Does the dataset meet your expectations? Explaining sample representation in image data*

Dhasarathy Parthasarathy[1,2] and Anton Johansson[2]

[1] Volvo Group, Sweden
dhasarathy.parthasarathy@volvo.com
[2] Chalmers University of Technology, Sweden
johaant@chalmers.se

Abstract. Since the behavior of a neural network model is adversely affected by a lack of diversity in training data, we present a method that identifies and explains such deficiencies. When a dataset is labeled, we note that annotations alone are capable of providing a human interpretable summary of sample diversity. This allows explaining any lack of diversity as the mismatch found when comparing the *actual* distribution of annotations in the dataset with an *expected* distribution of annotations, specified manually to capture essential label diversity. While, in many practical cases, labeling (samples → annotations) is expensive, its inverse, simulation (annotations → samples) can be cheaper. By mapping the expected distribution of annotations into test samples using parametric simulation, we present a method that explains sample representation using the mismatch in diversity between simulated and collected data. We then apply the method to examine a dataset of geometric shapes to qualitatively and quantitatively explain sample representation in terms of comprehensible aspects such as size, position, and pixel brightness.

Keywords: Sample selection bias · Explainability · Outlier detection.

1 Introduction

Choosing the right data has always been an important precondition to deep learning. However, with increasing application of trained models in systems which are required to be dependable ([20], [2]), there is increasing emphasis on making this choice well-informed ([4], [36]). Consider the perception system of a self-driving vehicle which is partially realized using deep learning and is expected to dependably detect pedestrians. To ensure that the system meets such an expectation, it is necessary to choose training and validation sets that adequately cover critical scenarios ([31], [34]) like residential areas and school zones, where the vehicle is likely to meet pedestrians. Choosing, conversely, a dataset that contains only scenes of motorway traffic, which does not cover many scenarios involving pedestrians, is likely to produce a trained model that violates

* Work supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

expectations on pedestrian detection. Scenarios covered by a dataset may be considered sufficient when samples of adequate variety are represented in it. With practical image datasets typically being high-dimensional and large, posing and evaluating explicit conditions on the adequacy of sample representation is not straightforward.

Interpretable assessment of sample representation Consider a traffic dataset $\mathcal{S}$ of images $X_i \sim P(X|Y)$ and annotations $Y_i \sim P(Y)$. A major practical concern in such datasets is whether it adequately represents corner cases like intersections with stop signs, roundabouts with five exits, etc. With the true/target distribution of traffic scenes $P(X, Y)$ clearly containing instances of such cases, *any under-representation* in $\mathcal{S}$ can be broadly framed as shortcomings in data collection and processing, otherwise known as *sample selection bias*([37]). Given that the dataset is eventually used to train a model that is deployed in a safety-critical system, engineers may actively seek to properly comprehend and account for such bias. But how does one express such bias in human interpretable terms? One clue comes from annotations $Y_i \sim P(Y)$. In typical traffic datasets, Y encodes object class labels and bounding box positions. If necessary and feasible, Y can be expanded to contain information such as location, lighting conditions, weather conditions, etc. When Y is adequately detailed, the distribution of annotations $P_{\mathcal{S}}(Y)$ clearly becomes a reasonable, low-dimensional, and therefore a human interpretable measure of sample representation in $\mathcal{S}$. Engineers can exploit this notion to *specify* a distribution of annotations $P_{\mathcal{T}}(Y)$, expressing the sample representation that is *expected* in the dataset. While the target distribution of annotations $P(Y)$ may be unknowable, $P_{\mathcal{T}}(Y)$ is an explicit declaration of the sub-space that the dataset is expected to cover at the minimum. If $\mathcal{S}$ is equivalently labeled, then selection bias (and thereby sample under-representation) is simply given by the mismatch between expectations $P_{\mathcal{T}}$ and reality $P_{\mathcal{S}}$. In practice, however, due to the effort and expense involved in labeling, $\mathcal{S}$ may either lack labels or may be completely unlabeled, meaning that $P_{\mathcal{S}}(Y)$ is often unavailable. Combining simulation, outlier detection, and input attribution, we show that it is possible to explain sample representation in a comprehensible low-dimensional form, even when annotations are not explicitly available in $\mathcal{S}$.

