Okapi: Generalising Better by Making Statistical Matches Match

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose *Okapi*, a simple, efficient, and general method for robust semi-supervised learning based on online statistical matching. Our method uses a nearest-neighbours-based matching procedure to generate cross-domain views for a consistency loss, while eliminating statistical outliers. In order to perform the online matching in a runtime- and memory-efficient way, we draw upon the self-supervised literature and combine a memory bank with a slow-moving momentum encoder. The consistency loss is applied within the feature space, rather than on the predictive distribution, making the method agnostic to both the *modality* and the *task* in question. We experiment on the WILDS 2.0 datasets [39], which significantly expands the range of modalities, applications, and shifts available for studying and benchmarking real-world unsupervised adaptation. Contrary to [39], we show that it is in fact possible to leverage additional unlabelled data to improve upon empirical risk minimisation (ERM) results with the right method. Our method outperforms the baseline methods in terms of out-of-distribution (OOD) generalisation on both the iWildCam (a multi-class classification task) and PovertyMap (a regression task) datasets. Furthermore, from a qualitative perspective, we show that the matches produced from the learned encoder are related in semantically meaningful ways.

1 Introduction

Machine learning models have been deployed for safety-critical applications such as disease diagnosis [47] and self-driving cars [51], and in socially important contexts such as the allocation of healthcare, education, and credit (e.g. [27, 18]). Many machine learning algorithms, however, rely on supervision from a large amount of labelled data, and are typically trained to exploit complex relationships and distant correlations present in the training dataset. This strategy is proven to be effective in the setting when we have training (source) and test (target) data that are identically and independently distributed.

In reality, machine learning models are often deployed on target data whose distribution is different from the source distribution they were trained on. For example, in the task of classifying animal species in a camera trap image, we wish to learn a model that can generalise to new camera trap locations despite variations in illumination, background, and label frequencies, given training examples from a limited set of camera trap locations. Exploiting correlations that only hold in these limited locations but not in the new locations can hurt out-of-distribution (OOD) generalisation. While we only have a small subset of camera traps that have their images labelled, we have a large amount of unlabelled data from the other camera traps that capture diverse operating conditions. In general, unlabelled data is much more readily available than labelled data and can often be obtained from distributions beyond the source distribution. Taking advantage of these unlabelled data during train-

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

ing is a key element to build robust models that have good OOD performance without sacrificing in-distribution (ID) performance.

The inspiration of our work comes from the surprising empirical conclusion of [39] on their WILDS 2.0 datasets. WILDS 2.0 extended the WILDS benchmark datasets [30] – built for evaluating the performance of models under natural distribution shifts – with unlabelled data. Sagawa et al. [39] performed experiments with a variety of state-of-the-art methods that leverage unlabelled data, including domain-invariant, self-training, and self-supervised methods to assay the usefulness of the unlabelled data in improving OOD performance. In the majority of cases, however, these methods failed to outperform the combination of effective data-augmentation and standard empirical risk minimisation (ERM). In a related empirical study of the usefulness of unlabelled data for OOD generalization, [28] showed that one can achieve state-of-the-art results on an array of a range of benchmarks by simply scaling up the architecture and performing a pre-training and fine-tuning of the model only on the labelled source domain. Those recent empirical studies might suggest that there might not be any value in the abundant yet unlabelled task-specific data.

In this work, we argue that the size of unlabelled data is both advantage and disadvantage. Large unlabelled data offer diverse operating conditions but they might dilute the information contained in the labelled data due to the sheer volume of the unlabelled data. We need to find a way to identify and link labelled and unlabelled data that correspond to similar statistical units but that need not be the same real-world entity. We turn to a statistical matching framework [38, 37] which is a model-based approach for providing joint information on variables and indicators collected through multiple sources. Statistical matching has been widely utilised to assess the effect of interventions in numerous fields, such as education, medical and community policies (e.g. [14, 8]). In statistical matching, intervened units are paired with control units, and unpaired units are then dropped from the matched sample. In the running example of animal-species classification, intervened units refer to limited camera trap locations with labelled example, while control units refer to many more camera trap locations with unlabelled data. Pairing is beneficial to capture diverse operating conditions, but the ability to drop unpaired units is crucial to ensure that unlabelled data do not overwhelm the supervisory signal from labelled data.

By developing an online method for statistically matching samples from different domains (camera-trap locations) and using this to define a consistency loss, we arrive at our proposed semi-supervised method, *Okapi*. This consistency loss is based on the simple idea of pulling together similar samples from different domains within the latent space of the encoder, and using this to bootstrap said encoder such that the distributions become progressively more aligned over the course of training. Since matching samples using the full dataset at each step of training is computationally infeasible, we instead approximate it using a combination of momentum-encoding and a memory-bank that has been well-proven in self-supervised learning [25, 31]. Compared with other consistency-based methods such as FixMatch [41], Okapi has the advantage of being agnostic to both the task and the modality, in addition to being distributionally robust. Contrary, to Sagawa et al. [39], we show that the supplementary unlabelled data and domain information can be leveraged by Okapi to improve upon standard ERM on datasets from the WILDS 2.0 benchmark.

