., 2020b), and we are grateful to the Open Graph Benchmark team for their advice and help in setting up our benchmark.

This project was funded by an Open Philanthropy Project Award and NSF Award Grant No. 1805310. S. Sagawa was supported by the Herbert Kunzel Stanford Graduate Fellowship. H. Marklund

was supported by the Dr. Tech. Marcus Wallenberg Foundation for Education in International Industrial Entrepreneurship, CIFAR, and Google. S. M. Xie and M. Zhang were supported by NDSEG Graduate Fellowships. W. Hu was supported by the Funai Overseas Scholarship and the Masason Foundation Fellowship. J. Leskovec is a Chan Zuckerberg Biohub investigator. C. Finn is a CIFAR Fellow in the Learning in Machines and Brains Program.

We also gratefully acknowledge the support of DARPA under Nos. N660011924033 (MCS); ARO under Nos. W911NF-16-1-0342 (MURI), W911NF-16-1-0171 (DURIP); NSF under Nos. OAC-1835598 (CINES), OAC-1934578 (HDR), CCF-1918940 (Expeditions), IIS-2030477 (RAPID); Stanford Data Science Initiative, Wu Tsai Neurosciences Institute, Chan Zuckerberg Biohub, Amazon, JPMorgan Chase, Docomo, Hitachi, JD.com, KDDI, NVIDIA, Dell, Toshiba, and UnitedHealth Group.

References

B. Abelson, K. R. Varshney, and J. Sun. Targeting direct cash transfers to the extremely poor. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

A. Ahadi, R. Lister, H. Haapala, and A. Vihavainen. Exploring machine learning methods to automatically identify students in need of assistance. In *Proceedings of the eleventh annual International Conference on International Computing Education Research*, pages 121–130, 2015.

Jorge A Ahumada, Eric Fegraus, Tanya Birch, Nicole Flores, Roland Kays, Timothy G O'Brien, Jonathan Palmer, Stephanie Schuttler, Jennifer Y Zhao, Walter Jetz, et al. Wildlife insights: A platform to maximize the potential of camera trap and other passive sensor wildlife data for the planet. *Environmental Conservation*, 47(1):1–6, 2020.

E. AlBadawy, A. Saha, and M. Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Med Phys.*, 45, 2018.

A. Alexandari, A. Kundaje, and A. Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning (ICML)*, pages 222–232, 2020.

Miltiadis Allamanis and Marc Brockschmidt. Smartpaste: Learning to adapt source code. *arXiv preprint arXiv:1705.07867*, 2017.

Miltiadis Allamanis, Earl T Barr, Christian Bird, and Charles Sutton. Suggesting accurate method and class names. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 38–49, 2015.

E. Amorim, M. Cançado, and A. Veloso. Automated essay scoring in the presence of biased ratings. In *Association for Computational Linguistics (ACL)*, pages 229–237, 2018.

R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. In *Language Resources and Evaluation Conference (LREC)*, pages 4218–4222, 2020.

M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Arthur Asuncion and David Newman. UCI machine learning repository, 2007.

J. Atwood, Y. Halpern, P. Baljekar, E. Breck, D. Sculley, P. Ostyakov, S. I. Nikolenko, I. Ivanov, R. Solovyev, W. Wang, et al. The inclusive images competition. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 155–186, 2020.

K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations (ICLR)*, 2019.

M. A. Badgeley, J. R. Zech, L. Oakden-Rayner, B. S. Glicksberg, M. Liu, W. Gale, M. V. McConnell, B. Percha, T. M. Snyder, and J. T. Dudley. Deep learning predicts hip fracture using confounding patient and healthcare variables. *npj Digital Medicine*, 2, 2019.

P. Bandi, O. Geessink, Q. Manson, M. V. Dijk, M. Balkenhol, M. Hermsen, B. E. Bejnordi, B. Lee, K. Paeng, A. Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018.

A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9453–9463, 2019.

P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research (JMLR)*, 9(0):1823–1840, 2008.

T. Baumann, A. Köhn, and F. Hennig. The Spoken Wikipedia Corpus collection: Harvesting, alignment and an application to hyperlistening. *Language Resources and Evaluation*, 53(2):303–329, 2019.

BBC. A-levels and gcses: How did the exam algorithm work? *The British Broadcasting Corporation*, 2020. URL https://www.bbc.com/news/explainers-53807730.

A. H. Beck, A. R. Sangoi, S. Leung, R. J. Marinelli, T. O. Nielsen, M. J. V. D. Vijver, R. B. West, M. V. D. Rijn, and D. Koller. Systematic analysis of breast cancer morphology uncovers stromal features associated with survival. *Science*, 3(108), 2011.

Axel D Becke. Perspective: Fifty years of density-functional theory in chemical physics. *The Journal of chemical physics*, 140(18):18A301, 2014.

E. Beede, E. Baylor, F. Hersch, A. Iurchenko, L. Wilcox, P. Ruamviboonsuk, and L. M. Vardoulakis. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Conference on Human Factors in Computing Systems (CHI)*, pages 1–12, 2020.

S. Beery, G. V. Horn, and P. Perona. Recognition in terra incognita. In *European Conference on Computer Vision (ECCV)*, pages 456–473, 2018.

S. Beery, E. Cole, and A. Gjoka. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020a.

Sara Beery, Dan Morris, and Siyu Yang. Efficient pipeline for camera trap image review. *arXiv preprint arXiv:1907.06772*, 2019.

Sara Beery, Guanhang Wu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Context r-cnn: Long term temporal context for per-camera object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13075–13085, 2020b.

B. E. Bejnordi, M. Veta, P. J. V. Diest, B. V. Ginneken, N. Karssemeijer, G. Litjens, J. A. V. D. Laak, M. Hermsen, Q. F. Manson, M. Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22): 2199–2210, 2017.

D. Bellamy, L. Celi, and A. L. Beam. Evaluating progress on machine learning for longitudinal electronic healthcare data. *arXiv preprint arXiv:2010.01149*, 2020.

M. G. Bellemare, S. Candido, P. S. Castro, J. Gong, M. C. Machado, S. Moitra, S. S. Ponda, and Z. Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588, 2020.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 137–144, 2006.

A. BenTaieb and G. Hamarneh. Adversarial stain transfer for histopathology image analysis. *IEEE transactions on medical imaging*, 37(3):792–802, 2017.

G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in neural information processing systems*, pages 2178–2186, 2011.

J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.

S. L. Blodgett and B. O'Connor. Racial disparity in natural language processing: A case study of social media African-American English. *arXiv preprint arXiv:1707.00061*, 2017.

S. L. Blodgett, L. Green, and B. O'Connor. Demographic dialectal variation in social media: A case study of African-American English. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1119–1130, 2016.

J. Blumenstock, G. Cadamuro, and R. On. Predicting poverty and wealth from mobile phone metadata. *Science*, 350, 2015.

D. Borkan, L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman. Limitations of pinned auc for measuring unintended bias. *arXiv preprint arXiv:1903.02088*, 2019a.

D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *WWW*, pages 491–500, 2019b.

M. Broussard. When algorithms give real students imaginary grades. *The New York Times*, 2020. URL https://www.nytimes.com/2020/09/08/opinion/international-baccalaureate-algorithm-grades.html.

L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):770–787, 2009.

D. Bug, S. Schneider, A. Grote, E. Oswald, F. Feuerhake, J. Schüler, and D. Merhof. Context-based normalization of histological stains using deep convolutional features. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 135–142, 2017.

Rudy Bunel, Matthew Hausknecht, Jacob Devlin, Rishabh Singh, and Pushmeet Kohli. Leveraging grammar and reinforcement learning for neural program synthesis. In *International Conference on Learning Representations (ICLR)*, 2018.

J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

M. Burke, S. Heft-Neal, and E. Bendavid. Sources of variation in under-5 mortality across sub-saharan africa: a spatial analysis. *Lancet Global Health*, 4, 2016.

S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

G. Campanella, M. G. Hanna, L. Geneslaw, A. Miraflor, V. W. K. Silva, K. J. Busam, E. Brogi, V. E. Reuter, D. S. Klimstra, and T. J. Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019.

K. Cao, Y. Chen, J. Lu, N. Arechiga, A. Gaidon, and T. Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*, 2020.

Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. The open catalyst 2020 (oc20) dataset and community challenges. *arXiv preprint arXiv:2010.09990*, 2020.

I. Y. Chen, P. Szolovits, and M. Ghassemi. Can AI help reduce disparities in general medical and mental health care? *AMA Journal of Ethics*, 21(2):167–179, 2019a.

I. Y. Chen, E. Pierson, S. Rose, S. Joshi, K. Ferryman, and M. Ghassemi. Ethical machine learning in health care. *arXiv preprint arXiv:2009.10576*, 2020.

V. Chen, S. Wu, A. J. Ratner, J. Weng, and C. Ré. Slice-based learning: A programming model for residual learning in critical data slices. In *Advances in neural information processing systems*, pages 9397–9407, 2019b.

G. Christie, N. Fendley, J. Wilson, and R. Mukherjee. Functional map of the world. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. *Proc. Interspeech*, pages 1086–1090, 2018.

J. H. Clark, E. Choi, M. Collins, D. Garrette, T. Kwiatkowski, V. Nikolaev, and J. Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *arXiv preprint arXiv:2003.05002*, 2020.

N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

A. Conneau and G. Lample. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7059–7069, 2019.

A. Conneau, R. Rinott, G. Lample, A. Williams, S. Bowman, H. Schwenk, and V. Stoyanov. Xnli: Evaluating cross-lingual sentence representations. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2475–2485, 2018.

L. P. Cordella, C. D. Stefano, F. Tortorella, and M. Vento. A method for improving classification reliability of multilayer perceptrons. *IEEE Transactions on Neural Networks*, 6(5):1140–1147, 1995.

P. Courtiol, C. Maussion, M. Moarii, E. Pronier, S. Pilcer, M. Sefta, P. Manceron, S. Toldo, M. Zaslavskiy, N. L. Stang, et al. Deep learning-based classification of mesothelioma improves prediction of patient outcome. *Nature medicine*, 25(10):1519–1525, 2019.

F. Croce, M. Andriushchenko, V. Sehwag, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Anne-Sophie Crunchant, David Borchers, Hjalmar Kühl, and Alex Piel. Listening and watching: Do camera traps or acoustic sensors more efficiently detect wild chimpanzees in an open habitat? *Methods in Ecology and Evolution*, 11(4):542–552, 2020.

Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019.

D. Dai and L. Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

A. D'Amour, K. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

H. Daumé III. Frustratingly easy domain adaptation. In *Association for Computational Linguistics (ACL)*, 2007.

S. E. Davis, T. A. Lasko, G. Chen, E. D. Siew, and M. E. Matheny. Calibration drift in regression and machine learning models for acute kidney injury. *Journal of the American Medical Informatics Association*, 24(6):1052–1061, 2017.