Contributions Delving into the less-explored area of *explaining* sample representation in a dataset, we demonstrate a method that

- explains sample representation in interpretable terms for annotated data
- uses parametric simulation and outlier detection to do the same for non-annotated data

In addition to visualization, we propose a quantitative explanation of sample under-representation using an *overlap index*. Also, unlike existing methods that mainly address imbalances in available data, ours can explain gaps in the availability of data. Such an explanation helps engineers better understand data as a crucial ingredient of the training process. Downstream, this helps them re-asses data collection methods and to verify, reason, or argue about – at times a re-

quirement for standards compliance [3] – the overall dependability of the model trained with this data. Data and code used in this work are publicly available[3].

2 Explaining sample representation using annotations

Visualizing sample representation We now introduce a simple running example of examining sample representation in a dataset $\mathcal{S}$ containing images of two hand-drawn shapes[4] – circles and squares (Figure 1). With the shape as the sole available label, one can define $\mathcal{S} = \{(X_i, Y_i^1)\},\ i = 1...N$, where X_i is a grayscale image of size $(128, 128)$ and $Y_i^1 \in K = \{0, 1\}$ is the shape label, corresponding to circle and square respectively. Understanding sample representation in this dataset may be necessary when it is a candidate for training a model that, for example, either recognizes or generates shapes. To ensure dependable model performance, system designers may want to confirm that images of adequate variety are represented in $\mathcal{S}$. In a dataset of grayscale geometric shapes, it is intuitive to analyze sample representation in terms of concerns such as the size and position of the shapes on the image canvas, and the average brightness of pixels in the shape. All these concerns can be captured by defining a 6-d annotation vector $Y = (Y^1, ..., Y^6)$, including shape-type, which is known. With $\mathcal{U}$ denoting the discrete uniform distribution, designers can begin with defining an expected spread of shape-size using a latent label $Y^S \sim \mathcal{U}\{30, 120\}$, denoting the side-length in pixels of a square box bounding the shape. This can be followed by defining expectations on the spread of (i) the top-left corner of the bounding box, $Y^2, Y^3 \sim \mathcal{U}\{0, 128 - Y^S\}$, (ii) the bottom-right corner of the bounding box $Y^4, Y^5 \sim \mathcal{U}\{Y^S, 128\}$, and (iii) the average pixel brightness $Y^6 \sim \mathcal{U}\{100, 255\}$. Put simply, $P_{\mathcal{T}}(Y)$ expects shapes of a specified range of sizes and brightness to be uniformly represented in the dataset $\mathcal{S}$. All positions are also expected to be uniformly represented, as long as the shape can be fully fit in the image canvas.

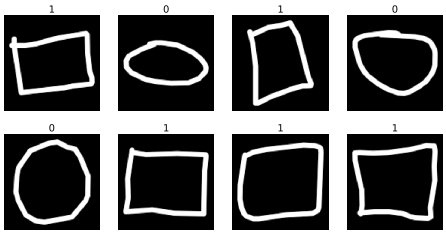


Fig. 1: Samples from the dataset $\mathcal{S}$. Only the class label Y^1 is available

To illustrate the idea of explaining sample representation using annotations, an automatic labeling scheme $Y_i = L(X_i)$ is used to produce complete 6-d annotations for X_i. For circles and squares, it is easy to define a scheme that looks

[3] https://github.com/dhas/SpecCheck
[4] Collected from Quick, Draw! with Google – https://quickdraw.withgoogle.com/data

at the extent of the shape and draws bounding boxes. The average brightness is given by the mean of non-zero pixels in the canvas. The availability of labels Y_i helps assemble the actual distribution of samples in the dataset $P_{\mathcal{S}}(Y)$, allowing direct comparison with expectations $P_{\mathcal{T}}(Y)$. Jointly visualizing label distributions for each shape (Figure 2) shows that, along all design concerns Y^j, the spread of $P_{\mathcal{T}}$ (marked black) is much wider than the very narrow $P_{\mathcal{S}}$ (marked red). This shows that, while $P_{\mathcal{T}}$ expects shapes of a broad range of sizes, positions and brightness to be represented, $P_{\mathcal{S}}$ is clearly biased and massively over-represents large and bright shapes located in the center on the canvas. As long as the annotation vector Y is of manageable length, joint visualization becomes an interpretable qualitative explanation of sample representation in the dataset.

