Okapi: Generalising Better by Making Statistical Matches Match

Myles Bartlett[1*] **Sara Romiti**[1] **Viktoriia Sharmanska**[1,2] **Novi Quadrianto**[1,3,4]

[1]Predictive Analytics Lab, University of Sussex [2]Imperial College London
[3]BCAM Severo Ochoa Strategic Lab on Trustworthy Machine Learning
[4]Monash University, Indonesia

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 [63], which significantly expands the range of modalities, applications, and shifts available for studying and benchmarking real-world unsupervised adaptation. Contrary to [63], 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 the iWildCam (a multi-class classification task) and PovertyMap (a regression task) image datasets as well as the CivilComments (a binary classification task) text dataset. Furthermore, from a qualitative perspective, we show the matches obtained from the learned encoder are strongly semantically related. Code for our paper is publicly available at `https://github.com/wearepal/okapi/`.

1 Introduction

Machine learning models have been deployed for safety-critical applications such as disease diagnosis [73] and self-driving cars [77], and in socially important contexts such as the allocation of healthcare, education, and credit (e.g. [23, 35]). 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 has proven to be effective in the setting when we have training (source) and test (target) data that are i.i.d.

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, one aims 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

*Corresponding author: `m.bartlett@sussex.ac.uk`.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

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

Our work is a direct response to the empirical conclusions of [63] for the WILDS 2.0 dataset, which extends the WILDS benchmark datasets of [41] through the addition of unlabelled data. In Sagawa et al. [63] a variety of state-of-the-art methods leveraging unlabelled data, including domain-invariant, self-training, and self-supervised methods were evaluated for their ability to improve OOD generalisation. In the all but a few cases, however, these methods failed to outperform the combination of effective data-augmentation and standard empirical risk minimisation (ERM), and among those select cases none persisted across datasets.

We show that it is possible to make effective use of large volumes of unlabelled data as supplement to a smaller set of labelled data, from a limited set of domains, to achieve strong generalisation to data from domains outside the training distribution. We turn to a statistical matching (SM) framework [57, 59, 60], a model-based approach for providing joint information on variables and indicators collected through multiple sources. SM has been widely utilised to assess the effect of interventions in numerous fields, such as education, medical and community policies (e.g. [10, 17]). In SM, intervened units are paired with control units and those units without a sufficiently-good match according to a given statistical criteria are excluded when estimating the treatment effect. In the running example of animal-species classification, intervened units may correspond to the limited set of camera trap locations that are fully-annotated, while control units refer to the many more camera trap locations that are only partially annotated. Pairing is beneficial for capturing diverse operating conditions, yet the ability to drop unpaired units is crucial for mitigating the risk of statistically-poor matches corrupting the training signal.

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 predicated 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 [33, 42]. Compared with other consistency-based methods such as FixMatch [67], 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. [63], 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, $\mathcal{D}_l \triangleq \{x_i, y_i\}_{i=1}^{N_l}$, 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, $\mathcal{D}_u \triangleq \{x_i\}_{i=1}^{N_u}$, 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* [4, 19] 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 [4]. Such a learning paradigm is referred to as domain generalisation (DG). While some DG works consider the more extreme case of s being unobserved [19], we follow the more conventional setup [4, 43, 63] in which the domain(s) associated with each sample (labelled and unlabelled) is indicated by the discrete label (set of labels) s. We denote the set of possible domains for the in-distribution labelled and unlabeled 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 [63], $\mathcal{D}_u$ 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 $\mathcal{D}_l \triangleq \{x_i, y_i, s_i\}_{i=1}^{N_l}$ and $\mathcal{D}_u \triangleq \{x_i, s_i\}_{i=1}^{N_u}$.

2.2 Statistical matching

Statistical matching is a sampling strategy which aims to balance the distribution of the observed covariates in the *treated* and *control* groups. In general terms, observed covariates x are measured characteristics of the samples; in our work we refer to the encodings generated by a deep neural network as covariates instead of the original characteristics. The treated and control groups are two partitions of the data; specifically, the treated group is the set of samples having a specific value of a variable of interest (here, the domain indicator, s) and the control group is its complement.

In this work we utilise Nearest Neighbour (NN) matching, a distance-based matching method that pairs sample i of the treated group with the closest sample j belonging to the control group. A distance measure is used to define how close two samples, i and j, are, with *propensity score distance* (PSD) and *Euclidean distance* being two widely-used distances that we rely on – indirectly (as a means of filtering) and directly, respectively – in this work.

The propensity score distance is defined as the difference between propensity scores, e_i and e_j, of samples i and j, i.e. $\text{PSD}(e_i, e_j) \triangleq |e_i - e_j|$. In causal inference, the propensity score refers to the probability of sample i belonging to the treated group, given its covariates x_i [58]; in practice, this conditional probability is rarely known a priori and thus requires estimation, typically via logistic regression [68]. The Euclidean-distance approach, in contrast, computes the distance between the covariates, x_i and x_j, of a given pair of samples. Despite PSD being the more prevalent of the two distances, it is ill-suited to cases in where pairs are close in value w.r.t. all covariates and in such cases Euclidean distance should be preferred [40]. Nevertheless, propensity scores remain a relevant component of NN-based matching for defining *calipers* that can reduce the likelihood of false-positive matches.

In this work we make use of two types of caliper, *fixed* and *standard deviation*. The fixed caliper [20], t_f, defines a region of common support between the estimated propensity score distribution of the two groups; only those samples within the feasible region are admissible for matching ($\{i|e_i \in (1 - t_f, t_f)\}$). This selection rule helps by removing samples with extreme propensity scores. The standard deviation-based caliper (std-caliper), on the other hand, [59] determines the maximum discrepancy that might exist between two samples while still being admissible for pairing. The discrepancy is usually expressed in terms of estimated PSD: $|e_i - e_j| < \sigma \cdot t_\sigma$. Here, σ denotes the mean of the group-wise standard deviations of the propensity scores and t_σ 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 [67], 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 [15] and self-distillation [6, 14, 30]. 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 [72], masking [6], and nearest-neighbours [24, 42, 71]. Approaches such as FixMatch, AlphaMatch [28] and CSSL [46] 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 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 having the advantage of being both computationally efficient and agnostic to the modality and task in question. Qualitatively speaking, we see that matches produced with 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 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, along with their union $\mathcal{D} \triangleq \mathcal{D}_l \cup \mathcal{D}_u$ 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

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

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

where $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is some distance function. Here, we follow [30] 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 [58].