A. J. DeGrave, J. D. Janizek, and S. Lee. AI for radiographic COVID-19 detection selects shortcuts over signal. *medRxiv*, 2020.

M. C. Desmarais and R. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1):9–38, 2012.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Association for Computational Linguistics (ACL)*, pages 4171–4186, 2019.

N. DigitalGlobe and C. Works. Spacenet. *https://aws.amazon.com/publicdatasets/spacenet/*, 2016.

Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *science*, 338(6110): 1042–1046, 2012.

L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman. Measuring and mitigating unintended bias in text classification. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 67–73, 2018.

Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1–7. IEEE, 2017.

Q. Dou, D. Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

J. Duchi, T. Hashimoto, and H. Namkoong. Distributionally robust losses against mixture covariate shifts. https://cs.stanford.edu/~thashim/assets/publications/condrisk.pdf, 2019.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science (ITCS)*, pages 214–226, 2012.

C. D. Elvidge, P. C. Sutton, T. Ghosh, B. T. Tuttle, K. E. Baugh, B. Bhaduri, and E. Bright. A global poverty map derived from satellite data. *Computers and Geosciences*, 35, 2009.

J. Espey, E. Swanson, S. Badiee, Z. Chistensen, A. Fischer, M. Levy, G. Yetman, A. de Sherbinin, R. Chen, Y. Qiu, G. Greenwell, T. Klein, , J. Jutting, M. Jerven, G. Cameron, A. M. A. Rivera, V. C. Arias, , S. L. Mills, and A. Motivans. Data for development: A needs assessment for SDG monitoring and statistical capacity development. *Sustainable Development Solutions Network*, 2015.

A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

OpenAI et al. Solving Rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

C. Fang, Y. Xu, and D. N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *International Conference on Computer Vision (ICCV)*, pages 1657–1664, 2013.

J. Feng, A. Sondhi, J. Perry, and N. Simon. Selective prediction-set models with coverage guarantees. *arXiv preprint arXiv:1906.05473*, 2019.

D. Filmer and K. Scott. Assessing asset indices. *Demography*, 49, 2011.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17, 2016.

S. Garg, Y. Wu, S. Balakrishnan, and Z. C. Lipton. A unified view of label shift estimation. *arXiv preprint arXiv:2003.07554*, 2020.

Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Y. Geifman and R. El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning (ICML)*, 2019.

Y. Geifman, G. Uziel, and R. El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. In *International Conference on Learning Representations (ICLR)*, 2018.

R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

M. Geva, Y. Goldberg, and J. Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning (ICML)*, pages 1273–1272, 2017.

K. Goel, A. Gu, Y. Li, and C. Ré. Model patching: Closing the subgroup performance gap with data augmentation. *arXiv preprint arXiv:2008.06775*, 2020.

B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073, 2012.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

N. Graetz, J. Friedman, A. Osgood-Zimmerman, R. Burstein, M. H. Biehl, C. Shields, J. F. Mosser, D. C. Casey, A. Deshpande, L. Earl, R. C. Reiner, S. E. Ray, N. Fullman, A. J. Levine, R. W. Stubbs, B. K. Mayala, J. Longbottom, A. J. Browne, S. Bhatt, D. J. Weiss, P. W. Gething, A. H. Mokdad, S. S. Lim, C. J. L. Murray, E. Gakidou, and S. I. Hay. Mapping local variation in educational attainment across africa. *Nature*, 555, 2018.

M Grooten, T Peterson, and R.E.A Almond. *Living Planet Report 2020 - Bending the curve of biodiversity loss*. WWF, Gland, Switzerland, 2020.

S. Gu, E. Holly, T. Lillicrap, and S. Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *International Conference on Robotics and Automation (ICRA)*, 2017.

I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

A. Gupta, A. Murali, D. Gandhi, and L. Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener. Histopathological image analysis: A review. *IEEE reviews in biomedical engineering*, 2:147–171, 2009.

M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. High-resolution global maps of 21st-century forest cover change. *Science*, 342, 2013.

T. B. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning (ICML)*, 2018.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Y. He, Z. Shen, and P. Cui. Towards non-IID image classification: A dataset and baselines. *Pattern Recognition*, 110, 2020.

Vincent J Hellendoorn, Sebastian Proksch, Harald C Gall, and Alberto Bacchelli. When code completion fails: A case study on real-world completions. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 960–970. IEEE, 2019.

B. E. Henderson, N. H. Lee, V. Seewaldt, and H. Shen. The influence of race and ethnicity on the biology of cancer. *Nature Reviews Cancer*, 12(9):648–653, 2012.

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017.

D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2020a.

D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020b.

J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018.

D. Hovy and S. L. Spruit. The social impact of natural language processing. In *Association for Computational Linguistics (ACL)*, pages 591–598, 2016.

J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*, 2020a.

W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning (ICML)*, 2018.

W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020b.

G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. Codesearchnet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*, 2019.

N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353, 2016.

N. Jean, S. M. Xie, and S. Ermon. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Enforcing predictive invariance across structured biomedical domains. *arXiv preprint arXiv:2006.03908*, 2020.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, K. Tunyasuvunakool, O. Ronneberger, R. Bates, A. Žídek, A. Bridgland, C. Meyer, S. A A Kohl, A. Potapenko, A. J Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, M. Steinegger, M. Pacholska, D. Silver, O. Vinyals, A. W Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. High accuracy protein structure prediction using deep learning. *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction*, 2020.

Jongbin Jung, Sharad Goel, Jennifer Skeem, et al. The limits of human predictions of recidivism. *Science advances*, 6(7):eaaz0652, 2020.

A. K. Jørgensen, D. Hovy, and A. Søgaard. Challenges of studying and processing dialects in social media. In *ACL Workshop on Noisy User-generated Text*, pages 9–18, 2015.

G. Kahn, P. Abbeel, and S. Levine. BADGR: An autonomous self-supervised learning-based navigation system. *arXiv preprint arXiv:2002.05700*, 2020.

A. Kamath, R. Jia, and P. Liang. Selective question answering under domain shift. In *Association for Computational Linguistics (ACL)*, 2020.

Z. Katona, M. Painter, P. N. Patatoukas, and J. Zeng. On the capital market consequences of alternative data: Evidence from outer space. *Miami Behavioral Finance Conference*, 2018.

D. Kaushik, E. Hovy, and Z. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations (ICLR)*, 2019.

M. Kearns, S. Neel, A. Roth, and Z. S. Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning (ICML)*, pages 2564–2572, 2018.

J. H. Kim, M. Xie, N. Jean, and S. Ermon. Incorporating spatial context and fine-grained detail from satellite imagery to predict poverty. *Stanford University*, 2016a.

N. Kim and T. Linzen. Cogs: A compositional generalization challenge based on semantic interpretation. *arXiv preprint arXiv:2010.05465*, 2020.

Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1):D1202–D1213, 2016b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel. Racial disparities in automated speech recognition. *Science*, 117(14): 7684–7689, 2020.

P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*, 2020.

B. Kompa, J. Snoek, and A. Beam. Empirical frequentist coverage of deep learning uncertainty quantification procedures. *arXiv preprint arXiv:2010.03039*, 2020.

D. Komura and S. Ishikawa. Machine learning methods for histopathological image analysis. *Computational and structural biotechnology journal*, 16:34–42, 2018.

Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy S Liang. Spoc: Search-based pseudocode to code. In *Advances in Neural Information Processing Systems*, pages 11906–11917, 2019.

C. Kulkarni, P. W. Koh, H. Huy, D. Chia, K. Papadopoulos, J. Cheng, D. Koller, and S. R. Klemmer. Peer and self assessment in massive online classes. *Design Thinking Research*, pages 131–168, 2015.

C. E. Kulkarni, R. Socher, M. S. Bernstein, and S. R. Klemmer. Scaling short-answer grading by combining peer assessment with algorithmic scoring. In *Proceedings of the first ACM conference on Learning@Scale conference*, pages 99–108, 2014.

A. Kumar, T. Ma, and P. Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

B. Lake and M. Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning (ICML)*, 2018.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.

Agostina J Larrazabal, Nicolás Nieto, Victoria Peterson, Diego H Milone, and Enzo Ferrante. Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. *Proceedings of the National Academy of Sciences*, 117(23):12592–12594, 2020.

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica*, 9(1), 2016.

R. Y. Lau, C. Li, and S. S. Liao. Social analytics: Learning fuzzy product ontologies for aspect-oriented sentiment analysis. *Decision Support Systems*, 65:80–94, 2014.

Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

J. T. Leek, R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. Baggerly, and R. A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10), 2010.

D. Li, Y. Yang, Y. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017a.

H. Li, D. Quang, and Y. Guan. Anchor: trans-cell type prediction of transcription factor binding sites. *Genome research*, 29(2):281–292, 2019a.

J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston. Learning through dialogue interactions by asking questions. In *International Conference on Learning Representations (ICLR)*, 2017b.

T. Li, M. Sanjabi, A. Beirami, and V. Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019b.

Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. In *International Conference on Learning Representations Workshop (ICLRW)*, 2017c.

S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Z. Lipton, Y. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*, 2018.

Y. Liu, K. Gadepalli, M. Norouzi, G. E. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Q. Nelson, G. S. Corrado, et al. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442*, 2017.

M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105, 2015.

M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1107–1110, 2009.

Brian A Malloy and James F Power. Quantifying the transition from python 2 to 3: an empirical study of python applications. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 314–323. IEEE, 2017.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330, 1993.

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. *arXiv preprint arXiv:1911.02969*, 2019a.

R. T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL)*, 2019b.

S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. C. Corrado, A. Darzi, et al. International evaluation of an AI system for breast cancer screening. *Nature*, 577(7788):89–94, 2020.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, C. Kavukcuoglu, D. Kumaran, and R. Hadsell. Learning to navigate in complex environments. In *International Conference on Learning Representations (ICLR)*, 2017.

John Moult, Jan T Pedersen, Richard Judson, and Krzysztof Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 23 (3):ii–iv, 1995.

K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, pages 10–18, 2013.

W. Nekoto, V. Marivate, T. Matsila, T. Fasubaa, T. Kolawole, T. Fagbohungbe, S. O. Akinola, S. H. Muhammad, S. Kabongo, S. Osei, S. Freshia, R. A. Niyongabo, R. Macharm, P. Ogayo, O. Ahia, M. Meressa, M. Adeyemi, M. Mokgesi-Selinga, L. Okegbemi, L. J. Martinus, K. Tajudeen, K. Degila, K. Ogueji, K. Siminyu, J. Kreutzer, J. Webster, J. T. Ali, J. Abbott, I. Orife, I. Ezeani, I. A.

Dangana, H. Kamper, H. Elsahar, G. Duru, G. Kioko, E. Murhabazi, E. van Biljon, D. Whitenack, C. Onyefuluchi, C. Emezue, B. Dossou, B. Sibanda, B. I. Bassey, A. Olabiyi, A. Ramkilowan, A. Öktem, A. Akinfaderin, and A. Bashir. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of Empirical Methods in Natural Language Processing (Findings of EMNLP)*, 2020.