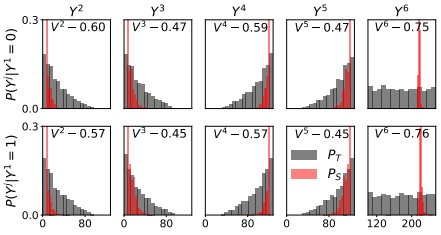


Fig. 2: Explaining sample representation

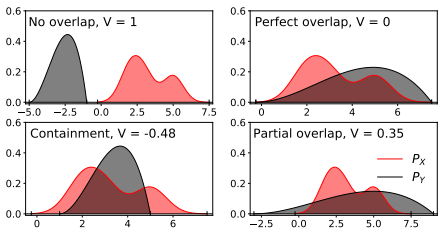


Fig. 3: Illustration of $V(P_X, P_Y)$

Quantifying sample representation By framing sample selection bias, and thereby sample under-representation, as the mismatch between expected and true label probability distributions, it becomes possible to quantify it using measures of statistical similarity. Choosing the right measure, however, requires a proper understanding of the nature of each distribution. Having calculated it using true labels of each sample, it is clear that $P_{\mathcal{S}}(Y)$ represents the actual sample distribution in $\mathcal{S}$. The distribution of expectations $P_{\mathcal{T}}$ is of a slightly different nature and, to better understand it, let us consider the expectation $P_{\mathcal{T}}(Y^6) = \mathcal{U}\{100, 255\}$, placed on the representation of average brightness of shapes in the dataset. While the expectation on brightness being spread between specified lower and upper limits is strict, imposing the spread to be uniform is arbitrary. This is a deliberate measure of simplification to ease the considerable burden in modeling expectations $P_{\mathcal{T}}$ and let it capture the critical range of interest in the target distribution. Put simply, expected sample representation is primarily encoded by the *support* (1) of $P_{\mathcal{T}}$. By specifying strict support, but arbitrary distribution of mass, sample representation can be quantified as the level of *overlap* between the actual sample distribution $P_{\mathcal{S}}$ and the expected sample representation $P_{\mathcal{T}}$. To achieve this, we propose an overlap index $V(P_X, P_Y)$ (2), which is a measure of whether the supports of two distributions are similar. With set difference Δ and 1-d Lebesgue measure (length) of a set λ, V is essentially the Steinhaus distance [11] with an added term I to make $-1 < V < 0$ indicate containment of P_Y within P_X. When not contained, for some positive likelihood

in both distributions, as illustrated in Figure 3, $V = 0$ when they exactly overlap, $V = 1$ when they do not overlap, and $0 < V < 1$ when the overlap is partial. Indices $V^j(P_{\mathcal{T}})$ (3) quantitatively measure the level of overlap between true and expected distributions for each label. Complementing the visual explanation, overlap indices $0.4 < V^j(P_{\mathcal{T}}) < 1$ seen in Figure 2, indicate that there is only slight partial overlap between expectations and reality, confirming notable sample selection bias and, therefore, significant sample under-representation.

$$R_X = \{x \in \mathbb{R} : P_X(x) > 0\} \tag{1}$$

$$V(P_X, P_Y) = I \ \frac{\lambda(R_X \ \Delta \ R_Y)}{\lambda(R_X \ \cup \ R_Y)}, \quad I = \begin{cases} -1 & R_Y \subset R_X \\ +1 & otherwise \end{cases} \tag{2}$$

$$V^j(P) = V(P_{\mathcal{S}}(Y^j \ |Y^1), P(Y^j \ |Y^1)), \quad j = 2...6 \tag{3}$$

It is therefore clear that, given the expected representation and actual distribution of labels in the dataset, it is possible to comprehensibly explain sample under-representation both visually and quantitatively. However, the overlap index, which eschews mass and uses only support, is an incomplete measure of sample selection bias, the pros and cons of which is discussed in Section 4.