2 Preliminaries

2.1 Problem setting

In the standard supervised setting, one is given a dataset, $\mathcal{D}_l \triangleq \{x_i, y_i\}_{i=0}^{N_l}$, and trains a model, parameterised by θ, to well-approximate the empirical distribution as $p_\theta(y|x)$. Labelled data is limited by the cost of annotation yet one often has access to a far larger corpus of unlabelled data, $\mathcal{D}_u \triangleq \{x_i\}_{i=0}^{N_u}$, which can be used to supplement $\mathcal{D}_l$. Semi-supervised learning is motivated by the idea that this additional data can often be used to improve the ID and/or OOD performance of $p_\theta(y|x)$ We can view unsupervised domain adaptation (UDA) as a special case of semi-supervised learning, where there is assumed to be some distribution shift (adverse to a naïvely-trained predictor) between $\mathcal{D}_l$ and $\mathcal{D}_u$. Here, $\mathcal{D}_u$ comes from the domain on which $p_\theta(y|x)$ is to be evaluated, such that we have $\mathcal{D}_u \triangleq \mathcal{D}_{\text{OOD}}$, where $\mathcal{D}_{\text{OOD}}$ denotes the target domain, that is OOD w.r.t. D_l. In the most general sense, a *domain*, or *environment* [3, 15] describes some partitioning of the data according to its source or some secondary characteristic, such as time of day, weather, location, lighting, or the model of the device used to collect said data; one would hope that a predictor trained under one set

of conditions (e.g. day) would perform with minimal degradation under another set of conditions (e.g. night) when those conditions are irrelevant to the task at hand.

Assuming the data follows the conditional generative distribution $x \sim p(x|s)$, where s is the domain label, one would ideally use $\mathcal{D}_{\text{OOD}}$ to learn invariance to the marginal distribution, $p(s)$, and thereby achieve the equivalence $p_\theta(y|x) = p_\theta(y|x, s)$. In practice, one typically does not have access to $\mathcal{D}_{\text{OOD}}$ but does have access to training data sourced from a mixture of domains which can be leveraged to learn a more general invariance that extends to those domains outside the training distribution [3]. Such a learning paradigm is referred to as domain generalisation (DG). While some DG works consider the more extreme case of s being unobserved [15], we follow the more conventional setup [3, 32, 39] in which the domain associated with each sample (labelled and unlabelled) is indicated by the discrete label s. We denote the set of possible domains for the in-distribution labelled and unlabelled data, as $\mathcal{S}_l$ and $\mathcal{S}_u$, respectively, and their union as $\mathcal{S} \triangleq S_l \cup S_u$. Following the setup established in [39], $\mathcal{D}_u$ is assumed to be unlabelled only w.r.t the targets and not w.r.t the domain labels and thus that both $\mathcal{D}_l$ and $\mathcal{D}_u$ can be augmented with the latter to give the re-definitions $\mathcal{D}_l \triangleq \{x_i, y_i, s_i\}_{i=0}^{N_l}$ and $\mathcal{D}_u \triangleq \{x_i, s_i\}_{i=0}^{N_u}$.

2.2 Statistical matching

Matching is a sampling strategy which aims to balance the distribution of the observed covariates in the *treatment* and *control* groups. In general terms, observed covariates x are measured characteristics of the samples; in our work we will refer to the image encodings as covariates. The treatment and control groups are two partitions of the data. Specifically, the treatment group is the set of samples having a specific value of a variable of interest $\mathcal{S} = s$ and the control group is its complement.

In this work we utilise Nearest Neighbour (NN) matching, a distance matching method that pairs sample i of the treatment group with the closest sample j belonging to the control group. A distance measure, d_{ij}, is used to define how close two samples i and j are; *propensity score* and *Euclidean* are examples of different distances commonly used. Propensity score distance (PSD) is the difference between propensity scores, e_i and e_j, of samples i and j, i.e. $d_{ij} = |e_i - e_j|$. In causal inference, the propensity score (e_i) is the probability of sample i belonging to the treatment group, given its covariates x_i [36]; in practice, however, this conditional probability is rarely known. Although any binary classifier could be used for the prediction, logistic regression is typically utilised for the task [42]. The Euclidean-distance approach, on the other hand, computes the distance between the covariates, x_i and x_j, of a given pair of samples. While PSD is the most common distance measure used for matching, Euclidean distance should be preferred for generating pairs with close values on all of the covariates [29]. Nevertheless, propensity scores can still be a relevant component of NN-based matching for defining calipers to reduce the risk of poor matches.

In this work we make use of two types of *caliper*, fixed and standard deviation. The fixed caliper [16] defines a region of common support between the estimated propensity score distribution of the two groups; all samples outside such overlap region are discarded before matching. This selection rule helps removing samples with extreme propensity scores. The standard deviation-based caliper (std-caliper) [37] determines the maximum discrepancy between two samples before they are disallowed from pairing with each other. The discrepancy is usually expressed in terms of estimated PSD and the std-caliper is then defined as $|e_i - e_j| \leq \sigma \cdot \alpha$. Here, the variable σ denotes the mean of the propensity score standard deviation across the treatment and control groups and α is a parameter controlling the percentage of bias reduction of the covariates. In the following section, we describe how one can leverage this matching framework to define a consistency loss encouraging inter-domain robustness.

3 Method

Here, we introduce *Okapi*, a simple, efficient, and general (in the sense that it is applicable to any task *or* modality) method for robust semi-supervised learning based on online statistical matching. Our method belongs to the broad family of consistency-based methods, characterised by methods such as FixMatch [41], where the idea is to enforce similarity between a model's outputs for two views of an unlabelled image. These semi-supervised approaches based on minimising the discrepancy between two views of a given sample are closely related with self-supervised methods based on instance discrimination [12] and self-distillation [23, 11, 5]. Many of the methods within this family, however, are limited in applicability due to their dependence on modality-specific transformations and only

recently has research into self-supervision sought to redress this problem with modality-agnostic alternatives such as MixUp [46], masking [5], and nearest-neighbours [45, 19, 31]. Approaches such as FixMatch, AlphaMatch [21] and CSSL [33] that enforce consistency between the *predictive* distributions suffer further from not being directly generalisable to tasks other than classification. Okapi addresses both of these aforementioned issues through 1) the use of a statistical matching procedure – that we call $\mathrm{CaliperNN}$ and detail in Sec. 3.2 – to generate multiple views for a given sample; 2) enforcing consistency between encodings rather than between predictive distributions.