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

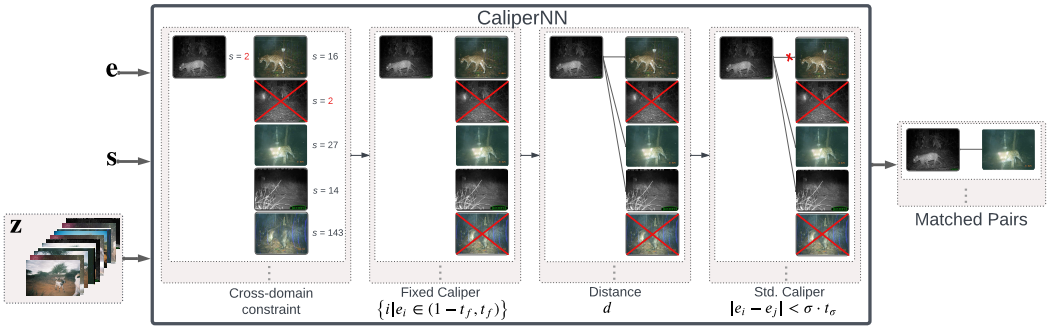


Figure 1: Illustration of our proposed statistical matching algorithm, CaliperNN. Given the anchor image encoding $\mathbf{z}$, the corresponding domain label s, and propensity score e, CaliperNN outputs the closest samples, according to distance d, subject to their being from different domains to the anchor.

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 wherein 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). See Fig 1 for a pictorial representation of these steps and Appendix G for reference pseudocode.

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 $\mathcal{D}$, 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 D how one can determine suitable values for these in practice.

For convenience we define the set of all encodings, given by f_θ, as $\mathbf{z} \triangleq \{f_\theta(x) \,|\, x \in \mathcal{D}\}$, the set of all associated propensity scores as $\mathbf{e} \triangleq \{h_\psi(x) \,|\, z \in \mathbf{z}\}$, and the set of associated domain labels as $\mathbf{s}$. 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 [30, 33], $f_{\theta'}$, which we refer to as the *target* encoder, in line with [30], and accordingly refer to f_θ as the *online* encoder. Unlike [30], 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 $\emptyset$, 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 $\mathbf{s}$. 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 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 are 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. These steps are illustrated pictorially in Fig 2 and as pseudocode in Appendix G.

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

$$\mathcal{L}_{\text{ps}} = \frac{1}{|\mathbf{z}_k|} \sum_{z \in \mathbf{z}_k, s \in \mathbf{s}_k} w_{\mathbf{s}_k}(s)\mathcal{H}(h_\psi(z), s), \tag{6}$$

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, and $w_{\mathbf{s}_k} : \mathcal{S} \to \mathbb{R}_*^+$ is a function assigning to each s an importance weight [66] based on the inverse of its frequency in $\mathbf{s}_k$ to counteract label imbalance. 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 $\mathcal{D}_l \triangleq \{x_i, y_i, 1\}_{i=1}^{N_l}$ and $\mathcal{D}_u \triangleq \{x_i, 0\}_{i=1}^{N_u}$), thus reducing the propensity scorer and CaliperNN to their binary forms. Knowing whether this condition is satisfied a priori (and thus whether the use of domain labels can be forgone completely from our pipeline) is not unrealistic: 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 [31, 41]. Indeed, ERM has repeatedly shown to be a strong baseline – frequently outperforming dedicated methods that leverage domain information or additional unlabelled data – for DG [31, 63], despite the theoretical problems associated with using it when the training and test sets are misaligned. Until now, only pre-training on larger, more diverse datasets (with harder examples), has consistently proven to improve OOD generalisation, yet allowing pre-trained models to fit the ID data too closely can undo any such benefit conferred by the pre-training [3, 39, 69, 74]. Similar to Okapi, MatchDG [51] draws upon causal matching to tackle DG. Despite the surface-level similarity, there are a number of significant differences, principally in the respects that we consider semi-supervised DG (whereas

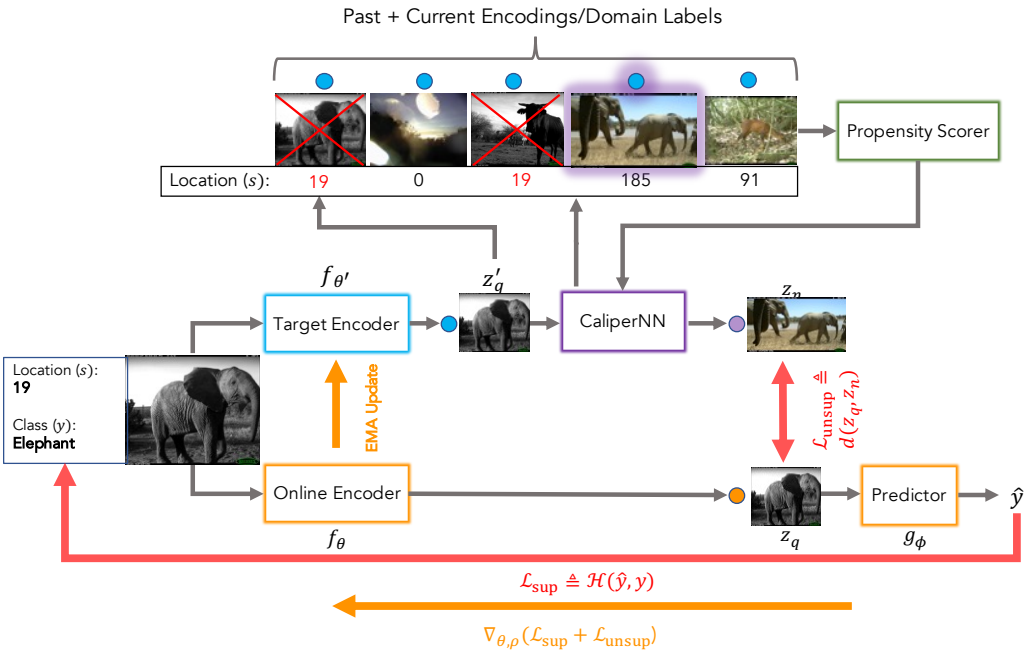


Figure 2: Overview of Okapi's online-learning pipeline based using the iWildCam dataset for the sake of illustration. For simplicity, we limit k to 1 so that the output of matching is a single vector rather than a set of vectors; for the same reason we illustrate the process for only a single sample taken from the labelled data set $\mathcal{D}_l$, annotated with both domain (s; in this case, *camera location*) and class (y) information. 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, 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 – alongside the output of the propensity scorer – 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 data. The cross-domain constraint, prohibiting matching of samples belonging to the same domain, is denoted through a red coloring of the location identifiers, the nearest sample obeying this constraint and the constraints of the calipers with purple highlighting. The consistency loss is the distance between z_q and z_n, defined by function some distance function d. Finally, the supervised loss, $\mathcal{L}_{\text{sup}}$ (here instantiated as the standard cross-entropy loss, $\mathcal{H}$), is computed using the output of the predictor acting on z_q and the ground-truth given by y.