3 Explaining sample representation using simulation

The dataset $\mathcal{S}$ contains information X_i in the image domain, while lacking information Y_i in the annotations domain. Expectations, on the contrary, are expressed using annotations $\hat{Y}_i \sim P_{\mathcal{T}}(Y)$, but lacks images. It is this gap in information that prevents estimation of sample under-representation by direct comparison. There are two possible ways to bridge this gap, one of which is the labeling scheme $Y_i = L(X_i)$ introduced earlier. Another way could be to generate images $\hat{X}_i = G(\hat{Y}_i)$, which is essentially *parametric simulation*. In this case of circles and squares, it is possible to use a graphics package[5] to draw shapes using size, position, and brightness labels as parameters. We, in fact, choose this simple dataset because both labeling and simulation of samples are easy, helping illustrate both ways of bridging the gap and cross-checking the plausibility of estimating sample representation. In many practical cases, however, the right method to bridge the gap is difficult to judge since the relative expense is domain and problem specific. Addressing those numerous instances where unlabeled data is available and labeling is expensive, we now show that it is possible to bridge the gap using simulation. This is done using a two-step process, described below, of (i) detecting outlier annotations and (ii) estimating marginal sample representation.

Step 1 - Detecting outlier annotations To a dataset that mainly contains large, centered shapes, can simulated small off-centered shapes appear as outliers? In order to explore this simple notion, we pose the following outlier hypothesis - *a test annotation $\hat{Y}_i$, that is unlikely to be observed in $\mathcal{S}$, maps to a simulated test sample $\hat{X}_i = G(\hat{Y}_i)$, that appears as an outlier to $\mathcal{S}$.* Bridging the gap

[5] We use OpenCV – https://opencv.org/

by simulating shape images that follow specified expectations $P_{\mathcal{T}}$, the problem of detecting sample selection bias turns into one of detecting outlier images. The hypothesis is realized by an outlier detector $E_{\mathcal{S}}$ (Figure 4) that samples test annotations from $P_{\mathcal{T}}$ and maps them into images using a simulator, creating a test set $\mathcal{T} = \{(\hat{X}_i, \hat{Y}_i)\},\ i = 1...M$ (examples in Figure 5). Following [17], the subsequent assessment of whether under-represented simulated images appear as outliers to $\mathcal{S}$ is done using the predictive certainty of a shape label classifier $F(X) = P_{\mathcal{S}}(Y^1|X;\theta)$, trained on the dataset $\mathcal{S}$. The complete detector of outlier annotations $E_{\mathcal{S}}$ is formally described below in (4), where F_k is the logit score for the k^{th} shape and T is the temperature parameter which, as shown later, eases the detection process. With F using a softmax output layer, we use maximum softmax score as the measure of certainty. Put simply, with sets of outlier and familiar annotations (5), the outlier hypothesis asserts that a good detector $E_{\mathcal{S}}$ assigns low scores S_i for outlier annotations $\hat{Y}^-$ and high scores for familiar ones $\hat{Y}^+$.

$$S_i = E_{\mathcal{S}}(\hat{Y}_i, F, T) = \max_{k \in K} \frac{exp(F_k(G(\hat{Y}_i))/T)}{\sum_{k \in K} exp(F_k(G(\hat{Y}_i))/T)}, \hat{Y}_i \sim P_{\mathcal{T}}(Y), K = \{0, 1\} \quad (4)$$

$$\hat{Y}^- = \{\hat{Y}_i : P_{\mathcal{S}}(\hat{Y}_i) = 0\}, \qquad \hat{Y}^+ = \{\hat{Y}_i : P_{\mathcal{S}}(\hat{Y}_i) > 0\} \quad (5)$$