We show that models trained to maximise the similarity between the encoding of a given sample and those of its CaliperNN-generated match are significantly more robust to real-world distribution shifts than the baseline methods, while also having the advantage of being both computationally efficient and agnostic to the modality and task in question. Qualitatively speaking, we see that the matches obtained from the final model are related in semantically-meaningful ways. Furthermore, since the only constraint is that samples be from different domains, the method is applicable whether information about the domain is coarse or fine-grained.

In the following subsections, we begin by giving a general formulation of our proposed semi-supervised loss employing a generic cross-domain k-NN algorithm. We then explain how we can replace this algorithm with CaliperNN in order to mitigate the risk of poorly-matched samples, and how the loss may be computed in an online fashion to give our complete algorithm.

3.1 Enforcing consistency between cross-domain pairs

We consider our predictor as being composed of an encoder (or *backbone*) network, $f_\theta : \mathcal{X} \to \mathbb{R}^d$, generating intermediary outputs (features) $z \triangleq f_\theta(x)$, and a prediction head, g_ϕ, such that the prediction for sample x is given by $\hat{y} \triangleq g_\phi \circ f_\theta(x)$. We similarly consider the aggregate loss $\mathcal{L}$ as having a two-part decomposition given by

$$\mathcal{L} \triangleq \mathcal{L}_{\text{sup}} + \lambda\, \mathcal{L}_{\text{unsup}}, \tag{1}$$

where $\mathcal{L}_{\text{sup}}$ is the supervised component measuring the the discrepancy (as computed, for example, by the MSE loss) between $\hat{y}$ and the ground-truth label y, $\mathcal{L}_{\text{unsup}}$ is the unsupervised component based on some kind of pretext task, such as cross-view consistency, and λ is a positive pre-factor determining the trade-off between the two components. For our method, we do not assume any particular form for $\mathcal{L}_{\text{sup}}$ and focus solely on $\mathcal{L}_{\text{unsup}}$.

Given a pair of datasets $\mathcal{D}_l$ and $\mathcal{D}_u$, sourced from the labelled domain $\mathcal{S}_l$, and unlabelled domain $\mathcal{S}_u$ respectively, our goal is to train a predictor that is robust (invariant) to changes in domain, including those unseen during training. To do this, we propose to regularise $z \triangleq f_\theta(x)$ to be smooth (consistent) within local, cross-domain neighbourhoods. At a high-level, for any given *query* sample x_q sourced from domain s_q, we compute $\mathcal{L}_{\text{unsup}}$ as the mean distance between its encoding $z_q \triangleq f_\theta(x_q)$ and that of its k-nearest neighbours, $V_k(z_q)$ with the constraint that $\{s_q\} \cap \mathbf{s}_n = \emptyset$, where $\mathbf{s}_n$ is the set of domain-labels associated with $V_k(z_q)$. The general form of this loss for a given sample can then be written as

$$V_k(z_q) \triangleq \mathrm{NN}(z_q, \{f_\theta(x) \mid (x, s) \in \mathcal{D}, s \neq s_q\}, k), \tag{2}$$

$$\mathcal{L}_{\text{unsup}} \triangleq \frac{1}{k} \sum_{z_n \in V_k(z_q)} d(z_q, z_n) \tag{3}$$

where $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is some distance function. Here, we follow [23] and define d to be the squared Euclidean distance between normalised encodings for our experiments. Allowing the NN algorithm to select pairs in an unconstrained manner, given the pool of queries and keys, however, can lead to poorly-matched pairs that are detrimental to the optimisation process. To address this, we replace the standard NN algorithm with a propensity-score-based variant, inspired by the statistical matching framework [36].

3.2 Cross-domain matching

For the matching component of our algorithm, we propose to use a variant of k-NN which, in addition to incorporating the above cross-domain constraint, filters the queries and keys that represent

probable outliers, according to their learned propensity scores. The initial stage of filtering employs a fixed caliper, where samples with propensity scores surpassing a fixed confidence threshold are removed; this is followed by a second stage of filtering where any two samples (from different domains) can only be matched if the Euclidean distance between their respective propensity scores is below a pre-defined threshold (std-caliper). We illustrate these steps in Appendix B.

The propensity score, e, for a given sample x is estimated as $p(s|z)$ using a linear classifier f_θ, $h_\psi : \mathbb{R}^d \to \triangle^{|\mathcal{S}|}$ where $\triangle^{|\mathcal{S}|}$ is the probability simplex over possible domain labels, $\mathcal{S}$. h_ψ^d is trained via maximum (weighted) likelihood to predict the domain label of a given sample for all samples within the aggregate dataset $\mathcal{D} \triangleq \mathcal{D}_l \cup \mathcal{D}_u$, or (typically) a subset of it, encoded by f_θ. Since we apply both calipers to the learned propensity score, the shape of this distribution can have a significant effect on the outcome of matching. Accordingly, we apply temperature-scaling, with scalar $\tau \in \mathbb{R}_\star^+$, to sharpen or flatten the learned propensity-score distribution such that we have $e \triangleq \mathrm{softmax}(\tau^{-1}\, h_\psi(z))$. We denote the set of associated parameters ($\{\, t_f, t_\sigma, \tau\}$, as the threshold for the fixed-caliper, the threshold for the std-caliper, and the temperature, respectively) as ξ and discuss in Appendix A how one can determine suitable values for these in practice.