B. Nestor, M. McDermott, W. Boag, G. Berner, T. Naumann, M. C. Hughes, A. Goldenberg, and M. Ghassemi. Feature robustness in non-stationary health records: caveats to deployable model performance in common clinical machine learning tasks. *arXiv preprint arXiv:1908.00690*, 2019.

J. Ni, J. Li, and J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 188–197, 2019.

Marius Nita and David Notkin. Using twinning to adapt programs to alternative apis. In *2010 ACM/IEEE 32nd International Conference on Software Engineering*, volume 1, pages 205–214. IEEE, 2010.

A. Noor, V. Alegana, P. Gething, A. Tatem, and R. Snow. Using remotely sensed night-time light as a proxy for poverty in africa. *Population Health Metrics*, 6, 2008.

Mohammad Sadegh Norouzzadeh, Dan Morris, Sara Beery, Neel Joshi, Nebojsa Jojic, and Jeff Clune. A deep active learning system for species identification and counting in camera trap images. *arXiv preprint arXiv:1910.09716*, 2019.

NYTimes. The Times is partnering with Jigsaw to expand comment capabilities. *The New York Times*, 2016. URL https://www.nytco.com/press/the-times-is-partnering-with-jigsaw-to-expand-comment-capabilities/.

Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

Y. Oren, S. Sagawa, T. Hashimoto, and P. Liang. Distributionally robust language modeling. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

A. Osgood-Zimmerman, A. I. Millear, R. W. Stubbs, C. Shields, B. V. Pickering, L. Earl, N. Graetz, D. K. Kinyoki, S. E. Ray, S. Bhatt, A. J. Browne, R. Burstein, E. Cameron, D. C. Casey, A. Deshpande, N. Fullman, P. W. Gething, H. S. Gibson, N. J. Henry, M. Herrero, L. K. Krause, I. D. Letourneau, A. J. Levine, P. Y. Liu, J. Longbottom, B. K. Mayala, J. F. Mosser, A. M. Noor, D. M. Pigott, E. G. Piwoz, P. Rao, R. Rawat, R. C. Reiner, D. L. Smith, D. J. Weiss, K. E. Wiens, A. H. Mokdad, S. S. Lim, C. J. L. Murray, N. J. Kassebaum, and S. I. Hay. Mapping child growth failure in africa between 2000 and 2015. *Nature*, 555, 2018.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

S. J. Pan, X. Ni, J. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760, 2010.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

Jason Parham, Jonathan Crall, Charles Stewart, Tanya Berger-Wolf, and Daniel I Rubenstein. Animal population censusing at scale with citizen science and photographic identification. In *AAAI Spring Symposium-Technical Report*, 2017.

J. H. Park, J. Shin, and P. Fung. Reducing gender bias in abusive language detection. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2799–2804, 2018.

G. K. Patro, A. Biswas, N. Ganguly, K. P. Gummadi, and A. Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of The Web Conference 2020*, pages 1194–1204, 2020.

X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2021–2026, 2018.

X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *International Conference on Computer Vision (ICCV)*, 2019.

X. Peng, E. Coumans, T. Zhang, T. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems (RSS)*, 2020.

L. Perelman. When "the state of the art" is counting words. *Assessing Writing*, 21:104–111, 2014.

N. A. Phillips, P. Rajpurkar, M. Sabini, R. Krishnan, S. Zhou, A. Pareek, N. M. Phu, C. Wang, A. Y. Ng, and M. P. Lungren. Chexphoto: 10,000+ smartphone photos and synthetic photographic transformations of chest x-rays for benchmarking deep learning robustness. *arXiv preprint arXiv:2007.06199*, 2020.

C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. Tuned models of peer assessment in moocs. *Educational Data Mining*, 2013.

M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

Kerrie A Pipal, Jeremy J Notch, Sean A Hayes, and Peter B Adams. Estimating escapement for a low-abundance steelhead population using dual-frequency identification sonar (didson). *North American Journal of Fisheries Management*, 32(5):880–893, 2012.

W Nicholson Price and I Glenn Cohen. Privacy in the age of medical big data. *Nature medicine*, 25 (1):37–43, 2019.

Veselin Raychev, Martin Vechev, and Eran Yahav. Code completion with statistical language models. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 419–428, 2014.

C. Ré, F. Niu, P. Gudipati, and C. Srisuwananukorn. Overton: A data system for monitoring and improving machine-learned products. *arXiv preprint arXiv:1909.05372*, 2019.

R. C. Reiner, N. Graetz, D. C. Casey, C. Troeger, G. M. Garcia, J. F. Mosser, A. Deshpande, S. J. Swartz, S. E. Ray, B. F. Blacker, P. C. Rao, A. Osgood-Zimmerman, R. Burstein, D. M. Pigott, I. M. Davis, I. D. Letourneau, L. Earl, J. M. Ross, I. A. Khalil, T. H. Farag, O. J. Brady, M. U. Kraemer, D. L. Smith, S. Bhatt, D. J. Weiss, P. W. Gething, N. J. Kassebaum, A. H. Mokdad, C. J. Murray, and S. I. Hay. Variation in childhood diarrheal morbidity and mortality in africa, 2000–2015. *New England Journal of Medicine*, 379, 2018.

M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Association for Computational Linguistics (ACL)*, pages 4902–4912, 2020.

S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118, 2016.

E. Rolf, M. I. Jordan, and B. Recht. Post-estimation smoothing: A simple baseline for learning with side information. In *Artificial Intelligence and Statistics (AISTATS)*, 2020.

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

F. Sadeghi and S. Levine. CAD2RL: Real single-image flight without a single real image. In *Robotics: Science and Systems (RSS)*, 2017.

K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226, 2010.

M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

D. E. Sahn and D. Stifel. Exploring alternative measures of welfare in the absence of expenditure data. *The Review of Income and Wealth*, 49, 2003.

S. Santurkar, D. Tsipras, and A. Madry. Breeds: Benchmarks for subpopulation shift. *arXiv*, 2020.

Stefan Schneider and Alex Zhuang. Counting fish and dolphins in sonar images using deep learning. *arXiv preprint arXiv:2007.12808*, 2020.

L. Seyyed-Kalantari, G. Liu, M. McDermott, and M. Ghassemi. Chexclusion: Fairness gaps in deep chest X-ray classifiers. *arXiv preprint arXiv:2003.00827*, 2020.

Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D Sculley. No classification without representation: Assessing geodiversity issues in open data sets for the developing world. *Advances in Neural Information Processing Systems (NeurIPS) Workshop on Machine Learning for the Developing World*, 2017.

V. Shankar, A. Dave, R. Roelofs, D. Ramanan, B. Recht, and L. Schmidt. Do image classifiers generalize across time? *arXiv preprint arXiv:1906.02168*, 2019.

J. Shen, Y. Qu, W. Zhang, and Y. Yu. Wasserstein distance guided representation learning for domain adaptation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

M. D. Shermis. State-of-the-art automated essay scoring: Competition, results, and future directions from a united states demonstration. *Assessing Writing*, 20:53–76, 2014.

Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8218–8226, 2019.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 2000.

Richard Shin, Neel Kant, Kavi Gupta, Christopher Bender, Brandon Trabucco, Rishabh Singh, and Dawn Song. Synthetic datasets for neural program synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

Yu Shiu, KJ Palmer, Marie A Roch, Erica Fleishman, Xiaobai Liu, Eva-Marie Nosal, Tyler Helble, Danielle Cholewiak, Douglas Gillespie, and Holger Klinck. Deep neural networks for automated detection of marine mammal species. *Scientific Reports*, 10(1):1–12, 2020.

Brian K Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, 2004.

N. Sohoni, J. Dunnmon, G. Angus, A. Gu, and C. Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Teague Sterling and John J. Irwin. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015. doi: 10.1021/acs.jcim.5b00559. PMID: 26479676.

Dan Stowell, Michael D Wood, Hanna Pamuła, Yannis Stylianou, and Hervé Glotin. Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge. *Methods in Ecology and Evolution*, 10(3):368–380, 2019.

A. Subbaswamy, R. Adams, and S. Saria. Evaluating model robustness to dataset shift. *arXiv preprint arXiv:2010.15100*, 2020.

B. Sun and K. Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450, 2016.

B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2016.

P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, S. Zhao, S. Cheng, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020a.

Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros, and M. Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020b.

Alexey Svyatkovskiy, Ying Zhao, Shengyu Fu, and Neel Sundaresan. Pythia: ai-assisted code completion system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2727–2735, 2019.

Michael A Tabak, Mohammad S Norouzzadeh, David W Wolfson, Steven J Sweeney, Kurt C VerCauteren, Nathan P Snow, Joseph M Halseth, Paul A Di Salvo, Jesse S Lewis, Michael D White, et al. Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods in Ecology and Evolution*, 10(4):585–590, 2019.

K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891, 2016.

R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644*, 2020.

R. Tatman. Gender and dialect bias in YouTube's automatic captions. In *Workshop on Ethics in Natural Langauge Processing*, volume 1, pages 53–59, 2017.

D. Tellez, M. Balkenhol, I. Otte-Höller, R. van de Loo, R. Vogels, P. Bult, C. Wauters, W. Vreuls, S. Mol, N. Karssemeijer, et al. Whole-slide mitosis detection in h&e breast histology using phh3 as a reference to train distilled stain-invariant convolutional networks. *IEEE transactions on medical imaging*, 37(9):2126–2136, 2018.

D. Tellez, G. Litjens, P. Bándi, W. Bulten, J. Bokhorst, F. Ciompi, and J. van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical image analysis*, 58, 2019.

Dogancan Temel, Jinsol Lee, and Ghassan AlRegib. Cure-or: Challenging unreal and real environments for object recognition. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 137–144. IEEE, 2018.

T. G. Tiecke, X. Liu, A. Zhang, A. Gros, N. Li, G. Yetman, T. Kilic, S. Murray, B. Blankespoor, E. B. Prydz, and H. H. Dang. Mapping the world population one building at a time. *arXiv*, 2017.

J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528, 2011.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

B. Uzkent and S. Ermon. Learning when and where to zoom with deep reinforcement learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Marko Vasic, Aditya Kanade, Petros Maniatis, David Bieber, and Rishabh Singh. Neural program repair by jointly learning to localize and repair. In *International Conference on Learning Representations (ICLR)*, 2019.

Sindre Vatnehol, Hector Peña, and Nils Olav Handegard. A method to automatically detect fish aggregations using horizontally scanning sonar. *ICES Journal of Marine Science*, 75(5):1803–1812, 2018.

B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pages 210–218, 2018.

H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5018–5027, 2017.

M. Veta, P. J. V. Diest, M. Jiwa, S. Al-Janabi, and J. P. Pluim. Mitosis counting in breast cancer: Object-level interobserver agreement and comparison to an automatic method. *PloS one*, 11(8), 2016.