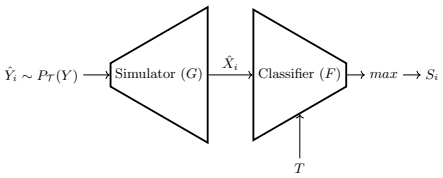


Fig. 4: Detecting outlier annotations

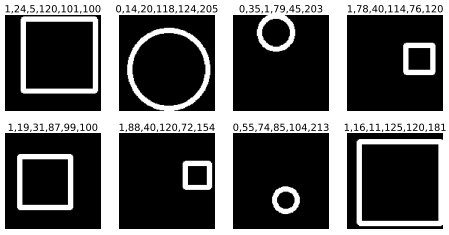


Fig. 5: Samples $\hat{X}_i$ from the test set $\mathcal{T}$

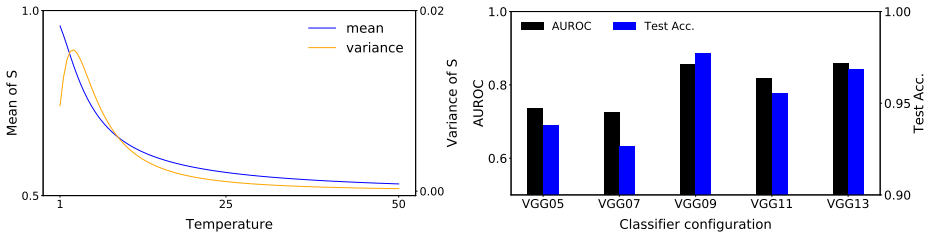
To test the outlier hypothesis, four variants of the classifier F, all of which follow the VGG architecture [33], are used. Classifiers mainly differ in the number of layers, with VGG05 (5 layers) and VGG13 (13 layers) being the shallowest and deepest respectively. Each F is trained[6] for 5 epochs on $\mathcal{S}$ with 50k samples using the Adam optimizer [18] to achieve validation accuracy (on a separate set of 10k samples) greater than 97%. However, [14] shows that deep neural nets tend to predict with high confidence, making raw maximum softmax scores poor measures of predictive certainty, and a simple way to mitigate this is temperature scaling, i.e. setting $T > 1$, in (4). As seen in Figure 6a, scores S_i are tightly clustered at $T = 1$ with relatively low variance, which makes it difficult to identify differences in predictive certainty between familiar and outlier annotations. There is, however, a range of temperatures at which scores are better spread and can exaggerate these differences. While a temperature that maximizes the

[6] Each classifier trains within 10 – 15 minutes on an NVidia GTX 1080 Ti GPU

variance of the score distribution seems appropriate, as seen in Figure 6a, scaling also reduces its mean. Therefore a safeguard may be necessary to prevent the mean certainty score from reducing to a level that questions the confidence of predictions. These twin requirements can be achieved by the search objective (6), which ensures a good spread in scores S_i, while keeping its mean close to the chosen safeguard S^T.

$$\begin{aligned} T^* &= \underset{T}{argmin}\ L^T - L^V, \quad L^T = \left(\mu_S - S^T\right)^2 \\ L^V &= \frac{\sum_{i=1}^{M}\left(E_{\mathcal{S}}(\hat{Y}_i, F, T) - \mu_S\right)^2}{M}, \quad \mu_S = \frac{\sum_{i=1}^{M} E_{\mathcal{S}}(\hat{Y}_i, F, T)}{M} \end{aligned} \tag{6}$$