MatchDG requires full-labeling w.r.t. y) and employ an augmented form of k-NN for bias-reduction in the absence of y.

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 [11] that have emerged in NLP [13, 16, 21], Computer Vision [29], and at their intersection [2, 78]. Common pretext tasks include those based on the masked-language-modelling approach – originally popularised by BERT [21] and recently generalised to other modalities [6, 7] – [15, 33], contrastive captioning [56, 78], and instance discrimination and self-distillation [14, 30] 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

Table 1: A comparison between Okapi and different baselines on two benchmark image datasets. We include both the results of our re-run of the baselines and those of [63]. Both ID and OOD performances are reported. For iWildCam we average over results from 3 different seeds, for PovertyMap we do so over the 5 pre-defined CV folds. Standard deviations are shown in parentheses.

Method	**iWildCam** macro F1 ↑		**PovertyMap** worst U/R corr. ↑		**PovertyMap** worst U/R MSE ↓	
	ID	OOD	ID	OOD	ID	OOD
ERM [63]	47.0 (1.4)	32.2 (1.2)	0.66 (0.04)	0.49 (0.06)	-	-
FixMatch [63]	46.3 (0.5)	31.0 (1.3)	0.54 (0.10)	0.30 (0.11)	-	-
ERM	48.6 (1.1)	33.3 (0.3)	0.72 (0.03)	0.53 (0.09)	0.23 (0.03)	0.35 (0.12)
FixMatch	51.1 (1.0)	35.2 (0.7)	0.50 (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 (0.10)	0.22 (0.02)	0.33 (0.10)
Okapi (no calipers)	-	-	0.72 (0.02)	0.54 (0.12)	0.22 (0.02)	0.36 (0.14)

works have sought to obviate this problem through the use of MixUp [72], masking [6, 34], and k-NN [24, 42, 71], the latter of which is directly relevant to our work. Okapi bears closest resemblance to [42] in combining momentum-encoding with nearest-neighbours lookup to generate the views for a BYOL-style [30] 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 deemed 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 [65] 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 [27], and fledgling field of SelfSL [14] 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 [75] or enforcing consistency between predictions that themselves have been perturbed [5, 75] or that have been generated from perturbed/multi-view inputs [67]. FixMatch [67] is one example of a consistency-based method which has proven effective for semi-supervised classification, despite its simplicity, and various works [28, 46] 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 [58], 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 [70], 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 [33] in the respect that the loss is imposed on the latent space.

5 Experiments

5.1 Datasets

We evaluate Okapi on three datasets taken from the WILDS 2.0 benchmark [63]. These span a variety of modalities and tasks, allowing us to showcase the generality of our proposed method (Okapi): **iWildCam** (images, multiclass classification), **PovertyMap** (multispectral images, regression), and **CivilComments** (text, binary classification). Details of each dataset can be found in Appendix A.

5.2 Image experiments

Results of our image-data experiments are summarised in Table 1. Due to spacial constraints, we defer the full set of results, including those for the 'offline' (w.r.t. the matching) version of Okapi to Appendix. C. For both datasets in question, we use the same metrics as [63]: macro-F1 for iWildCam and worst-group (with the group defined as urban (U) vs. rural (R)) Pearson correlation for Poverty

Map. For completeness, we include mean squared error (MSE) as a secondary metric for the latter dataset. Following [63], we compute the mean and standard deviation (shown in parentheses) over multiple runs for both ID and OOD test sets, with these runs conducted with 3 different random seeds and 5 pre-defined cross-validation folds for iWildCam and PovertyMap, respectively.

We compare Okapi against two baselines, ERM and FixMatch [67], both according to our re-implementation and according to the original implementation given in [63]. 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 [63]) 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 D, the main difference between the baselines run included in [63] and our re-runs is in the backbone architecture, with us opting for a ConvNeXt [47] architecture over a ResNet one. For both datasets, and for both baselines we observe significant improvements stemming the change of backbone. 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.

Okapi, convincingly outperforms the baselines, w.r.t the OOD metric of interest, on both datasets. We observe 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 regression task in PovertyMap, Okapi achieves 0.55 and 0.33 on the OOD test set in terms of Pearson correlation and MSE, respectively, in contrast to the 0.53 and 0.33 of ERM. At the same time, we note that FixMatch fails to generalise well to this task, yielding by far the worst results amongst the evaluated methods.

5.3 Text classification

Method	**Civil Comments** worst-group acc ↑
	OOD
ERM [63]	66.6 (1.6)
ERM (fully-labelled) [63]	69.4 (0.6)
ERM (reproduction)	68.5 (2.2)
Okapi (ours)	69.7 (2.0)

Table 2: Comparison between Okapi and the baselines methods on the Civil Comments dataset. We include both the original results of [63] as well as those of our reproduction of their ERM baseline. Performance is measured in terms of worst-group accuracy and averaged over seeds; standard deviations are shown in parentheses.

In Table 2 we summarise the numerical results for the CivilComments dataset. Remaining consistent with [63], we evaluate models according to the worst-group accuracy – the minimum of the conditional accuracies obtained by conditioning on each of the 8 dimensions of s – averaged over 5 replicates. Since there is no canonical ID test split available for this dataset, we report only the results only for the OOD split that is, rather than doing so for a custom split to avoid misrepresentation. We compare Okapi against both ERM variants featured in [63] – one trained on only the official labelled data and one trained with annotated unlabelled data (fully-labelled) – as well as our re-implementation of the ERM variant trained on only the labelled data with an identical hyperparameter configuration to the former. In contrast to the image datasets, we do not diverge in our choice of architecture, with all models trained with a pre-trained DistilBERT [64] backbone.

We observe marked improvement in the worst-group accuracy of this baseline compared with that reported therein. We attribute this partly to the high variance of the model-selection procedure (inherited from [63]) based on intermittently-computed validation performance (which does not consistently align with test performance) to determine the final model. This aside, we observe that Okapi outperforms the ERM baseline by a significant margin, to the point of parity with the fully-labelled baseline.