For convenience we define the set of all encodings, given by f_θ, as $\mathbf{z} \triangleq \{f_\theta(x) \,|\, x \in \mathcal{D}\}$ and, using this, the set of all associated propensity scores as $\mathbf{e} \triangleq \{h_\psi(x) \,|\, z \in \mathbf{z}\}$. In the offline case, the matches for $\mathcal{D}$ are then computed as

$$\mathrm{MatchedSamples} \triangleq \{(z, \mathrm{CaliperNN}_\xi(z, \mathbf{z}, \mathbf{e}, s, k)) \,|\, z \in \mathbf{z}, s \in \mathbf{s}\}, \tag{4}$$

with CaliperNN returning the set of k-nearest neighbours according to d, subject to the aforementioned cross-domain and caliper-based constraints. We allow for the fact that there may be no valid matches for some samples due to these constraints; in such cases we have $\emptyset$ as the second element of their tuples, indicating that $\mathcal{L}_{\mathrm{unsup}}$ should be set to 0.

3.3 Scaling up with Online Learning

Re-encoding the dataset following each update of the feature-extractor, in order to recompute $\mathrm{MatchedSamples}$, is prohibitively expensive, with cost scaling linearly with $N \triangleq N_l + N_u$. Moreover, CaliperNN requires explicit computation of the pairwise distance matrices, which can be prohibitive memory-wise for large values of N. We address these problems using a fixed-size memory bank, $\mathcal{M}_z^{N_\mathcal{M}}$ storing only the last $N_\mathcal{M}$ (where $N_\mathcal{M} \ll N$) encodings from a slow-moving momentum encoder [25, 23], $f_{\theta'}$, which we refer to as the *target* encoder, in line with [23], and accordingly refer to f_θ as the *online* encoder. Unlike [23], however, we make use of neither a projector nor a predictor head (in the case of the target encoder) in order to compute the inputs to the consistency loss and simply use the output of the backbone as is – this is possible in our setting due to $\mathcal{L}_{\mathrm{sup}}$ preventing representational collapse. More specifically, the target encoder's parameters, θ', are computed as a moving average of the online encoder's, θ, with decay rate $\zeta \in (0, 1)$, per the recurrence relation

$$\theta'_t = \zeta\theta'_{t-1} + (1 - \zeta)\theta_t, \tag{5}$$

As the associated domain labels are also needed both for matching and to compute the loss for the propensity scorer, we also store the labels associated with $\mathcal{M}_z^{N_\mathcal{M}}$ in a companion memory bank $\mathcal{M}_s^{N_\mathcal{M}}$. We initialise $\mathcal{M}_z^{N_\mathcal{M}}$ and $\mathcal{M}_s^{N_\mathcal{M}}$ to $\emptyset$, rather than randomly, resulting in fewer than $N_\mathcal{M}$ samples being used during the initial stages of training when the memory banks are yet to be populated.

Each iteration of training, we sample a batch of size B from $\mathcal{D}$ consisting of inputs $\mathbf{x}$ and domain labels $\mathbf{s}$. During the matching phase, the inputs are passed through the *target* encoder to obtain $\mathbf{z}'_q \triangleq \{f_{\theta'}(x) | x \in \mathbf{x}\}$, serving as the queries for CaliperNN. We also experiment with a simpler variant where the *online* encoder is instead used for this query-generation step, such that we instead have $\mathbf{z}'_q \triangleq \{f_\theta(x) | x \in \mathbf{x}\}$), and find this can work equally well if ζ is sufficiently high. The keys are then formed by combining the current queries with the past queries contained in the memory bank: $\mathbf{z}_k \triangleq \mathbf{z}'_q \cup \mathcal{M}_z^{N_\mathcal{M}}$. The domain labels associated with $\mathbf{z}_k$ are likewise formed by concatenating the domain labels in the current batch with those stored in $\mathcal{M}_s^{N_\mathcal{M}}$: $\mathbf{s}_k \triangleq \mathbf{s}_q \cup \mathcal{M}_s^{N_\mathcal{M}}$. Once the matches for the current samples have been computed, the oldest B samples in $M_z^{N_\mathcal{M}}$ and $M_s^{N_\mathcal{M}}$ are overwritten with $\mathbf{z}_k$ and $\mathbf{s}_k$, respectively. The consistency loss is then enforced between each