M. Veta, Y. J. Heng, N. Stathonikos, B. E. Bejnordi, F. Beca, T. Wollmann, K. Rohr, M. A. Shah, D. Wang, M. Rousson, et al. Predicting breast tumor proliferation from whole-slide images: the tupac16 challenge. *Medical image analysis*, 54:111–121, 2019.

R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020a.

S. Wang, M. Bai, G. Mattyus, H. Chu, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun. Torontocity: Seeing the world with a million eyes. In *International Conference on Computer Vision (ICCV)*, 2017.

S. Wang, W. Chen, S. M. Xie, G. Azzari, and D. B. Lobell. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing*, 12, 2020b.

OR Wearn and P Glover-Kapfer. Camera-trapping for conservation: a guide to best-practices. *WWF conservation technology series*, 1(1):2019–04, 2017.

S. Weinberger. Speech accent archive. *George Mason University*, 2015.

Ben G Weinstein. A computer vision for animal ecology. *Journal of Animal Ecology*, 87(3):533–545, 2018.

J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10), 2013.

R. West, H. S. Paskov, J. Leskovec, and C. Potts. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics (TACL)*, 2:297–310, 2014.

J. J. Williams, J. Kim, A. Rafferty, S. Maldonado, K. Z. Gajos, W. S. Lasecki, and N. Heffernan. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@Scale*, pages 379–388, 2016.

Benjamin Wilson, Judy Hoffman, and Jamie Morgenstern. Predictive inequity in object detection. *arXiv preprint arXiv:1902.11097*, 2019.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. HuggingFace's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Ho Yuen Frank Wong, Hiu Yin Sonia Lam, Ambrose Ho-Tung Fong, Siu Ting Leung, Thomas Wing-Yan Chin, Christine Shing Yen Lo, Macy Mei-Sze Lui, Jonan Chun Yin Lee, Keith Wan-Hang Chiu, Tom Chung, et al. Frequency and distribution of chest radiographic findings in covid-19 positive patients. *Radiology*, page 201160, 2020.

D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5028–5037, 2017.

M. Wu, M. Mosse, N. Goodman, and C. Piech. Zero shot learning for code education: Rubric sampling with deep learning inference. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 33, pages 782–790, 2019a.

M. Wu, R. L. Davis, B. W. Domingue, C. Piech, and N. Goodman. Variational item response theory: Fast, accurate, and expressive. *International Conference on Educational Data Mining*, 2020.

Y. Wu, E. Winston, D. Kaushik, and Z. Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning (ICML)*, pages 6872–6881, 2019b.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

M. Wulfmeier, A. Bewley, and I. Posner. Incremental adversarial domain adaptation for continually changing environments. In *International Conference on Robotics and Automation (ICRA)*, 2018.

K. Xiao, L. Engstrom, A. Ilyas, and A. Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

M. Xie, N. Jean, M. Burke, D. Lobell, and S. Ermon. Transfer learning from deep features for remote sensing and poverty mapping. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2016.

S. M. Xie, A. Kumar, R. Jones, F. Khani, T. Ma, and P. Liang. In-N-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. *arXiv*, 2020.

K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2018.

Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. *Geographic Information Systems*, 2010.

Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, and V. Sindhwani. Data efficient reinforcement learning for legged robots. In *Conference on Robot Learning (CoRL)*, 2019.

Michihiro Yasunaga and Percy Liang. Graph-based, self-supervised program repair from diagnostic feedback. In *International Conference on Machine Learning (ICML)*, 2020.

C. Yeh, A. Perez, A. Driscoll, G. Azzari, Z. Tang, D. Lobell, S. Ermon, and M. Burke. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature Communications*, 11, 2020.

J. You, X. Li, M. Low, D. Lobell, and S. Ermon. Deep gaussian process for crop yield prediction based on remote sensing data. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

N. Yuval, W. Tao, C. Adam, B. Alessandro, W. Bo, and N. A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. In *PLOS Medicine*, 2018.

K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning (ICML)*, pages 819–827, 2013.

M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. *arXiv preprint arXiv:2007.02931*, 2020.

Y. Zhang, J. Baldridge, and L. He. Paws: Paraphrase adversaries from word scrambling. In *North American Association for Computational Linguistics (NAACL)*, 2019.

X. Zhou, Y. Nie, H. Tan, and M. Bansal. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. *arXiv preprint arXiv:2004.13606*, 2020.

C Lawrence Zitnick, Lowik Chanussot, Abhishek Das, Siddharth Goyal, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Thibaut Lavril, Aini Palizhati, Morgane Riviere, et al. An introduction to electrocatalyst design using machine learning for renewable energy storage. *arXiv preprint arXiv:2010.09435*, 2020.

Appendix A. Additional experimental details

A.1 Model hyperparameters

For each hyperparameter setting, we use early stopping to pick the epoch with the best validation performance (as measured by the specified metrics for each dataset, which are described in Section 5), and then pick the model hyperparameters with the best early-stopped validation performance. Note that even if the model does not use additional metadata (e.g., for ERM), the dataset metric might implicitly use this metadata for model selection (e.g., by selecting the model that has the best accuracy over different subpopulations defined in the metadata).

In general, we select model hyperparameters with ERM and use the same hyperparameters for the other algorithm baselines (e.g., DeepCORAL, IRM, or Group DRO). For DeepCORAL and IRM, we do a subsequent grid search over the weight of the penalty term, using the defaults from Gulrajani and Lopez-Paz (2020). Specifically, we try penalty weights of $\{0.1, 1, 10\}$ for DeepCORAL and penalty weights of $\{1, 10, 100, 1000\}$ for IRM.

A.2 In-distribution validation and test sets

In addition to the OOD validation and test sets, where possible, we also provide an in-distribution (ID) validation and/or test set to measure ID performance. While we do not use the ID validation set for model selection, users are free to use the ID validation set for hyperparameter tuning (see discussion in Section 8). For example, in the iWildCam2020-wilds dataset mentioned above, the ID validation set comprises photos from the same set of camera traps used for the training set. With a couple of exceptions, we typically use a fixed train/val/test split and report results averaged across 3 replicates (random seeds for model initialization and minibatch order), as well as the unbiased standard deviation over those replicates. For PovertyMap-wilds, we report results averaged over 5-fold cross validation, as model training is relatively fast on this dataset. For Camelyon17-wilds, results vary substantially between replicates, so we report results averaged over 10 replicates instead.

Appendix B. Additional dataset details

B.1 FMoW-wilds

The FMoW-wilds dataset is derived from Christie et al. (2018), which collects over 1 million satellite images from over 200 countries over 2002-2018. We use the RGB version of the original dataset,

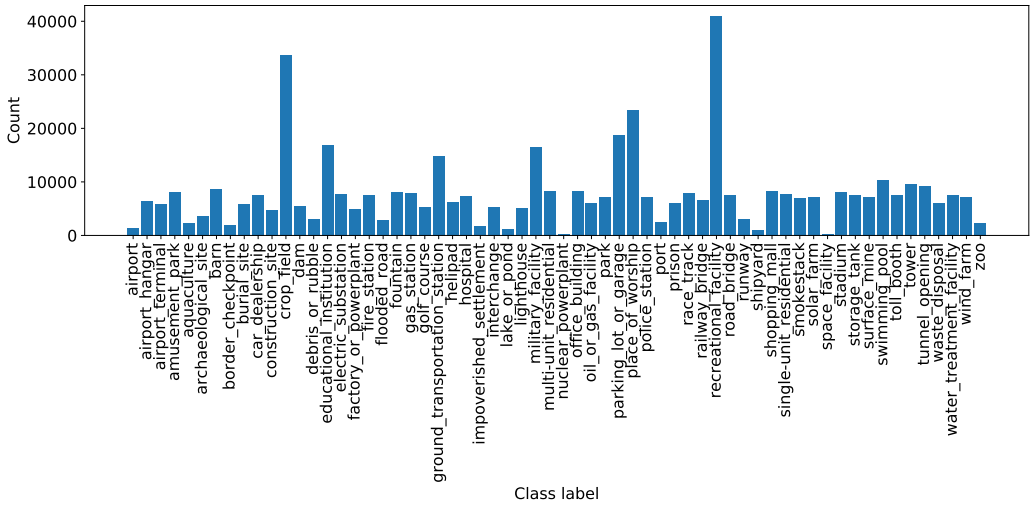

Figure 14: Number of examples from each category in FMoW-wilds in non-African regions.

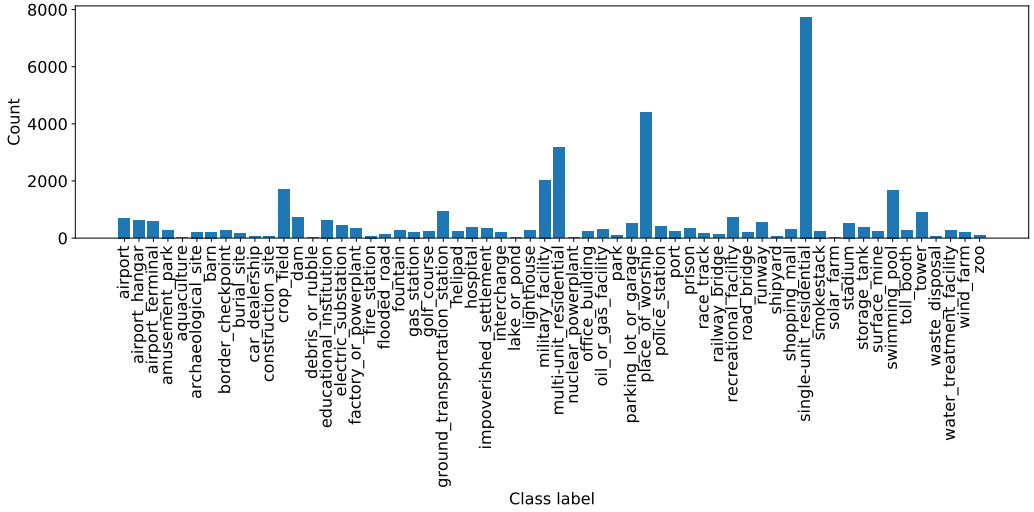

Figure 15: Number of examples from each category in FMoW-wilds in Africa. There is a large label shift between non-African regions and Africa.