5.4 Ablations and qualitatitive analysis

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

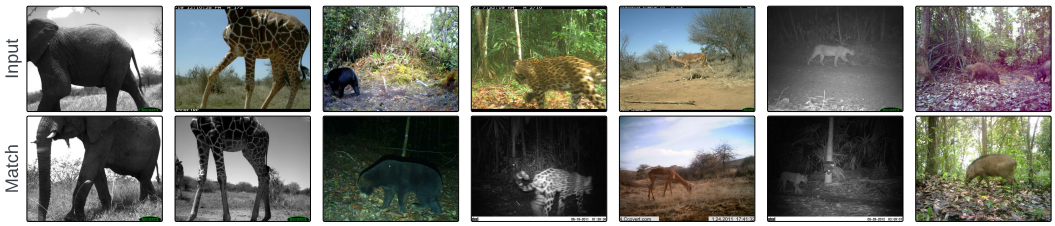


Figure 3: Examples of input (labelled) images and their 1-NN matched (unlabelled) images retrieved using CaliperNN on iWildCam dataset. Here, we match images from the labelled-train set to images from the unlabelled-extra set, taking advantage the fact that their domains are disjoint.

both metrics of interest, and, crucially, that the standard deviation of the runs is significantly higher, in line with our expectation that filtering out poor matches should stabilise optimisation. We provide additional ablation experiments in Appendix F, 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.

Finally, in Fig. 3 we show samples of matched pairs retrieved by CaliperNN from the encodings of 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 E. In Appendix H, we include matching results for the PACS (photo (P), art painting (A), cartoon (C), and sketch (S)) dataset [45] demonstrating how temperature scaling, in conjunction with the fixed caliper, can be used to control the filtering rate.

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 [63], 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 three datasets from the WILDS 2.0 benchmark, representing two different tasks (classification and regression) and modalities (image and text), 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. 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.

Acknowledgments and Disclosure of Funding

This research was supported by a European Research Council (ERC) Starting Grant for the project "Bayesian Models and Algorithms for Fairness and Transparency", funded under the European Union's Horizon 2020 Framework Programme (grant agreement no. 851538). NQ is also supported by the Basque Government through the BERC 2018-2021 program and by Spanish Ministry of Sciences, Innovation and Universities: BCAM Severo Ochoa accreditation SEV-2017-0718.

References

[1] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[2] 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.

[3] 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.

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

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

[6] 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.

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

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

[9] Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020.

[10] 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.

[11] 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.

[12] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] E. Creager, D. Madras, J.-H. Jacobsen, M. Weis, K. Swersky, T. Pitassi, and R. Zemel. Flexibly fair representation learning by disentanglement. In *International conference on machine learning*, pages 1436–1445. PMLR, 2019.

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

[20] 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.

[21] 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.

[22] M. Donini, L. Oneto, S. Ben-David, J. S. Shawe-Taylor, and M. Pontil. Empirical risk minimization under fairness constraints. *Advances in Neural Information Processing Systems*, 31, 2018.

[23] 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.

[24] 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.

[25] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[26] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

[32] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[33] 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.

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

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

[36] B. Y. Idrissi, M. Arjovsky, M. Pezeshki, and D. Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022.

[37] F. Kamiran and T. Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012.

[38] T. Kehrenberg, M. Bartlett, O. Thomas, and N. Quadrianto. Null-sampling for interpretable and fair representations. In *European Conference on Computer Vision*, pages 565–580. Springer, 2020.

[39] 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.

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

[41] 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.

[42] 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.

[43] 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.

[44] P. Lahoti, K. Gummadi, and G. Weikum. Operationalizing individual fairness with pairwise fair representations. *Proceedings of the VLDB Endowment*, 13(4):506–518, 2019.

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

[46] 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`.

[47] 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.

[48] I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=Skq89Scxx`.

[49] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[50] D. Madras, E. Creager, T. Pitassi, and R. Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR, 2018.

[51] D. Mahajan, S. Tople, and A. Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[52] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[53] L. Oneto, M. Donini, G. Luise, C. Ciliberto, A. Maurer, and M. Pontil. Exploiting mmd and sinkhorn divergences for fair and transferable representation learning. *Advances in Neural Information Processing Systems*, 33:15360–15370, 2020.

[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[55] N. Quadrianto, V. Sharmanska, and O. Thomas. Discovering fair representations in the data domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8227–8236, 2019.

[56] 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.

[57] S. Romiti, C. Inskip, V. Sharmanska, and N. Quadrianto. Realpatch: A statistical matching framework for model patching with real samples. *CoRR*, abs/2208.02192, 2022.

[58] 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.

[59] 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.

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

[61] D. B. Rubin. Using propensity scores to help design observational studies: application to the tobacco litigation. *Health Services and Outcomes Research Methodology*, 2(3):169–188, 2001.

[62] 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. *arXiv preprint arXiv:1911.08731*, 2019.

[63] 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`.

[64] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

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

[66] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[67] 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.

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

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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`.

[75] 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.

[76] 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.

[77] 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.

[78] 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 [63], according to three 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 – containing with the requisite instructions and configuration files needed to run all of the experiments detailed in Sec 5 – is publicly available and linked to in the Abstract.
   (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 D.
   (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/folds; we report the standard deviation over the runs for each method-dataset combination in parentheses in Table 1 and Table 2.
   (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, however do provide estimates of the carbon footprint for a single run of our method and of the ERM and FixMatch baselines for the iWildCam dataset.
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 [63] 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?
   Yes, we include a URL link to the code for the paper at the end of the Abstract.
   (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

A Datasets

We evaluate Okapi using three datasets – iWildCam, PovertyMap, and CivilComments – taken from the WILDS 2.0 benchmark [63]. These datasets were chosen specifically due to the poor performance reported by [63] 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 [8]. 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. 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 introduced in [76]. 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 not represented in the source data. The dataset also includes 261K unlabelled images from the same 23 countries.

CivilComments-WILDS is an online-comment dataset adapted from [12], comprising 448K online comments annotated with both a binary indicator of toxicity ($\{\textit{toxic}, \textit{not toxic}\}$) – serving as the target label, y – and the demographic identities mentioned within them – serving as the domain s. Here, $s \in \{0, 1\}^8$ is a binary vector rather than a scalar, with dimensions indicating membership (non-exclusively) to 8 demographic groups, spanning different genders, religions and ethnicities. For the WILDS 2.0 variant of the dataset, [63] introduce an additional corpus of 1551K comments acting as the unlabelled training data belonging to extra domains. While the comments are completely unlabelled, w.r.t. both y and s and thus are not domain-separable at the sample level, the majority (92%) of the comments are known to be sourced from the same documents as those comments comprising the (OOD) labelled test data. As noted in [63], CivilComments-WILDS exhibits label imbalance w.r.t. y; this is amended both therein and herein (as appertains all methods) through the use of class-balanced sampling, though with the minor distinction that for our experiments we ensure each batch is exactly balanced rather by sampling equally from each class, in contrast to [63] who sample hierarchically – sampling y_i uniformly from $\{0, \ldots, |\mathcal{Y}_l|\}$ and then uniformly from $\mathcal{D}_l$, conditioned on y_i – such that balance is achieved only in expectation.