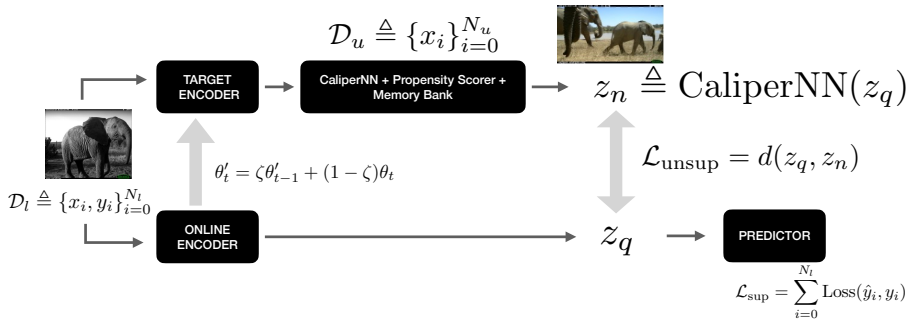

Figure 1: Overview of Okapi's online-learning pipeline. For simplicity, we illustrate here the special case in which the domains of the labelled, $\mathcal{D}_l$, and unlabelled data $\mathcal{D}_u$, are disjoint and matching is thus performed bidirectionally between them; for the same reason we limit k to 1 so that the output of matching is a single vector rather than a set of vectors. Inspired by recent advances in self-supervised learning, we maintain a copy (the target encoder) of the online encoder, f_θ, whose parameters, θ', are an exponential moving average (EMA) of θ. This EMA update is performed at the beginning of each training set at a rate governed by the decay coefficient, ζ. For a given sample from $\mathcal{D}_l$, we first compute its embedding using the target encoder to produce the query vector, z'_q, and by the online encoder to produce z_q, which will serve as the 'anchor' in the consistency loss. This query vector is then used by CaliperNN to compute its cross-domain nearest neighbour, z_n, where the keys are taken to be the current and past (stored in the Memory Bank) $N_{\mathcal{M}}$ encodings of the unlabelled data. The consistency loss is the distance between z_q and z_n, defined by function some distance function d. Finally, the supervised loss, $\mathcal{L}_{\text{sup}}$, is computed using the output of the predictor acting on z_n and the ground-truth given by y.

query $\mathbf{z}_q \triangleq \{f_\theta(x) | x \in \mathbf{x}\}$, according to the differentiable *online* encoder, and each of its matches, $V_k(z'_q) \triangleq \text{CaliperNN}_\xi(z_q, \mathbf{z}_k, h_\psi(\mathbf{z}_k), \mathbf{s}_k)$ providing that $V_k(z'_q) \neq \emptyset$ (that is, under the condition that the estimated propensity score for z'_q does not violate the caliper(s) and there is at least k valid matches whose estimated propensity scores also do not), with the loss simply 0 otherwise. Since $f_{\theta'}$ is frozen, $\mathbf{z}_k$ carries an implicit stop-gradient and gradients are computed only w.r.t. θ.

Similarly, rather than solving for the optimal parameters, $\psi^\star$ for the propensity scorer given the current values of $\mathbf{z}_k$, which is infeasible for the large values of $N_{\mathcal{M}}$ needed to well-approximate the full dataset, we resort to a biased estimate of $\psi^\star$. Namely, we train h_ψ in an online fashion to minimise the per-batch loss

$$\mathcal{L}_{ps} = \sum_{z \in \mathbf{z_k}, s \in \mathbf{s}_k} \mathcal{H}(h_\psi(z), s), \tag{6}$$

where $\mathcal{H}$ is the standard cross-entropy loss between the predictive distribution and the (degenerate) ground-truth distribution given by the one-hot encoded domain labels. In the special case in which the $\mathcal{D}_l$ and $\mathcal{D}_u$ are known to have disjoint support over S (that is, $\mathcal{S}_l \cap \mathcal{S}_u = \emptyset$), we can substitute their domain labels with 1 and 0, respectively (such that we have $\mathcal{D}_l \triangleq \{x_i, y_i, 1\}_{i=0}^{N_l}$ and $\mathcal{D}_u \triangleq \{x_i, 0\}_{i=0}^{N_u}$), thus reducing the propensity scorer and CaliperNN to their binary forms. We note that it is entirely possible for this condition is known to be satisfied without having s explicitly labelled for each sample meaning that no ground-truth domain-labels are required at any stage of our pipeline. One may, for example know that two sets of satellite imagery cover two different parts of the world (e.g. Africa and Asia) yet not know the exact coordinates underlying their respective coverage.

4 Related Work

Domain Generalisation The goal of domain generalisation (DG) is to produce models that are robust to a wide range of distribution shifts (including those outside the training distribution), given a training set consisting of samples sourced from multiple domains. Despite the various techniques (many well theoretically-motivated) designed to improve the generalisation of deep neural networks current methods continue to fall short in the face of natural distribution shifts [24, 30]. Indeed, ERM has repeatedly shown to be a strong baseline – frequently outperforming dedicated methods that

leverage domain information or additional unlabelled data – for DG [24, 39], despite the theoretical problems associated with using it when the training and test sets are misaligned. Thus far, only pre-training on larger, more diverse datasets (with harder examples), has consistently proven to improve OOD generalisation and allowing the pre-trained models to fit the ID data too closely is often detrimental in this regard [43, 2, 48, 28]. In this work we question whether this really is the case and establish that one can effectively leverage in-distribution data (labelled and unlabelled) to create more robust models given that the data is sufficiently diverse.

Self-Supervised Learning In self-supervised learning (SelfSL), models are trained to solve pretext tasks constructed from the input data. This learning paradigm has led to significant breakthroughs in unsupervised learning in recent years, with performance now approaching (or even surpassing, along some axes such as adversarial robustness) that of supervised methods for many tasks while requiring significantly less labelled data. Due to its generality, SelfSL has seen use across the complete spectrum of applications and modalities and underlies many of the foundation models [9] that have emerged in NLP [17, 10, 13], Computer Vision [22], and at their intersection [52, 1]. Common pretext tasks include those based on the masked-language-modelling approach – originally popularised by BERT [17] and recently generalised to other modalities [6, 5] – [12, 25], contrastive captioning [35, 52], and instance discrimination and self-distillation [11, 23] which rely on transformations of the data to generate multi-view inputs. Approaches belonging to the latter two categories were originally limited by the fact that the transforms had to be tailored for a particular modality and for some modalities, such as tabular data, there is no obvious way to define them. A number of recent works have sought to obviate this problem through the use of MixUp [46], masking [5, 26], and k-nearest neighbours [19, 31, 45], the latter of which is directly relevant to our work. Okapi bears closest resemblance to [31] in combining momentum-encoding with nearest-neighbours lookup to generate the views for a BYOL-style [23] consistency loss. However, a key distinction lies in the use of an augmented form of nearest-neighbours, CaliperNN, which both constrains pairs of samples to being from *different* domains and filters out any queries or keys representing probable outliers according to a learned *propensity score*.