	Training	Val (ID)	Val (OOD) (2013-2016)	Test (ID) (<2013)	Test (OOD) (≥ 2016)	Last year (2017)
ERM	100 (0.0)	61.6 (0.21)	59.7 (0.14)	59.9 (0.27)	53.1 (0.25)	48.3 (0.60)
DeepCORAL	100 (0.0)	58.7 (0.44)	56.7 (0.06)	57.1 (0.15)	50.5 (0.30)	45.8 (0.32)
IRM	99.4 (0.06)	59.2 (0.31)	57.2 (0.01)	58.0 (0.25)	50.9 (0.32)	46.2 (0.21)
ERM (Mixed split)	100 (0.0)	61.8 (0.21)	65.0 (0.25)	59.3 (0.26)	57.6 (0.44)	54.4 (0.91)

Table 17: Time shift accuracies (%) for models trained on data before 2013 and tested on held-out locations from in-distribution (ID) or out-of-distribution (OOD) test sets in FMoW-wilds. The accuracy of ERM drops significantly in the last year of the dataset. The models are early-stopped with respect to OOD validation accuracy. Standard deviations over 3 trials are in parentheses. Mixed split models use both ID + OOD training examples.

which contains 523,846 total examples, excluding the multispectral version of the images. Methods that can utilize a sequence of images can group the images from the same location across multiple years together as input, but we consider the simple formulation here for our baseline evaluation.

Model. We use a Densenet-121 model (Huang et al., 2017) pretrained on ImageNet for 62-way classification. We train to minimize cross entropy loss using the Adam optimizer, decaying the learning rate by 0.96 per epoch (initial learning rate 1e-4). We use a batch size of 64 and train for 50 epochs, using early stopping with an in-distribution development set. All results reported in the tables are averaged over 3 random seeds.

Data processing and modifications to the original dataset. The original dataset from Christie et al. (2018) is provided as a set of hierarchical directories with JPEG images of varying sizes. We process the images as a collection of 100 NumPy arrays, where can we use the fast memory mapped reading mode to load our training set quickly. We also collect all the metadata into CSV format for easy processing.

The original dataset is posed as a image time-series classification problem, where the model has access to a sequence of images at each location. For simplicity, we treat each image as a separate example, while making sure that the data splits all contain disjoint locations. We use the ID train/val/test splits from the original dataset, but use two OOD time segments: we treat data from 2013-2016 as OOD val and 2016-2018 as OOD test. This reduces the size of the training set in comparison to the original dataset.

Label shift between non-African regions and Africa. Figures 14 and 15 plot the category frequencies of examples restricted to non-African regions or Africa-only. We find that there is a large label shift when moving to Africa, especially with a drop in recreational facilities and a large increase in single residential units. We do not find a similarly large label shift between < 2013 and ≥ 2013 splits of the dataset.

Additional challenges in high-resolution satellite datasets. Compared to PovertyMap-wilds, FMoW-wilds contains much higher resolution images (sub-meter resolution vs. 30m resolution) and contains a larger variety of viewpoints/tilts, both of which could present computational or algorithmic challenges. For computational purposes, we resized all images to 224×224 (following Christie et al. (2018)), but raw images can be thousands of pixels wide. Some recent works have tried to balance this tradeoff between viewing overall context and the fine-grained detail (Uzkent and Ermon, 2020; Kim et al., 2016a), but how best to do this is an open question. FMoW-wilds also contains additional information on azimuth and cloud cover which could be used to correct for the variety in viewpoints and image quality.

Additional baseline methods. For DeepCORAL, we used a penalty weight of 0.1. For IRM, we used $\lambda = 1$. These hyperparameters were chosen over a grid search over DeepCORAL penalty weight of $\{0.1, 1, 10\}$ and IRM $\lambda \in \{1, 10, 100, 1000\}$, taking the best over the OOD validation set.

B.2 PovertyMap-wilds

The PovertyMap-wilds dataset is derived from Yeh et al. (2020), which gathers LandSat imagery and Demographic and Health Surveys (DHS) data from 19669 villages in Africa across 23 countries. The images are 224×224 pixels large over 7 multispectral channels, with an optional eighth nighttime light intensity channel. The LandSat satellite has a 30m resolution, meaning that each pixel of the image covers a $30m^2$ spatial area. The location metadata is perturbed by the DHS as a privacy protection scheme; urban locations are randomly displaced by up to 2km and rural locations are perturbed by up to 10km. This adds some noise to the data, but having a large enough image can guarantee that the location is in the image most of the time. The target is a real-valued composite asset wealth index computed as the first principal component of survey responses about household assets, which is thought to be a less noisy measure of households' longer-run economic well-being

	Train	ID Val	OOD Val	ID Test	OOD Test
Overall					
ERM - NL	0.80 (0.12)	0.66 (0.02)	0.55 (0.05)	0.67 (0.03)	0.55 (0.06)
ERM (Mixed split) - NL	0.83 (0.16)	0.64 (0.03)	0.64 (0.06)	0.62 (0.03)	0.63 (0.05)
ERM	0.72 (0.03)	0.67 (0.02)	0.65 (0.06)	0.68 (0.03)	0.62 (0.05)
ERM (Mixed split)	0.75 (0.04)	0.70 (0.03)	0.70 (0.06)	0.69 (0.01)	0.69 (0.06)
DeepCORAL	0.75 (0.06)	0.65 (0.04)	0.66 (0.07)	0.68 (0.03)	0.61 (0.07)
IRM	0.73 (0.03)	0.67 (0.03)	0.65 (0.07)	0.68 (0.05)	0.61 (0.04)
Rural subpopulation					
ERM - NL	0.57 (0.25)	0.31 (0.02)	0.16 (0.07)	0.32 (0.04)	0.15 (0.07)
ERM (Mixed split) - NL	0.64 (0.33)	0.28 (0.05)	0.29 (0.09)	0.26 (0.03)	0.28 (0.08)
ERM	0.39 (0.06)	0.32 (0.05)	0.26 (0.06)	0.34 (0.08)	0.22 (0.07)
ERM (Mixed split)	0.48 (0.08)	0.37 (0.06)	0.38 (0.08)	0.35 (0.07)	0.35 (0.07)
DeepCORAL	0.42 (0.18)	0.28 (0.07)	0.27 (0.10)	0.29 (0.10)	0.20 (0.08)
IRM	0.39 (0.09)	0.31 (0.07)	0.29 (0.05)	0.32 (0.07)	0.24 (0.06)
Urban subpopulation					
ERM - NL	0.66 (0.20)	0.42 (0.02)	0.29 (0.07)	0.42 (0.05)	0.29 (0.05)
ERM (Mixed split) - NL	0.70 (0.27)	0.36 (0.04)	0.41 (0.10)	0.38 (0.04)	0.38 (0.07)
ERM	0.51 (0.04)	0.42 (0.05)	0.39 (0.10)	0.42 (0.05)	0.35 (0.10)
ERM (Mixed split)	0.53 (0.06)	0.42 (0.05)	0.47 (0.11)	0.43 (0.03)	0.43 (0.07)
DeepCORAL	0.55 (0.11)	0.39 (0.03)	0.44 (0.08)	0.40 (0.03)	0.34 (0.09)
IRM	0.51 (0.04)	0.42 (0.06)	0.38 (0.12)	0.42 (0.06)	0.35 (0.11)

Table 18: Squared Pearson correlation r^2 (higher is better) on in-distribution and out-of-distribution (unseen countries) held-out sets in PovertyMap-wilds, including results on rural or urban subpopulations. All results are averaged over 5 different OOD country folds taken from Yeh et al. (2020), with standard deviations across different folds in parentheses. All models are early-stopped with respect to OOD validation MSE. (- NL) models do not use nighttime light as input. Mixed split models use both ID + OOD examples as training data.

	Train	ID Val	OOD Val	ID Test	OOD Test
Baseline - NL	0.141 (0.086)	0.229 (0.033)	0.295 (0.046)	0.225 (0.040)	0.308 (0.030)
ERM (Mixed split) - NL	0.113 (0.102)	0.242 (0.016)	0.230 (0.032)	0.252 (0.022)	0.239 (0.016)
Baseline	0.185 (0.023)	0.216 (0.016)	0.234 (0.045)	0.214 (0.028)	0.259 (0.045)
ERM (Mixed split)	0.166 (0.026)	0.202 (0.021)	0.192 (0.028)	0.211 (0.007)	0.202 (0.043)
DeepCORAL	0.171 (0.041)	0.226 (0.033)	0.221 (0.047)	0.216 (0.019)	0.278 (0.034)
IRM	0.184 (0.023)	0.210 (0.034)	0.246 (0.017)	0.213 (0.041)	0.249 (0.047)

Table 19: Mean squared error (MSE) on in-distribution and out-of-distribution (unseen countries) held-out sets in PovertyMap-wilds. All results are averaged over 5 folds taken from Yeh et al. (2020). All models are early-stopped with respect to OOD validation MSE. (- NL) models do not use nighttime light as input. Mixed split models use both ID + OOD examples as training data.

than other welfare measurements such as consumption expenditure (Sahn and Stifel, 2003; Filmer and Scott, 2011). Asset wealth also has the advantage of not requiring adjustments for inflation or for purchasing power parity (PPP), as it is not based on a currency.

Model. Following (Yeh et al., 2020), we use a ResNet-18 model (He et al., 2016) trained with the Adam optimizer and mean squared-error loss function. We use almost all the same hyperparameters, including a batch size of 64 and decaying the learning rate by a factor of 0.96 after each epoch (initial learning rate 1e-3). We train for 200 epochs with early stopping on the OOD validation set.

Data processing and augmentation. We normalize each channel by the pixel-wise mean and standard deviation for each channel, following (Yeh et al., 2020). We also do a similar data augmentation scheme, adding random horizontal and vertical flips as well as color jitter (brightness factor 0.8, contrast factor 0.8, saturation factor 0.8, hue factor 0.1).

The data download process provided by Yeh et al. (2020) involves downloading and processing imagery from Google Earth Engine. We process all the imagery into a single NumPy array. We also provide all the metadata in a CSV format. We will provide a PyTorch implementation of a dataset loader and a baseline model training pipeline.

Additional baseline methods. For deepCORAL, we used a penalty weight of 10. For IRM, we used $\lambda = 1$. These hyperparameters were chosen over a grid search over deepCORAL penalty weight of $\{0.1, 1, 10\}$ and IRM $\lambda \in \{1, 10, 100, 1000\}$, taking the best over the OOD validation set.

Modifications to the original dataset. We see a much larger drop due to spatial shift than in Yeh et al. (2020). To explain this, we note that our data splitting method is slightly different to theirs. Since they have two separate experiments (with different data splits) to test in-distribution vs. out-of-distribution generalization, we compare both metrics simultaneously on one model as a more direct comparison. We use the same OOD country folds as the original dataset. However, Yeh et al. (2020) split the ID train/val/test while making sure that the spatial extent of the images within each split never overlap, while we simply take uniformly random splits of the ID data.

B.3 iWildCam2020-wilds

Additional dataset details. We generate the splits in three steps. First, to generate the OOD splits, we randomly split all locations into three groups: Validation (OOD), Test (OOD), and Others. Then, to generate the ID splits, we split Others by date into three sets: Training, Validation (ID), and Test (ID).

When doing the ID split according to date, some locations only ended up in some of but not all of Training, Validation (ID), and Test (ID). For instance, if there were very few dates for a specific location (camera trap), it may be that no examples from that location ended up in the train split. This defeats the purpose of the ID split, which is to test performance on locations that were seen during training. Thus, these locations were removed. Finally, any images in the test set with classes not present in the train set were also removed.