B Relation to Algorithmic Fairness

DG and Algorithmic Fairness overlap in their objective to train a model that yields predictions that are statistically independent of (and thus robust to variations in) domain, when for the latter the domain is taken to be some protected characteristic, such as age or gender, and fairness is measured according to invariance-driven notions of group fairness such as Demographic Parity [26] and Equal Opportunity [32]. Indeed, methods that focus on equalising the empirical risk across subgroups – such as by importance weighting [36, 66] – have featured extensively in both DG [4, 19, 43, 62] and fairness [1, 22, 37] and many approaches to fair representation learning [18, 38, 50, 53, 55] have roots in the former [52] and in the closely-related field of domain adaptation [27]. Beyond this more general equivalence, our work also has ties to notions of individual fairness pioneered by [25] – broadly prescribing that similar individuals be treated similarly – in that our unsupervised loss involves maximising the similarity between inter-domain samples within representation space. This is reminiscent of the operationalisation of individual fairness proposed by [44] that enforces similarity between a given representation and the representations of its neighbouring – in both the input space and according to a between-group (cross-domain) quantile graph – samples.

Table 3: An extended comparison between Okapi and different baselines on two benchmark image datasets. We include both the results of our re-run of the baselines and those of [63]. Both ID and OOD performances are reported. For iWildCam we average over results from 3 different seeds, for PovertyMap we do so over the 5 pre-defined CV folds. Standard deviations are shown in parentheses. The additions relative to Table 1 include results with an offline variant of Okapi – where the matches are generated prior to training from the features of the trained ERM model and then fixed for the course of training – and with the ResNet backbones employed by [63].

Method	**iWildCam** macro F1 ↑		**PovertyMap** worst U/R corr. ↑		worst U/R MSE ↓	
	ID	OOD	ID	OOD	ID	OOD
ERM [63]	47.0 (1.4)	32.2 (1.2)	0.66 (0.04)	0.49 (0.06)	-	-
FixMatch [63]	46.3 (0.50)	31.0 (1.3)	0.54 (0.10)	0.30 (0.11)	-	-
ERM (ConvNeXt)	48.6 (1.1)	33.3 (0.3)	0.72 (0.03)	0.53 (0.09)	0.23 (0.03)	0.35 (0.12)
FixMatch (ConvNeXt)	51.1 (1.0)	35.2 (0.7)	0.50 (0.13)	0.34 (0.12)	0.59 (0.42)	0.88 (0.61)
Okapi (ours; ConvNeXt)	50.6 (0.7)	36.1 (0.9)	0.72 (0.02)	0.55 (0.10)	0.22 (0.02)	0.33 (0.10)
Okapi (offline; ConvNeXt)	48.8 (0.8)	31.7 (0.2)	0.68 (0.02)	0.53 (0.07)	0.26 (0.02)	0.37 (0.13)
Okapi (no calipers; ConvNeXt)	-	-	0.72 (0.02)	0.54 (0.12)	0.22 (0.02)	0.36 (0.14)
ERM (ResNet)	46.5 (0.8)	29.7 (1.0)	0.69 (0.03)	0.53 (0.08)	0.24 (0.04)	0.34 (0.11)
FixMatch (ResNet)	43.0 (2.5)	25.5 (1.4)	0.70 (0.02)	0.53 (0.08)	0.24 (0.02)	0.35 (0.10)
Okapi (ResNet)	46.1 (0.7)	27.8 (0.3)	0.70 (0.04)	0.52 (0.07)	0.23 (0.02)	0.33 (0.10)

C Extended Results

We tabulate in Table 3 an extended version of the results presented in the main text. This includes additional results with the ResNet backbones per [63] (justifying our decision to adopt a ConvNext backbone for our main set of image-dataset results) as well as those for an 'offline' version of Okapi (Okapi (offline)) where the matches are generated prior to training using features of the respective ERM baseline for each dataset. Since the target encoder is necessitated by the need for online match-retrieval, only a single encoder is involved in Okapi (offline); in binary cases, the algorithm is then identical to the one proposed by [57] with the exception that consistency is still enforced via distance in encoding space rather than with a JSD loss on the predictive distributions which fails to generalise to regression tasks such as PovertyMap.

D Implementation details

Data Augmentation We follow [63] when defining the augmentations for the the WILDS datasets. In the case of PovertyMap-WILDS we corroborate the original finding that data-augmentation adversely affects performance, and, in light of this, elect only to use data-augmentation for FixMatch where it is needed to generate the weak and strong views used in computing the consistency loss. Since Okapi uses an NN-based approach for generating these views, it is decoupled from the augmentation strategy and problems that can arise from its misspecification.

Architecture For our image experiments, contrary to [63], we opt to use the recently proposed ConvNeXt architecture [47], finding this change to provide large performance gains and to be crucial in enabling semi-supervised methods to surpass the ERM baseline. This is in line with [39] who similarly found that a change of architecture (combined with large-scale pre-training) could greatly bolster performance on the iWildCam dataset. More precisely, we use the *tiny* variant of ConvNeXt, pre-trained on ImageNet 1k, as the initial backbone for our models. We compose this with a single fully-connected layer to construct the complete predictor both for the target and the propensity score. For our CivilComments experiments, in contrast, we do not diverge from [63] in our choice of architecture, with all models trained with a pre-trained DistilBERT [64].

Optimisation For optimising all models, we use the AdamW optimiser [49] coupled with a cosine annealing schedule without warm restarts [48]. We set the initial learning to be 1×10^{-4} across the board, and forgo the use of weight decay. Models are trained for 120K, 30K, and 20K iterations for iWildCam, PovertyMap, and CivilComments, respectively. The decay coefficient, ζ, for the target encoder's exponential moving average is initialised to ζ_{start} and is linearly increased to ζ_{end} over the course of training. For PovertyMap we set ζ_{start} and ζ_{end} to be 0.996 and 1.0, respectively; for iWildCam, we set ζ_{start} and ζ_{end} to be 0.999 and 0.999, respectively, resulting in a fixed value of

ζ; 1.0, respectively; for CivilComments, we set ζ_{start} and ζ_{end} to be 0.996 and 0.996, respectively, again resulting in a fixed value of ζ. We similarly warm up the pre-factor for the consistency loss, λ, according to a linear schedule during the first 10% of training to allow a period for the encoder to learn meaningful relations between samples through the supervised loss before bootstrapping with the consistency loss, with a final value of 1.