Semi-Supervised Learning Semi-supervised learning (SemiSL) encompasses a broad class of algorithms that combine unsupervised learning with supervised learning in order to improve the performance of the latter, especially when labelled data is limited. Many SemiSL methods are based on the self-training paradigm which can trace its roots back decades to the early work in pattern recognition by [40] and continues to be relevant in the modern era due to its generality, both within SemiSL itself and in related fields such as domain adaptation [20], and fledgling field of SelfSSL [11] discussed above. Self-training applies to any framework predicated on using a model's own predictions to produce pseudo-labels for the unlabelled data which can either be used as targets for self-distillation [49] or enforcing consistency between predictions that themselves have been perturbed [49, 4] or that have been generated from perturbed/multi-view inputs [41]. FixMatch [41] is one example of a consistency-based method which has proven effective for semi-supervised classification, despite its simplicity, and various works [21, 33] have since built on the its framework prescribing the use of weakly- and strongly-augmented inputs to generate the targets and predictions, respectively. Like these methods, Okapi also makes use of a cross-view consistency loss, however, the alternative views for a given sample are generated not through data-augmentation but through statistical matching [36], with the aim being to achieve invariance to the domain rather than a particular series of perturbations. Another example of particular relevance to our work is [44], which uses a copy of the model with exponentially-averaged weights to generate the targets for the unlabelled data. Okapi also uses such a model to produce the targets for its consistency loss, but is more akin to momentum-encoding [25] in the respect that the loss is imposed on the latent space.

5 Experiments

5.1 Datasets

We evaluate Okapi using two datasets, iWildCam and PovertyMap, taken from the WILDS 2.0 benchmark [39]. These datasets were chosen specifically due to the poor performance reported by [39] for semi-supervised and domain adaptation methods across the board, in relation to the ERM baselines. For PovertyMap in particular, ERM was found to vastly outperform any competing methods utilising the unlabelled data and/or domain labels.

iWildCam-WILDS is an extension of the iWildCam 2020 Competition Dataset [7]. The task is multi-class species classification of animals in camera trap images. The dataset contains 1022K images of animals annotated with the domain, s, that identifies the camera trap that captured it. The target label, y, is one of 182 different animal species and it is provided solely for the 203K labelled data. The labelled training set contains 130K images taken by 243 camera traps. The out-of-distribution (OOD) validation and target sets include images from 32 and 48 different camera traps which are disjoint from the 243 training domains. Additionally, 819K unlabelled images from 3215 new domains are available. As images from different cameras trap differ in characteristics such as illumination, background and relative animal frequency, models trained on the source domains might fail to generalise to images taken from new locations.

PovertyMap-WILDS is a variation of the dataset presented in [50]. The task is to predict the wealth index, y, from multispectral satellite images of 23 African countries. The country the image was taken in as well as whether it was taken in a rural or urban area represent the domain s. The dataset contains 5 cross-validation (CV) folds of roughly equal size, each one dividing the 23 countries differently across the source, OOD validation and OOD target splits. In each fold, the labelled training set contains 11K images from 14 different countries. The OOD validation and target sets include images from 5 different countries, each which are all different from the countries of the source data. The dataset also includes 261K unlabelled images from the same 23 countries.

5.2 Results

In Table 1 we summarise the numerical results. Here, we use the metrics reported in [39] for each dataset: macro-F1 for the iWildCam classification task and the Pearson correlation for the regression task defined by PovertyMap, specifically, the worst-case-performance between the correlation of the urban (U) and rural (R) areas. For completeness, we also include the mean squared error (MSE) measure for the latter. Following WILDS, we show the mean and standard deviation (in brackets) values across multiple runs for both ID and OOD test sets. While for iWildCam we aggregate the metric over three random seed runs, for PovertyMap we aggregate it over the 5 CV-folds contained in the dataset.

We compare our method (Okapi) against two baselines, ERM and FixMatch [41], both according to our re-implementation and according to the original implementation given in [39]. We note that since FixMatch, in its original form, is only applicable to classification problems due to its use of confidence-based thresholding, for the PovertyMap dataset, FixMatch represents a simplified variant (following [39]) without such thresholding, that is trained to simply minimise the MSE between *all* regressed values for the weakly- and strongly-augmented images. As described in Appendix A, the main difference between the baselines run included in [39] and our re-runs is in the backbone architecture, with us opting for a ConvNeXt [34] architecture over a ResNet one. For both datasets, and for both baselines we observe significant improvements stemming from using ConvNeXt over ResNet. Moreover, utilising ConvNeXt seems to be crucial in enabling FixMatch to surpass the ERM baseline in the classification task with 32.2 (ERM) vs 31.0 (FixMatch) and 33.3 (ERM) vs. 35.2 (FixMatch), with ResNet and ConvNeXt architecture respectively.

Our proposed method, Okapi, convincingly outperforms the baselines, w.r.t the OOD metric of interest, on both datasets. We see an improvement of $+0.9$ macro F1, i.e. 36.1 vs 35.2 of Okapi and FixMatch (the best baseline for iWildCam) respectively. For the PovertyMap regression task, Okapi achieves 0.55 and 0.33 on the OOD test set in terms of Pearson correlation and MSE, respectively, in contrast to the 0.50 and 0.33 of ERM. We also see that FixMatch fails to generalise well to this task, yielding by far the worst results amongst the evaluated methods.