Split	# Examples	# Camera traps
Training	142,202	245
Validation (ID)	7,819	223
Test (ID)	7,861	224
Validation (OOD)	20,784	32
Test (OOD)	38,943	47

Table 20: Dataset details for iWildCam2020-wilds.

Modifications to the original dataset. In the competition on Kaggle there is a held-out test set that we are not utilizing, as the test set is intended to be reused in a future competition and is not yet public. Instead, we constructed our own test set by splitting the Kaggle competition training data into our own splits: train, validation (ID), validation (OOD), test (ID), test (OOD). Moreover, as we mentioned above, we leave out 49 locations that did not span ID splits.

Images are organized into sequences, but we treat each image separately. In the iWildCam 2020 competition, the top participants utilize the sequence data and also use a pretrained MegaDetector animal detection model that outputs bounding boxes over the animals. These images are cropped using the bounding boxes and then fed into a classification network. As we discuss in Section 5.3, we intentionally do not use MegaDetector in our experiments.

Baseline model details. We train a Resnet-50 with batch size 16 for 18 epochs, on images resized to 224 by 224. We pick hyperparameters by doing a grid search over different learning rates, $1e-3$, $1e-4$ and $1e-5$ and different weight decay, 0, $1e-4$ and $1e-5$. The optimizer is Adam. We pick the best hyperparameters and run 3 seeds.

When training the DeepCoral baseline, we use the best best learning rate and weight decay from ERM. To pick the penalty weight we do a grid search over 0.1, 1, and 10.

Additional discussion Beery et al. (2018) studied shifts in camera traps with the Caltech Camera Traps-20 dataset (CCT-20) and showed that there were considerable performance drops when evaluating the model on held-out test locations. They identified several classification challenges arising from camera trap images, including poor illumination, motion blur, and size of the animal in the image, which could account for the large drop in performance. They show that making use of an animal detection model that first locates the animal, followed by classification can significantly reduce the generalization gap. However, this requires the collection of bounding box annotations, either using an already trained animal detection model or having humans annotate the images, which is costly, and non-trivial. Since there is already much data with image-level species labels, whilst not nearly as much data for training animal detection models, it would be useful to be able to train models that directly classify images without relying on the intermediate step of first detecting the animal. It has also been shown that utilizing the temporal signal, for instance, taking the median prediction across a burst of images captured for a single motion trigger can reduce the gap.

B.4 Camelyon17-wilds

Additional dataset details. The Camelyon17-wilds dataset is adapted from whole-slide images (WSIs) of breast cancer metastases in lymph nodes sections, obtained from the CAMELYON17 challenge (Bandi et al., 2018). The splits are described in Section 5.4. Each split is balanced to have an equal number of positive and negative examples. The varying number of patches per slide and hospital is due to this class balancing, as some slides have fewer tumor (positive) patches. We selected the test set hospital as the one whose patches were visually most distinct; the difference in test versus OOD validation performance shows that the choice of OOD hospital can significantly affect performance.

From these WSIs, we extracted patches in a standard manner, similar to Veeling et al. (2018). The WSIs were scanned at a resolution of 0.23μm–0.25μm in the original dataset, and each WSI contains multiple resolution levels, with approximately 10,000×20,000 pixels at the highest resolution level (Bandi et al., 2018). We used the third-highest resolution level, corresponding to reducing the size of each dimension by a factor of 4. We then tiled each slide with overlapping 96×96 pixel patches with a step size of 32 pixels in each direction (such that none of the central 32×32 regions overlap), labeling them as the following:

- *Tumor* patches have at least one pixel of tumor tissue in the central 32×32 region. We used the pathologist-annotated tumor annotations provided with the WSIs.

- *Normal* patches have no tumor and have at least 20% normal tissue in the central 32×32 region. We used Otsu thresholding to distinguish normal tissue from background.

We discarded all patches that had no tumor and <20% normal tissue in the central 32×32 region.

To maintain an equal class balance, we then subsampled the extracted patches in the following way. First, for each WSI, we kept all tumor patches unless the WSI had fewer normal than tumor patches, which was the case for a single WSI; in that case, we randomly discarded tumor patches from that WSI until the numbers of tumor and normal patches were equal. Then, we randomly selected normal patches for inclusion such that for each hospital and split, there was an equal number of tumor and normal patches.

Modifications to the original dataset. The task in the original CAMELYON17 challenge was the patient-level classification task of determining the pathologic lymph node stage of the tumor present in all slides from a patient. In contrast, our task is a lesion-level classification task. Patient-level, slide-level, and lesion-level tasks are all common in histopathology applications. As mentioned above, the original dataset provided WSIs and tumor annotations, but not a standardized set of patches or data splits, which we provide here.

The Camelyon17-wilds patch-based dataset is similar to one of the datasets used in Tellez et al. (2019), which was also derived from the CAMELYON17 challenge; there, only one hospital is used as the training set, and the other hospitals are all part of the test set. Camelyon17-wilds is also similar to PCam (Veeling et al., 2018), which is a patch-based dataset based on an earlier CAMELYON16 challenge; the data there is derived from only two hospitals.

Baseline model details. Our baseline models are DenseNet-121 (Huang et al., 2017) models trained from scratch with empirical risk minimization (ERM), using a learning rate of 10^{-3} and an L_2-regularization strength of 10^{-2}. See Figure 16 for results on each of the 10 random seeds we ran per baseline method. These were selected by a grid search with the learning rates 10^{-4}, 10^{-3}, 10^{-2}, and L_2-regularization strengths 0, 10^{-3}, 10^{-2}, using one random seed per hyperparameter setting, and choosing based on OOD validation accuracy. Optimization was done through stochastic gradient descent with momentum (set to 0.9).

We used the same learning rate and regularization strength for the DeepCORAL and IRM baselines, and grid searched separately over their respective penalty weights using the default grid (Appendix A).

As with the other Wilds datasets, we selected hyperparameters using the OOD validation set. We found that this improved performance relative to using the ID validation set. For example, using the same hyperparameters as above but early stopping on ID validation accuracy instead of OOD validation accuracy reduced test accuracy from an average of 73.3% (one s.d., 9.9%) to an average of 64.9% (one s.d., 12.2%) across ten random seeds. Note that the standard deviations are for the accuracy of a single random seed; the standard deviation of the average accuracy across 10 random seeds is correspondingly lower. In general, ID validation accuracies were very similar and did not provide as much of a signal on OOD accuracy.

Additional data sources. The full, original CAMELYON17 dataset contains 1000 WSIs from the same 5 hospitals, although only 50 of them (which we use here) have tumor annotations. The other 950 WSIs may be used as unlabeled data. Beyond the CAMELYON17 dataset, the largest source of unlabeled WSI data is the Cancer Genome Atlas (Weinstein et al., 2013), which typically has patient-level annotations (e.g., patient demographics and clinical outcomes).

Additional discussion. Many specialized methods have been developed to handle stain variation in the context of digital histopathology. These typically fall into one of two categories: data augmentation methods that perturb the colors in the training images (e.g., Liu et al. (2017); Bug et al. (2017); Tellez et al. (2018)) or stain normalization methods that seek to standardize colors across training images (e.g., Macenko et al. (2009); BenTaieb and Hamarneh (2017)). These methods

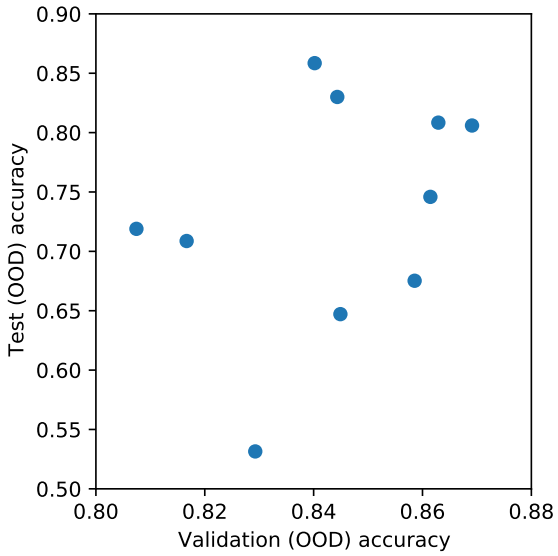

Figure 16: Test (OOD) accuracy versus validation (OOD) accuracy for different random seeds on Camelyon17-wilds, using the same hyperparameters. The test accuracy is far more variable than the validation accuracy.

are reasonably effective at mitigating stain variation, at least in some contexts (Tellez et al., 2019), though the general problem of learning digital histopathology models that can be effectively deployed across multiple hospitals/sites is still open.

Beyond stain variation, there are many other distribution shifts that might occur in histopathology applications. For example, as with all medical applications, patient demographics might differ from hospital to hospital, e.g., some hospitals might tend to see patients who are older or more sick, and patients from different backgrounds and countries vary in terms of cancer susceptibility (Henderson et al., 2012). Some cancer subtypes and tissues of origin are also more common than others, leading to potential subpopulation shift issues, e.g,. a rare cancer subtype in one context might be more common in another; or even if it remains rare, we would seek to leverage the greater quantity of data from other subtypes to improve model accuracy on the rare subtype (Weinstein et al., 2013).

B.5 OGB-MolPCBA

Additional dataset details. The OGB-MolPCBA dataset contains 437,929 molecules annotated with 128 kinds of labels, each representing a bioassay curated in the PubChem database (Kim et al., 2016b). More details are provided in the MolculeNet (Wu et al., 2018) and the Open Graph Benchmark (Hu et al., 2020b), from which the dataset is adopted.

In Wilds, we additionally provide the scaffold information that training algorithms can leverage in order to improve model's extrapolation capability. In Figure 17 (A), we plot the statistics of the scaffold groups in terms of their sizes. We see that the scaffold sizes are highly skewed. The training split contains the largest 44,930 scaffolds with 7.8 molecules per scaffold, the validation split contains the next largest 31,361 scaffolds with 1.4 molecules per scaffold, and the test split contains the smallest all-singleton 43,793 scaffolds. This implies that test molecules are maximally diverse in their structure, making it suitable to evaluate model's performance across diverse scaffold domains. How does the above data split affect the distribution of target labels? In Figures 17 (B) and (C), we quantify whether the scaffold split creates distribution shift in the prediction target labels. We see

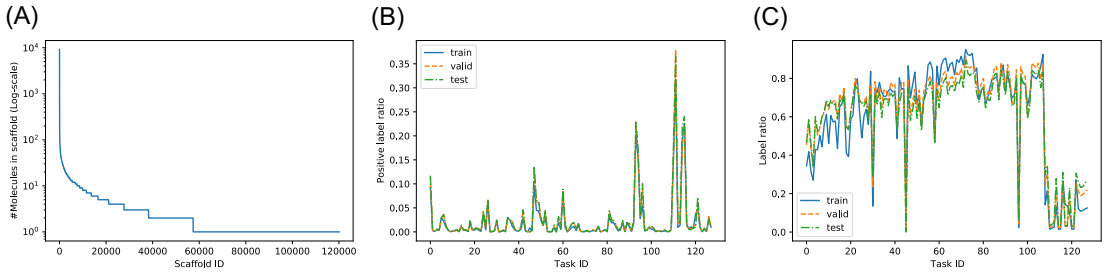

Figure 17: Analyses of scaffold groups in the OGB-MolPCBA dataset. (A) shows the distribution of the scaffold sizes, (B) and (C) show how the ratios of positive molecules and labeled molecules for the 128 tasks vary across the train/validation /test splits.

that the label statistics remain almost across train/validation/test splits, suggesting that the main distribution shift comes from the difference in the input molecular graph structure.