Matching In order to determine suitable hyperparameters, ξ, for CaliperNN, we perform a grid-search in the static setting, using a fixed model. Specifically, we use the backbone of an ERM-trained model as the encoder with which to generate the queries and keys for matching. The quality of matching with a given instantiation of ξ is measured using two metrics commonly used in the statistical matching literature: *Variance Ratio* (VR) and *Standard Mean Differences* (SMD) [61]. Both metrics operate on pairs of domains, but can be generalised to work when s is non-binary by simply aggregating over all pairwise results. For a given pair of domains, VR is defined as the ratio of the variances of the covariates between the two domains, with an ideal value of 1, while SMD is defined as the difference in their covariate means, normalised by the standard deviation for each covariate, and is to be minimised. While our proposed method is applicable whether S is binary or categorical, for the experiments in this paper we take advantage of the fact that the WILDS datasets specify splits with non-overlapping domains and match from $\mathcal{D}_l \to \mathcal{D}_u$ and in the reverse direction (from $\mathcal{D}_u \to \mathcal{D}_l$). This decision was based on preliminary experiments which found the binary variant generally enjoyed more stable optimisation, something which future work should seek to rectify. In the case of PovertyMap, however, the training splits themselves do not satisfy the aforementioned requirement of being sourced from mutually exclusive sets of domains and we instead treat the OOD validation set as $\mathcal{D}_u$ (and treat it as being unlabelled w.r.t. y, in that it is only used for $\mathcal{L}_{\text{unsup}}$).

E Additional Matching Examples

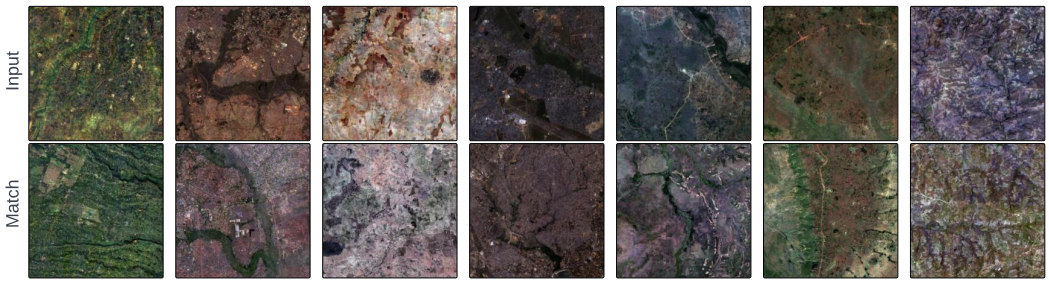


Figure 4: Examples of input (labelled) images and their 1-NN matched (unlabelled) images retrieved using CaliperNN from the PovertyMap-WILDS dataset. Here, we match images from the labelled-train set to images from the OOD-validation set, taking advantage the fact that their domains are disjoint.

F Ablations

We supplement the ablation experiment on the use of calipers featured in the main text with additional experiments concerning effect of the number of nearest neighbours (k), the relative importance of the two (fixed and std) calipers, and the feasibility of using the online encoder as the query-generator instead of the target encoder. The results of these experiments are tabulated in 4 with the key takeaways being:

1. The number of neighbours used for computing the consistency loss has little impact – according to the given level of precision – on the performance of Okapi along all axes.
2. While disabling the calipers altogether considerably harmed performance, using only the std. caliper allows us to recover the performance of the complete algorithm, (`Okapi (k=5)`), whereas the same is not true for the fixed caliper which, while aiding performance compared to the no-caliper baseline, falls short of that benchmark. A caveat attached to these conclusions, however, is that the selected values for ξ are likely suboptimal in the online

Table 4: Ablation experiments for Okapi conducted using the PovertyMap-WILDS dataset. Specifically, we assess the importance of four elements of our proposed method: the number of nearest neighbours used in computing $\mathcal{L}_{\text{unsup}}$ (k), the use (enabled/disabled) of the fixed- and std-calipers in CaliperNN (considering these to be separate components), and which encoder (online or target) is used to generate the queries for statistical matching (with use of the target encoder 'TE queries' being the default and 'OE queries' denoting the alternative). Both ID and OOD performances are reported. The results are computed by aggregating over the results for each of the 5 pre-defined cross-validation folds. We report the average and standard deviation value across replicates of the metric of interest.

(a) CaliperNN ablations.

Method	**PovertyMap**			
	worst U/R corr. ↑		worst U/R MSE ↓	
	ID	OOD	ID	OOD
Okapi (k=1)	0.72 (0.02)	0.55 (0.10)	0.22 (0.02)	0.33 (0.10)
Okapi (k=5)	0.72 (0.02)	0.55 (0.10)	0.22 (0.02)	0.33 (0.10)
Okapi (k=10)	0.72 (0.02)	0.55 (0.09)	0.22 (0.02)	0.33 (0.10)
Okapi (k=5, no calipers)	0.72 (0.02)	0.54 (0.12)	0.22 (0.02)	0.36 (0.14)
Okapi (k=5, no std caliper)	0.72 (0.02)	0.54 (0.12)	0.22 (0.02)	0.35 (0.14)
Okapi (k=5, no fixed caliper)	0.72 (0.02)	0.55 (0.10)	0.22 (0.02)	0.33 (0.10)

(b) Query-generator ablation (target encoder (TE) vs. online encoder (OE)).

Method	**PovertyMap**			
	worst U/R corr. ↑		worst U/R MSE ↓	
	ID	OOD	ID	OOD
Okapi (TE queries)	0.72 (0.02)	0.55 (0.10)	0.22 (0.02)	0.33 (0.10)
Okapi (OE queries)	0.72 (0.02)	0.55 (0.10)	0.23 (0.02)	0.34 (0.10)

setting, given that they were optimised for the static setting: with improved selection of ξ, either by learning it jointly with the model's parameters (using, for instance, the perturbed maximum method [9] to overcome the non-differentiability of the k-NN and thresholding operations), in an amortised fashion, or optimising it on a per-iteration basis.