In order to evaluate the importance of the caliper-based filtering to the performance of Okapi, we perform an ablation experiment on PovertyMap dataset (Okapi (no calipers)) with said filtering disabled (and all else constant), such that instead of CaliperNN we have standard k-NN, albeit with the cross-group constraint still in place (per Eq. 2). We see that performance degrades according to both metrics of interest, and, crucially, that the standard deviation of the runs is significantly higher, in line with our expectation that sifting out poor matches should stabilise optimisation. We provide additional ablation experiments in Appendix C, exploring the relative importance of the two (fixed and std-) calipers, the optimal number of neighbours to use for computing $\mathcal{L}_{\text{unsup}}$, and the feasibility of using the online encoder to generate the queries for CaliperNN.

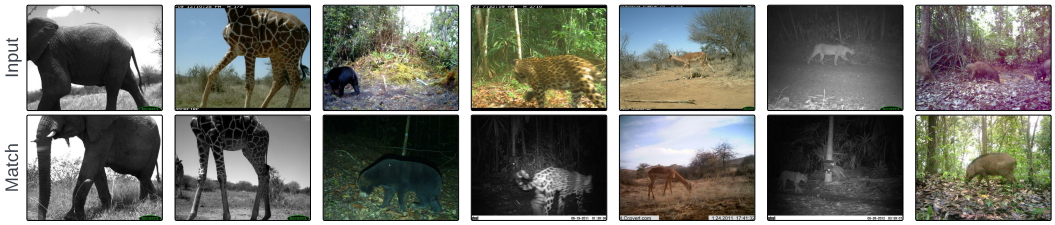

Figure 2: Examples of input (labelled) images and their 1-NN matched (unlabelled) images retrieved using CaliperNN on iWildCam dataset. Here, we match images from the labelled-train set to images from the unlabelled-extra set, taking advantage the fact that their set of domains are disjoint.

Table 1: A comparison between Okapi and different baselines on two benchmark datasets. We include both the results of our re-run of the baselines and the results shown in [39]. Both ID and OOD performances are reported. For iWildCam we replicated each run three times, using different random seeds. For PovertyMap we run an experiment on each of the 5 CV folds of the dataset. We report the average and standard deviation value across replicates of the metric of interest.

Method	**iWildCam** macro F1 ↑		**PovertyMap** worst U/R corr. ↑		worst U/R MSE ↓	
	ID	OOD	ID	OOD	ID	OOD
ERM [39]	47.0 (1.4)	32.2 (1.2)	0.66 (0.04)	0.49 (0.06)	-	-
FixMatch [39]	46.3 (0.5)	31.0 (1.3)	0.54 (0.1)	0.30 (0.11)	-	-
ERM	48.6 (1.1)	33.3 (0.3)	0.68 (0.03)	0.5 (0.1)	0.24 (0.04)	0.33 (0.07)
FixMatch	51.1 (1.0)	35.2 (0.7)	0.5 (0.13)	0.34 (0.12)	0.59 (0.42)	0.88 (0.61)
Okapi (ours)	50.6 (0.7)	36.1 (0.9)	0.72 (0.02)	0.55 (0.1)	0.22 (0.02)	0.33 (0.1)
Okapi (no calipers)	-	-	0.72 (0.02)	0.54 (0.12)	0.22 (0.02)	0.36 (0.14)

Finally, in Fig. 2 we show a random sample of matched pairs produced using our CaliperNN from the learned encoder for the iWildCam dataset. Here, we see that semantic information (encoding the species of animal) is preserved across pairs, while nuisance factors such as illumination, background and contrast vary. Further examples from PovertyMap are shown in Appendix B.

6 Conclusion

In this work, we introduced, Okapi, a semi-supervised method for training distributionally-robust models that is intuitive, effective, and is applicable to any modality or task. Okapi is based on the simple idea of supplementing the supervised loss with a cross-domain consistency loss that encourages the outputs of an encoder network to be similar for neighbouring (within the latent space of the encoder itself) samples belonging to different domains, which is made efficient using an online-learning framework. Rather than simply using k-NN with a cross-domain constraint, however, we propose an augmented form based on statistical matching (CaliperNN) that combines propensity scores with calipers to winnow out low-quality matches; we find this to be important for both the end-performance and consistency of Okapi. Our work serves as a response to [39], in that we find that it is in fact possible to effectively incorporate unlabelled data and domain information into a training algorithm in order to improve upon ERM with respect to an OOD test set, assuming an appropriate choice of architecture. Namely, on two datasets from the WILDS 2.0 benchmark, defining two different tasks (classification and regression), we show that Okapi outperforms both the ERM and FixMatch baselines according to the relevant OOD metrics.

Buoyed by these promising results, we intend to apply Okapi to other tasks (e.g. object detection and image segmentation) and other modalities (e.g. text and audio) to further establish its generality. Furthermore, one limitation of the current incarnation of the method is that the thresholds for the calipers are fixed over the course of training whereas it may be beneficial to set these adaptively with the view to optimise such measures of inter-domain balance as *Variance Ratio* and *Standard Mean Differences* that are commonly used to evaluate the the goodness of statistical matching procedures.

References

[1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[2] A. Andreassen, Y. Bahri, B. Neyshabur, and R. Roelofs. The evolution of out-of-distribution robustness throughout fine-tuning. *arXiv preprint arXiv:2106.15831*, 2021.