Additional baseline details. For model and optimization hyper-parameters, we follow the Open Graph Benchmark (Hu et al., 2020b). For model, we use the 5 GNN layers with the hidden dimensionality of 300. For regularization, we tune the dropout rate from $\{0, 0.5\}$. We use the Adam optimizer (Kingma and Ba, 2015) with the learning rate of 0.001.

B.6 Amazon-wilds

Additional setup. The input is a review text with a maximum token length of 512, and the label is the star rating out of 5 with $\mathcal{Y} = \{1, 2, 3, 4, 5\}$. For each example, the following additional metadata is available at both training and evaluation time: reviewer ID, product ID, product category, review time, and summary.

Additional dataset details. We consider a modified version of the Amazon reviews dataset (Ni et al., 2019). We consider disjoint reviewers between the training, OOD validation, and OOD test sets, and we also provide separate ID validation and test sets that include reviewers seen during training for additional reporting. These reviewers are selected uniformly at random from the reviewer pool, with the constraint that they have at least 150 reviews in the pre-processed dataset. Statistics for each split are described in Table 21. Notably, each reviewer has at least 75 reviews in the training set and exactly 75 reviews in the validation and test sets. For more details on pre-processing and subset selection, see the subsection on pre-processing below.

Split	# Reviews	# Reviewers	# Reviews per reviewer (mean / minimum / maximum)
Training	1,000,124	5,008	201 / 75 / 3,198
Validation (OOD)	100,050	1,334	75 / 75 / 75
Test (OOD)	100,050	1,334	75 / 75 / 75
Validation (ID)	100,050	1,334	75 / 75 / 75
Test (ID)	100,050	1,334	75 / 75 / 75

Table 21: Dataset details for Amazon-wilds.

Modifications to the original dataset. The original dataset does not consider a task nor a split. We consider a standard task of sentiment classification, but we depart from a standard split,

considering disjoint users between training and evaluation time as described above. In addition, we pre-process the data as detailed below.

Pre-processing and subset selection. We first eliminate reviews that are longer than 512 tokens, reviews without any text, and any duplicate reviews with identical star rating, reviewer ID, product ID, and time. We then obtain the 30-core subset of the reviews, which contains the maximal set of reviewers and products each of which have at least 30 reviews; this is a standard pre-processing procedure for the original dataset (Ni et al., 2019). To construct the dataset for reviewer shifts in particular, we further eliminate the following reviews: (i) reviews that contain HTML, (ii) reviews with identical text within a user in order to ensure sufficiently high effective sample size per reviewer, and (iii) reviews with identical text across users to eliminate generic reviews. Once we have the filtered set of reviews, we consider reviewers with at least 150 reviews and sample uniformly at random until the training set contains 1 million reviews and each evaluation set contains at least 100,000 reviews. As we construct the training set, we reserve a random sample of 75 reviews for each user for evaluation and put all other reviews to the training set. For the evaluation set, we put a random sample of 75 reviews for each user.

Additional baseline results. We report additional results on the baseline models in Table 22, reporting the performance on the train, ID validation, and ID test sets. We observe performance disparities across seen reviewers as well, and the performance gaps between seen and unseen reviewers are small across various metrics for our baseline models. We note that while group DRO successfully improves worst-group accuracy on seen users as advertised, it fails to significantly improve the wort-group accuracy for unseen users as well as the 10th percentile accuracies on seen and unseen users.

	Train		Validation (ID)	
Algorithm	Average accuracy	10th percentile accuracy	Average accuracy	10th percentile accuracy
ERM	60.4 (0.1)	76.4 (0.0)	58.7 (0.0)	75.5 (0.0)
DeepCORAL (reviewer)	59.4 (1.3)	76.7 (0.8)	57.3 (0.0)	75.0 (0.2)
IRM (reviewer)	62.1 (0.2)	78.2 (0.1)	58.2 (0.8)	74.9 (0.1)

Table 22: Additional results of baseline models on Amazon-wilds.

In addition, we include the detailed results of the in-distribution baseline models; these models are finetuned on reviews written by a particular user and are evaluated on the same reviewer (i.e., not evaluated on the in-distribution validation or test sets). We report the results in Table 23.

Reviewer ID	Train	Validation	Test
AV6QDP8Q0ONK4	88.8 (15.2)	55.6 (2.0)	44.4 (5.6)
A37BRR2L8PX3R2	90.7 (1.9)	75.6 (0.8)	63.1 (4.7)
A1UH21GLZTYYR5	99.9 (0.1)	82.7 (5.8)	68.0 (4.0)
ASVY5XSYJ1XOE	88.7 (2.4)	78.2 (3.1)	74.7 (2.3)
A1NE43T0OM6NNX	86.1 (20.3)	45.3 (5.8)	47.6 (5.4)
A9Q28YTLYREO7	98.6 (2.3)	74.7 (1.3)	66.7 (3.5)
A1CNQTCRQ35IMM	87.1 (11.1)	52.4 (6.0)	46.2 (6.2)
A20EEWWSFMZ1PN	95.9 (3.4)	72.9 (0.8)	56.0 (5.3)
A3JVZY05VLMYEM	93.7 (5.3)	81.3 (1.3)	79.6 (0.8)
A219Y76LD1VP4N	95.6 (4.2)	72.4 (3.8)	67.6 (1.5)

Table 23: Additional results of in-distribution baseline models on Amazon-wilds.

Demographic	Number of non-toxic comments	Number of toxic comments
Male	12092	2203
Female	14179	2270
LGBTQ	3210	1216
Christian	12101	1260
Muslim	5355	1627
Other religions	2980	520
Black	3335	1537
White	5723	2246

Table 24: Group sizes in the test data for CivilComments-wilds. The training and validation data follow similar proportions.

Baseline model details. For all baseline experiments, we fine-tune BERT-base-uncased models, using the implementation from Wolf et al. (2019), and with the following hyperparameter settings: batch size 8; learning rate 2×10^{-6}; L_2-regularization strength 0.01; 3 epochs; and a maximum number of tokens of 512. We select the above hyperparameters based on a grid search, considering learning rates $1 \times 10^{-6}, 2 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}$, and epochs 1-3. We select the hyperparameters that yield best average accuracy on the OOD validation set.

B.7 CivilComments-wilds

Additional dataset details. The CivilComments-wilds dataset comprises comments from a large set of articles from the Civil Comments platform, annotated for toxicity and demographic identities (Borkan et al., 2019b). We partitioned the articles into disjoint training, validation, and test splits, and then formed the corresponding datasets by taking all comments on the articles in those splits. In total, the training set comprised 269,038 comments (60% of the data); the validation set comprised 45,180 comments (10%); and the test set comprised 133,782 (30%).

Modifications to the original dataset. The original dataset[11] also had a training and test split with disjoint articles. These splits are related to ours in the following way. Let the number of articles in the original test split be m. To form our validation split, we took m articles (sampled uniformly at random) from the original training split, and to form our test split, we took $2m$ articles (also sampled uniformly at random) from the original training split and added it to the existing test split. We added a fixed validation set to allow other researchers to be able to compare methods more consistently, and we tripled the size of the test set to allow for more accurate worst-group accuracy measurement.

Similarly, we combined some of the demographic identities in the original dataset to obtain larger groups (for which we could more accurately estimate accuracy). Specifically, we created an aggregate *LGBTQ* identity that combines the original *homosexual_gay_or_lesbian*, *bisexual*, *other_sexual_orientation*, *transgender*, and *other_gender* identities (e.g., it is 1 if any of those identities are 1), and an aggregate *other_religions* identity that combines the original *jewish*, *hindu*, *buddhist*, *atheist*, and *other_religion* identities. We also omitted the *psychiatric_or_mental_illness* identity, which was evaluated in the original Kaggle competition, because of a lack of sufficient data for accurate estimation; but we note that baseline group accuracies for that identity seemed higher than for the other groups, so it is unlikely to factor into worst-group accuracy. In our new split, each identity we evaluate on (*male*, *female*, *LGBTQ*, *Christian*, *Muslim*, *other_religions*, *Black*, and *White*) has at least 500 positive and 500 negative examples. In Table 24 we show the sizes of each subpopulation in the test set; the training and validation sets follow similar proportions.

11. www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/

For convenience, we also add an *identity_ any* identity; this combines all of the identities in the original dataset, including *psychiatric_ or_ mental_ illness* and related identities.

The evaluation metric used in the original competition was a complex weighted combination of various metrics, including subgroup AUCs for each demographic identity, and a new pinned AUC metric introduced by the original authors (Borkan et al., 2019b); conceptually, these metrics also measure the degree to which model accuracy is uniform across the different identities. After discussion with the original authors, we replace the composite metric with worst-group accuracy for simplicity. As Borkan et al. (2019b) note, measuring subgroup AUCs can be misleading in this context, because it assumes that the classifier can set separate thresholds for different subgroups. See also Borkan et al. (2019a) for a discussion on the limitations of the pinned AUC metric.

Additional baseline details. Our baseline models are all fine-tuned BERT-base-uncased models, using the implementation from Wolf et al. (2019), and with the following hyperparameter settings: batch size 16; learning rate 10^{-5} using the AdamW optimizer for 5 epochs; the default L_2-regularization strength 10^{-2} from Devlin et al. (2019); and a maximum number of tokens of 300 (99.95% of the data had fewer than or equal to 300 tokens). We chose the learning rate through a brief grid search over learning rates 10^{-6}, 2×10^{-6}, 10^{-5}, 2×10^{-5}, picking the learning rate with the best worst-group validation accuracy when using a standard model fine-tuned through ERM. We used the same hyperparameter settings for the reweighted and group DRO baselines. All experiments used early stopping on worst-group validation accuracy.

The reweighted and group DRO baselines involve partitioning the training data into disjoint domains, as in Equations (6) and (7). We specifically study the following choices of domains, corresponding to different rows in Table 14:

1. *Label*: 2 subsets, 1 for each class.
2. *Label* $\times$ *Black*: 4 subsets, 1 for each combination of class and *Black*.