3. While less appealing from a conceptual standpoint, due to the mismatch between the networks used to generate the queries and keys, from an empirical standpoint it is perfectly feasible to use the online encoder to generate the queries for statistical matching instead the target encoder while experiencing minimal degradation in performance. This is particularly relevant when one wishes to perform the matching in only one direction (e.g. $\mathcal{D}_l \rightarrow \mathcal{D}_l$) due to the reduction in redundant encoding, with each encoder only encoding its respective subset of the data (e.g. f_θ only encodes samples from $\mathcal{D}_l$ and f'_θ only encodes samples from $\mathcal{D}_u$)

G Pseudocode

We provide Pytorch-style [54] pseudocode for the CaliperNN (described in 3.2) and online-learning (described in 3.3) algorithms in Algorithm 1 and Algorithm 2, respectively. In both cases, we restrict the pseudocode to the special case of binary domains – practically achieved by using the labelled/unlabelled as a proxy for domain – for ease of illustration. The CaliperNN algorithm can be generalised freely to multiclass cases by considering pairwise interactions between the propensity scores for each domain for applying the calipers and by computing the pairwise inequalities between $\mathbf{s}_q$ and $\mathbf{s}_k$ (giving the connectivity matrix $(\mathbf{s}_q \cdot \mathbf{1}^T) \neq (\mathbf{1} \cdot \mathbf{s}_k^T)$, where $\mathbf{1}$ denotes the ones vector of the same shape as its multiplicand and mediates broadcasting) for enforcing the cross-domain constraint.

Algorithm 1: Pytorch-style pseudocode for the CaliperNN matching algorithm for the special case where the domain is binary. The algorithm generalises freely to arbitrary numbers of domains however we restrict ourselves to the binary version here for illustrative purposes.

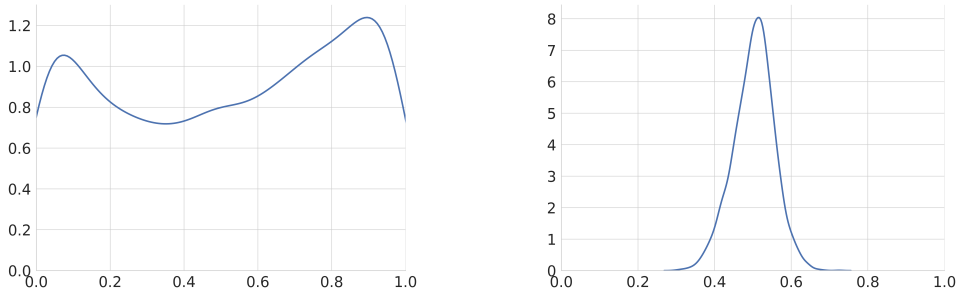
```
def binary_caliper_nn(
    x_query, # samples to be used as the queries for matching
    s_query, # binary labels indicating the domain of x_query
    x_key, # samples to which the query samples may be matched to.
    s_key, # binary labels indicating the domain of x_key
    ps_query, # propensity scores associated with x_query
    ps_key, # propensity scores associated with x_key
    t_f, # threshold for the fixed caliper
    t_sigma, # number of standard deviations at which to threshold
    k # number of neighbours to attempt to retrieve per query
):
    anchor_inds, positive_inds = [], []
    for direction in (0, 1): # which domain (0 or 1) to treat as the 'anchor'
        key_mask = s_key != direction
        # exclude samples with propensity scores outside the valid range
        # determined by t_f: (1 - t_f, t_f)
        fc_mask = (ps_query > (1 - t_f)) &  (ps_query < t_f)
        anchor_mask = fc_mask & (s_query == direction)
        queries_x_filtered = queries.x[anchor_mask]
        ps_query_filtered = ps_query[anchor_mask]
        fc_mask = (ps_key > (1 - t_f)) & (ps_key < t_f)
        key_mask &= fc_mask
        ps_key_filtered = ps_key[key_mask]
        # 2-norm distance between unfiltered propensity scores
        dists_ps = cdist(ps_query_filtered, ps_key_filtered, p=2)
        # 2-norm distance between the filtered anchors and keys
        dists_x = cdist(queries_x_filtered, x_key[key_mask], p=2)
        # compute sigma as the mean of the per-domain standard deviations
        std_ps = (0.5 * (ps_query_filtered.var() + ps_key_filtered.var())).sqrt()
        std_threshold = t_sigma * std_ps
        # filter out any samples that violate the std-caliper
        dists_x[dists_ps > std_threshold] = float("inf")
        nbr_dists, nbr_inds = dists_x.topk(dim=1, largest=False, k=k)
        # filter out queries not yielding the requisite number of matches (k)
        is_matched = ~nbr_dists.isinf().any(dim=1)

        anchor_inds.append(anchor_mask.nonzero()[is_matched])
        positive_inds.append(key_mask.nonzero()[nbr_inds[is_matched]])

    return cat(anchor_inds, dim=0), cat(positive_inds, dim=0)
```

H Matching for PACS dataset

In this section we perform some initial experiments on PACS dataset [45] (using features extracted for a pre-trained CLIP [56] model) to show how the temperature scaling can be used to smooth the propensity score distribution to better control how many sample are discarded during matching. There are 1,670 *photo*, 2,048 *art painting*, 2,344 *cartoon*, and 3,929 *sketch* in the dataset. Here will evaluate the results of matching across the two domains *photo* and *art painting* as well as across *photo* and *sketch*. In Fig. 5 and Fig. 6 we compare the shape of the estimated propensity score with its scaled version using a temperature value of 10. As we can see, in the case of a distribution with extremely heavy tails (photo, sketch), the effect of smoothing the distribution is that when a fixed caliper is applied most of the samples are retained. On the other hand, when the initial distribution is

smoother, a temperature of 10 is extreme, having the effect of transforming the bimodal distribution to a unimodal one. Additionally, we tabulate in Table 5 the number of matched pairs retrieved when matching across the two domains photo and sketch; here we can see that by increasing the temperature we smooth the estimated propensity score distribution and thereby retain more samples. Similarly, we can retrieve more pairs by reducing the fixed caliper threshold. We also analyse the case of matching across the two domains photo and art painting. Using a fixed caliper defined defined by a threshold $t_f = 0.1$ and no temperature scaling (i.e. $\tau = 1$) the algorithm retrieves 1,142 pairs matching in the direction photo $\rightarrow$ art and 1,501 in the direction art $\rightarrow$ photo.

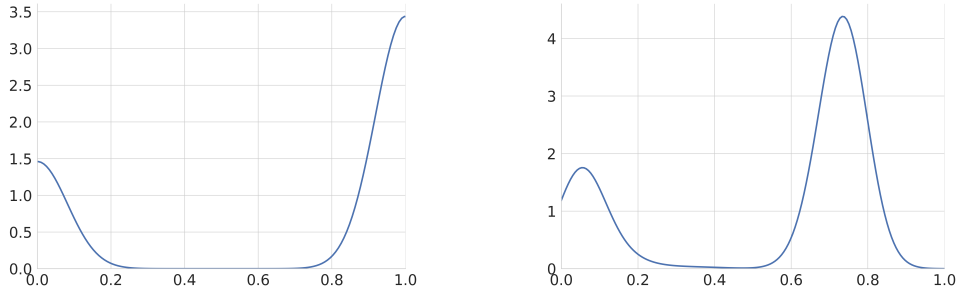
In Fig. 7 and Fig. 8 we show examples of matching pairs found using our CaliperNN algorithm. Although the features were not fine-tuned on PACS, we can see a few examples of intraclass matching. For the photo-art painting application we can see preservation in colour and background; while in the photo-sketch case shape and pose.