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

[4] P. Bachman, O. Alsharif, and D. Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014.

[5] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.

[6] H. Bao, L. Dong, and F. Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

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

[8] A. Biglan, D. Ary, and A. C. Wagenaar. The value of interrupted time-series experiments for community intervention research. *Prevention Science*, 1(1):31–49, 2000.

[9] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[13] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[14] P. Christian, L. E. Murray-Kolb, S. K. Khatry, J. Katz, B. A. Schaefer, P. M. Cole, S. C. LeClerq, and J. M. Tielsch. Prenatal micronutrient supplementation and intellectual and motor function in early school-aged children in nepal. *Jama*, 304(24):2716–2723, 2010.

[15] E. Creager, J.-H. Jacobsen, and R. Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[16] R. K. Crump, V. J. Hotz, G. W. Imbens, and O. A. Mitnik. Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96(1):187–199, 2009.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] J. A. Dunnmon, D. Yi, C. P. Langlotz, C. Ré, D. L. Rubin, and M. P. Lungren. Assessment of convolutional neural networks for automated classification of chest radiographs. *Radiology*, 290(2):537–544, 2019.

[19] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021.

[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[21] C. Gong, D. Wang, and Q. Liu. Alphamatch: Improving consistency for semi-supervised learning with alpha-divergence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13683–13692, 2021.

[22] P. Goyal, Q. Duval, I. Seessel, M. Caron, M. Singh, I. Misra, L. Sagun, A. Joulin, and P. Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022.

[23] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[24] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[26] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

[27] M. Hurley and J. Adebayo. Credit scoring in the era of big data. *Yale Journal of Law and Technology*, 18:148–216, 2017.

[28] D. Kim, K. Wang, S. Sclaroff, and K. Saenko. A broad study of pre-training for domain generalization and adaptation. *arXiv preprint arXiv:2203.11819*, 2022.

[29] G. King and R. Nielsen. Why propensity scores should not be used for matching. *Political Analysis*, 27(4):435–454, 2019.

[30] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[31] S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash. Mean shift for self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10326–10335, 2021.

[32] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[33] J. Lienen and E. Hüllermeier. Credal self-supervised learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14370–14382. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/7866c91c59f8bffc92a79a7cd09f9af9-Paper.pdf`.

[34] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[36] P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[37] P. R. Rosenbaum and D. B. Rubin. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39(1): 33–38, 1985.

[38] D. B. Rubin. Matching to remove bias in observational studies. *Biometrics*, pages 159–183, 1973.

[39] S. Sagawa, P. W. Koh, T. Lee, I. Gao, S. M. Xie, K. Shen, A. Kumar, W. Hu, M. Yasunaga, H. Marklund, S. Beery, E. David, I. Stavness, W. Guo, J. Leskovec, K. Saenko, T. Hashimoto, S. Levine, C. Finn, and P. Liang. Extending the WILDS benchmark for unsupervised adaptation. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=z7p2V6KROOV`.

[40] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[41] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[42] E. A. Stuart. Matching methods for causal inference: A review and a look forward. *Stat Sci*, 25(1):1–21, 2010. ISSN 0883-4237.

[43] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

[44] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[45] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, and L. V. Gool. Revisiting contrastive methods for unsupervised learning of visual representations. *Advances in Neural Information Processing Systems*, 34, 2021.

[46] V. Verma, T. Luong, K. Kawaguchi, H. Pham, and Q. Le. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, pages 10530–10541. PMLR, 2021.

[47] D. S. Watson, J. Krutzinna, I. N. Bruce, C. E. Griffiths, I. B. McInnes, M. R. Barnes, and L. Floridi. Clinical applications of machine learning algorithms: beyond the black box. *Bmj*, 364, 2019.

[48] O. Wiles, S. Gowal, F. Stimberg, S.-A. Rebuffi, I. Ktena, K. D. Dvijotham, and A. T. Cemgil. A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=Dl4LetuLdyK`.

[49] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[50] 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(1):1–11, 2020.

[51] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[52] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Yes. We show through our experiments detailed in Sec. 5 that our proposed method is able to outperform our baseline methods, and the original implementations of them given in [39], according on two recent OOD benchmark datasets.
   (b) Did you describe the limitations of your work? Yes. We describe the foremost limitation of our work in Sec. 6. Namely, we point out that using fixed thresholds for the caliper the entire duration of training is likely suboptimal and these hyperparameters should instead be set adaptively.
   (c) Did you discuss any potential negative societal impacts of your work? No, we do not anticipate there being any negative societal impacts specific to our work. On the contrary, we propose a method for training more distributionally-robust models with the view to develop safer, less biased, machine-learning systems.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes, we have read the review guidelines and have made sure that our paper complies with them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? N/A
   (b) Did you include complete proofs of all theoretical results? N/A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?
   Yes; the code, accompanied by the requisite instructions and configuration files needed to run all of the experiments detailed in Sec 5, will be provided with the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes; all implementation details, including those related to optimisation and hyperparameter-selection, are given in Appendix A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes, we ran all methods with multiple random seeds and multiple folds of the data in the case of the iWildCam and PovertyMap datasets, respectively; we report the standard deviation over the runs for each method-dataset combination in brackets in 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? No.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? Yes, we use datasets from the WILDS 2.0 benchmark developed by Sagawa et al. [39] for evaluating our models and cite the associated paper in both Sec. 1 and Sec. 5.
   (b) Did you mention the license of the assets? No.
   (c) Did you include any new assets either in the supplemental material or as a URL? No.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? N/A
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? N/A
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? N/A
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? N/A
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? N/A