Additional baseline results. In Table 15 of the main text, we showed that the group DRO (label) model performed poorly on the group of non-toxic comments that mentioned the Black identity. In Table 25, we show detailed results for the group DRO (label $\times$ Black) model where we did early stopping based on accuracy on comments that mentioned the Black identity (as opposed to worst-group accuracy over all identities, which we do for all other baselines). The results show that this indeed improves accuracy on both toxic and non-toxic Black comments; for comparison, the group DRO (label) model had an accuracy of 69.2% on non-toxic Black comments, whereas this model has an accuracy of 75.4% on the same group. Unfortunately, this improvement comes at the expense of some other groups; in particular, the accuracy on non-toxic LGBTQ comments drops from 74.6% to 60.1%. Note that the results in Table 25 are different from the results reported for the group DRO (label $\times$ Black) model in Table 14, because the latter is based on early stopping with the worst-group accuracy over all identities.

We also trained a group DRO model using $2^9 = 512$ domains, 1 for each combination of class and the 8 identities. This model performed similarly to the other group DRO models.

We note that the relatively small size of some of these subpopulations makes it infeasible to estimate how well a model could do on each subpopulation (corresponding to demographic identity) if it were trained on just that subpopulation. For example, Black comments comprise only $<$4% of the training data, and training just on those Black comments is insufficient to achieve high in-distribution accuracy.

Additional data sources. All of the data, including the data with identity annotations that we use and the data with just label annotations, are also annotated for additional toxicity subtype attributes, specifically *severe_ toxicity*, *obscene*, *threat*, *insult*, *identity_ attack*, and *sexual_ explicit*. These annotations can be used to train models that are more aware of the different ways that a comment can be toxic; in particular, using the *identity_ attack* attribute to learn which comments are

Demographic	Accuracy on non-toxic comments	Accuracy on toxic comments
Male	82.5 (2.2)	83.4 (2.6)
Female	84.6 (2.3)	82.1 (3.2)
LGBTQ	**60.1** (3.5)	86.9 (3.2)
Christian	88.5 (0.8)	81.1 (1.9)
Muslim	72.5 (2.5)	82.4 (2.0)
Other religions	81.0 (1.4)	81.5 (1.6)
Black	75.4 (1.0)	76.5 (1.4)
White	64.4 (1.6)	84.7 (0.6)

Table 25: CivilComments-wilds results for the Group DRO (label × Black) model with early stopping on accuracy on comments that mention the Black identity. Compared to the Group DRO (label) model in Table 14, accuracy on Black comments is higher but accuracy on LGBTQ comments is lower. We show standard deviation across random seeds in parentheses.

toxic because of the use of identities might help the model learn how to avoid spurious associations between toxicity and identity. These additional annotations are included in the metadata provided through the Wilds package.

The original CivilComments dataset (Borkan et al., 2019b) also contains ≈1.5M training examples that have toxicity (label) annotations but not identity (group) annotations. For simplicity, we have omitted these from the current version of CivilComments-wilds. These additional data points can be downloaded from the original data source and could be used, for example, by first inferring which group each additional point belongs to, and then running group DRO or a similar algorithm that uses group annotations at training time.

Additional discussion. Measuring worst-group accuracy treats false positives and false negatives equally; in deployment systems, one might want to weight these differently, e.g., using cost-sensitive learning or by simply raising or lowering the classification threshold. One could also binarize the labels and identities differently: in this benchmark, we simply use majority voting from the annotators.

In practice, models might do poorly on intersections of groups (Kearns et al., 2018), e.g., on comments that mention multiple identities. Given the size of the dataset and comparative rarity of some identities and of toxic comments in general, accuracies on these intersections are difficult to estimate from this dataset. A potential avenue of future work is to develop methods for evaluating models on such subgroups, e.g., by generating data in particular groups through templates (Park et al., 2018; Ribeiro et al., 2020).

Another important consideration for toxicity detection in practice is shifts over time, as online discourse changes quickly, and what is seen as toxic today might not have even appeared in the dataset from a few months ago. We do not study this distribution shift in this work. One limitation of the CivilComments-wilds dataset is that it is fixed to a relatively short period in time, with most comments being written in the span of a year; this makes it harder to use as a dataset for studying temporal shifts.

Appendix C. Other datasets

C.1 BDD100K: Object recognition in autonomous driving across locations

As discussed in Section 6.6, autonomous driving, and robotics in general, is an important application that requires effective and robust tools for handling distribution shift. Here, we discuss our findings on a modified version of the BDD100K dataset that evaluates on shifts based on time of day and location.

Our results below suggest that more challenging tasks, such as object detection and segmentation, may be more suited to evaluations of distribution shifts in an autonomous driving context.

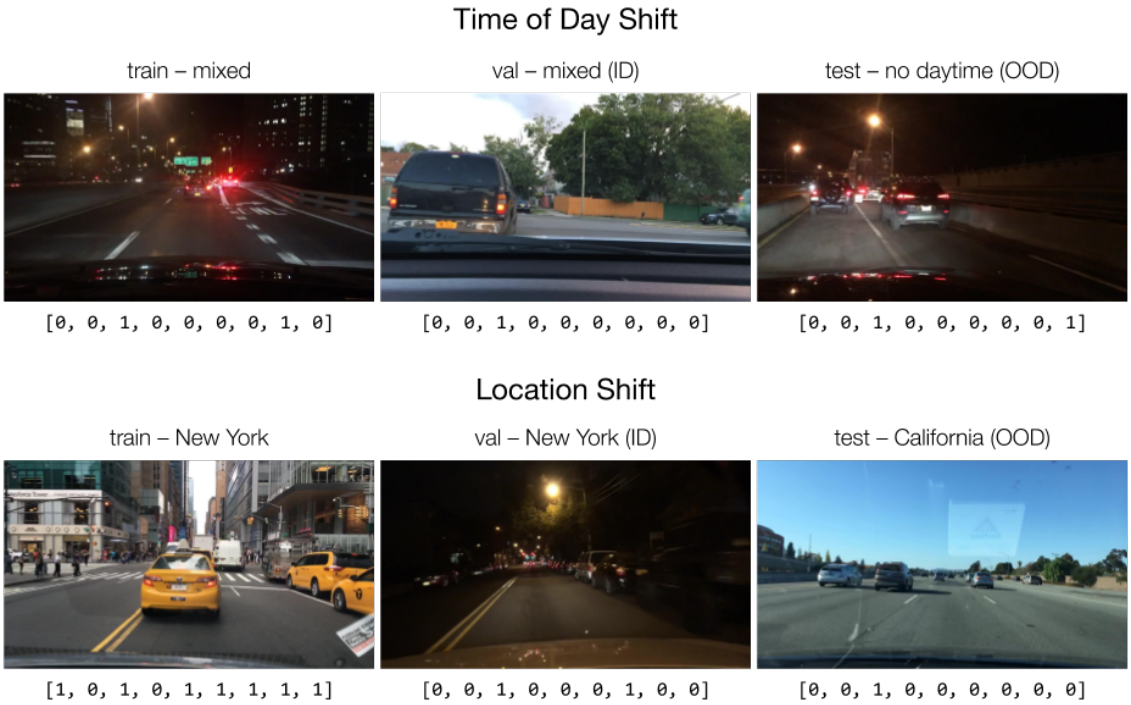

Figure 18: For BDD100K, we study two different types of shift, based on time of day and location. We visualize randomly chosen images and their corresponding labels from the training, validation, and test splits for both shifts. The labels are 9-dimensional binary vectors indicating the presence (1) or absence (0) of, in order: bicycles, buses, cars, motorcycles, pedestrians, riders, traffic lights, traffic signs, and trucks.

	Time of day shift		Location shift	
Algorithm	Validation (ID)	Test (OOD)	Validation (ID)	Test (OOD)
ERM	87.1 (0.3)	89.7 (0.2)	87.9 (0.0)	86.9 (0.0)

Table 26: Average multi-task classification accuracy of ERM trained models on BDD100K. All results are reported across 3 random seeds, with standard deviation in parentheses. We observe no substantial drops in the presence of test time distribution shifts.

C.1.1 Setup

Task. In line with the other datasets in Wilds, we evaluate using a classification task. Specifically, the task is to predict whether or not 9 different categories appear in the image x: bicycles, buses, cars, motorcycles, pedestrians, riders, traffic lights, traffic signs, and trucks. This is a multi-task binary classification problem, and the label y is thus a 9-dimensional binary vector.

Data. The BDD100K dataset is a large and diverse driving dataset crowd-sourced from tens of thousands of drivers, covering four different geographic regions and many different times of day,

weather conditions, and scenes (Yu et al., 2020). The original dataset contains 80,000 images in the combined training and validation sets and is richly annotated for a number of different tasks such as detection, segmentation, and imitation learning. We use bounding box labels to construct our task labels, and as discussed later, we use location and image tags to construct the shifts we evaluate.

Evaluation. In evaluating the trained models, we consider average accuracy across the binary classification tasks, averaged over each of the validation and test sets separately. We next discuss how we create and evaluate two different types of shift based on time of day and location differences.

C.1.2 Time of day shift

Distribution shift and evaluation. We evaluate two different types of shift, depicted in Figure 18. For time of day shift (Figure 18 top row), we use the original BDD100K training set, which has roughly equal proportions of daytime and non daytime images (Yu et al., 2020). However, we construct a test set using the original BDD100K validation set that only includes non-daytime images. We then split roughly the same number of images randomly from the training set to form an in-distribution validation set. There are 64,993, 4,860, and 4,742 images in the training, validation, and test splits, respectively.

ERM results. Table 26 summarizes our findings. For time of day shift, we actually observe slightly *higher* test performance, on only non daytime images, than validation performance on mixed daytime and non daytime images. We constrast this with findings from Dai and Van Gool (2018); Yu et al. (2020), who showed worse test performance for segmentation and detection tasks, respectively, on non daytime images. We believe this disparity can be attributed to the difference in tasks – for example, it is likely more difficult to draw an accurate bounding box for a car at night than to simply recognize tail lights and detect the presence of a car.

C.1.3 Location shift

Distribution shift. For location shift (Figure 18 bottom row), we combine all of the data from the original BDD100K training and validation sets. We construct training and validation sets from all of the images captured in New York, and we use all images from California for the test set. The validation set again is in distribution with respect to the training set and has roughly the same number of images as the test set. There are 53,277, 9,834, and 9,477 images in the training, validation, and test splits, respectively.

ERM results. In the case of location shift, we see from Table 26 that there is a small drop in performance, possibly because this shift is more drastic as the locations are disjoint between training and test time. However, the performance drop is relatively small and the test time accuracy is still comparable to validation accuracy. In general, we believe that these results lend support to the conclusion that, for autonomous driving and robotics applications, other more challenging tasks are better suited for evaluating performance. Generally speaking, incorporating a wide array of different applications will likely require a simultaneous effort to incorporate different tasks as well.