(a) No temperature scaling ($\tau = 1$).

(b) Temperature scaling, with $\tau = 10$.

Figure 5: Estimated propensity score distribution of *photo* and *art painting* on the PACS dataset. We compare (a) the original distribution ($\tau = 1$) and (b) the temperature-scaled distribution ($\tau = 10$). Here, the large temperature has the effect of transforming a bimodal distribution into a unimodal one.



(a) No temperature scaling ($\tau = 1$).

(b) Temperature scaling, with $\tau = 10$.

Figure 6: Estimated propensity score distribution of *photo* and *sketch* on the PACS dataset. We compare (a) the original distribution ($\tau = 1$) and (b) the temperature-scaled distribution ($\tau = 10$). Here, the large temperature has the effect of smoothing the distribution.

I Energy and Carbon Footprint Estimates

To highlight the efficiency of Okapi, we provide estimates in 6 of the carbon footprint associated with the running of it and of the ERM and FixMatch baselines on the iWildCam dataset, using the same hyperparameter configuration used to generate the results in the main text. The runs were conducted in a controlled fashion, using the computing infrastructure and device count in all cases.

Figure 7: Examples of input (photo) images and their 1-NN matched (art paint) images retrieved using CaliperNN from the PACS dataset.

Figure 8: Examples of input (photo) images and their 1-NN matched (sketch) images retrieved using CaliperNN from the PACS dataset.

Algorithm 2: Pytorch-style pseudocode for the online learning algorithm for the special case where the labelled and unlabelled datasets are treated as the domains. The algorithm generalises freely to arbitrary numbers of domains however we restrict ourselves to the binary version here for illustrative purposes.

```
# online_encoder: online encoder
# predictor_head: online predictor head
# propensity_scorer: online propensity scorer
# target_encoder momentum encoder (frozen)
# n_m: memory-bank capacity
# zeta: decay rate of the EMA updates
# tau: temperature-scaling parameter for the propensity scores.
# t_f: fixed caliper threshold for CaliperNN
# t_sigma: number of standard deviations at which to threshold in CaliperNN
# l_sup: supervised loss function
# k: number of matches to retrieve per query
# lambda_: loss pre-factor for the unsupervised loss
# D: Dimensionality of the encodings.

feature_mb = empty(n_m, D) # memory bank storing momentum-encoded features
label_mb = empty(n_m) # memory bank storing domain labels associated with feature_mb
 # load minibatches with B_l labelled samples and B_u unlabelled samples
for x_l, y, x_u in train_loader:
    # EMA update: \theta^\prime_t = \zeta \theta^\prime_{t - 1} + (1 - \zeta) \theta_t
    ema_update(target_encoder, online_encoder, zeta)
    features_o_l = online_encoder(x_l) # f_\theta(x_l) -> z_l
    features_t = target_encoder(cat([x_l, x_u])) # f_\theta(x_l \cup x_u) -> z_q^\prime
    y_hat = predictor_head(features_o_l) # g_\phi(z_l) -> \hat{y}
    features_o_u = online_encoder(x_u) # f(x_u) -> z_u
    features_o = cat([features_o_l, features_o_u]) # z_q := z_l \cup z_u
    # normalize the encodings to unit vectors.
    features_o_n = normalize(features_o, p=2, dim=1)
    queries = normalize(features_t, p=2, dim=1)
    # we treat x_l and x_u as coming from domains indexed by 0 and 1, respectively
    labels_l_q = ones(len(x_l)) # ones-vector of size B_l
    labels_u_q = zeros(len(x_u)) # zeros-vector of size B_u
    labels_q = cat([labels_l_q, labels_u_q])
    mb_mask = is_empty(label_mb) # mask indicating which elements of the MB are filled
    labels_k = cat([labels_q, label_mb[mb_mask].clone()])
    # keys are the union of the queries and the memory-bank-stored features
    keys = cat((queries, feature_mb[mb_mask].clone()), dim=0)
    feature_mb.push(queries) # update the feature memory bank
    label_mb.push(labels_q) # update the label memory bank
    logits_ps_k = propensity_scorer(keys) # h_\psi(z_k) -> e_k
    loss_ps = xent(logits_ps_k, labels_k) # (binary) cross-entropy loss
    # tempered logistic function: 1 / (1 + exp(-logits_ps_k / tau))
    logits_ps_k = sigmoid(logits_ps_k / tau)
    logits_ps_q = logits_ps_k[:len(queries)]
    # filter and match queries with (binary) CaliperNN
    inds_a, inds_p = binary_caliper_nn(
        features_t_n, labels_q, keys, labels_k,
        logits_ps_q, logits_ps_k, t_f, t_sigma, k
    )
    # compute the unsupervised loss (d(z_q, z_n)) for all matched queries
    z_q, v_k = features_o[inds_a], keys[inds_p]
    match_rate = len(z_q) / len(features_o)
    loss_u = match_rate * (z_q.unsqueeze(1) - v_q).pow(2).sum(-1).mean()
    loss = l_sup(y_hat, y) + lambda_ * loss_u + loss_ps # aggregate loss
    loss.backward() # compute gradients
    update(online_encoder, predictor_head, propensity_scorer) # optimizer updates
```

Table 5: Analysis of the number of the retrieved matched pairs when matching across the two domain *photo* and *sketch* on the PACS dataset. The fixed caliper threshold and temperature scaling can be used to smooth the propensity score distribution and effect the number of pairs.

Fixed Caliper (t_f)	**Temperature** (τ)	**photo** $\rightarrow$ **sketch**	**sketch** $\rightarrow$ **photo**
0.1	1	0	0
0	1	1540	3929
0.01	1	6	9
0.01	1.3	14	56
0.01	1.8	25	574
0.01	2.5	41	3082
0.01	10	1540	3929
0.1	10	298	3929

Table 6: Comparison of the estimated carbon footprint (kgCoeq) of Okapi with the ERM and FixMatch baselines per replicate of the iWildCam dataset. For the controlled training conducted to enable fair computation of these estimates, we used a private infrastructure with an estimated carbon efficiency of 0.432 kgCOeq/kWh and RTX 3090 GPUs, each job being run on a single GPU, coupled with four data-loading workers.

Method	**kgCOeq** $\downarrow$
ERM	1.36
FixMatch	2.12
Okapi (ours)	1.97