Upon temperature scaling with T^*, the effectiveness of the detector $E_{\mathcal{S}}$ in separating outlier annotations $\hat{Y}^-$ from familiar ones $\hat{Y}^+$ can be measured using the Area Under Receiver Operating Characteristic (AUROC). This is shown for each F, averaged over 5 separate training runs, in Figure 6b. Based on an informal grading scheme for classifiers using AUROC score suggested in [17][7], detectors using VGG05 and VGG07 receive a 'fair' grade in identifying outlier annotations, while the deeper networks get 'good' grades. The best outlier detectors, with AUROC ≈ 0.85, are those with F as VGG09 and VGG13. These results clearly endorse the viability of the outlier hypothesis that simulated images that are under-represented in $\mathcal{S}$, in terms of specified design concerns, appear as outliers to the right classifier trained on $\mathcal{S}$. While $P_{\mathcal{S}}$, derived from labeling, is used as a benchmark to test the outlier hypothesis, it is important to observe that (i) classifiers that are good at outlier detection are, as seen in Figure 6b, those that have the highest accuracy in predicting shape labels on the test set $\mathcal{T}$, and (ii) the temperature T^*, at which the classifiers become good outlier detectors, depends only upon the statistical properties of scores S_i. Together, these observations mean that a good detector of under-represented annotations can be assembled using only simulation, without any need for labeling.



(a) Effect of temperature scaling on the distribution of uncertainty scores F=VGG13

(b) AUROC for detecting outlier annotations per classifier at $T = T^*$ and $S^T = 0.7$

Fig. 6: Testing the novelty hypothesis

[7] Quality of classification based on AUROC score - 0.9—1: Excellent, 0.8—0.9: Good, 0.7—0.8: Fair, 0.6—0.7: Poor, 0.5—0.6: Fail

Step 2 - Estimating marginal sample representation As presented in Section 2, we seek to comprehensibly explain sample representation in the dataset $\mathcal{S}$ of geometric shapes on the basis of intuitive design concerns like size, position, and brightness. However, the detector $E_{\mathcal{S}}$ can only assess whether a single combined 6-d test annotation is an outlier. To assess, for example, the diversity of shape sizes in the dataset, independent of position, we turn to techniques of input attribution. Given the detector $E_{\mathcal{S}}$, attribution techniques estimate the contribution of each input label $\hat{Y}_i^j$ to its outlier score S_i. Among proposed methods for input attribution [29], one promising framework is Shapley Additive Explanations (SHAP)[25]. Using principles of cooperative game theory, SHAP estimates *marginal influence* ϕ_i^j (7), which indicates how label $\hat{Y}_i^j$ independently influences the uncertainty score S_i.

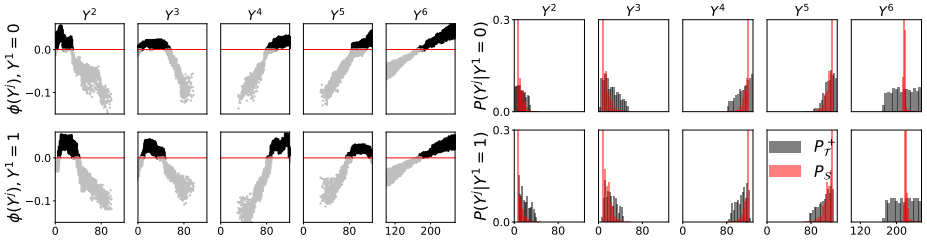
$$S_i = E_{\mathcal{S}}(\hat{Y}_i, F, T) = \phi^0 + \sum_{j=2}^{6} \phi_i^j \tag{7}$$

In satisfying an additive property, SHAP values are also semantically intuitive, with negative, positive, and zero values of ϕ_i^j respectively indicating negative, positive, and neutral influence of label $\hat{Y}_i^j$ on the score S_i. The outlier hypothesis verified earlier implies that outlier (familiar) annotations tend to have a lower (higher) certainty score S_i. Therefore SHAP value $\phi_i^j > 0$, which indicates that the individual label value $\hat{Y}_i^j$ tends to improve S_i, becomes an indicator of that label being represented in $\mathcal{S}$. Through a campaign directed by the test set $\mathcal{T}$, which systematically covers the specified range of scenarios $P_{\mathcal{T}}$, non-negative SHAP values identify sample representation in the dataset $\mathcal{S}$ in terms of each individual label. This can be seen in Figure 7a, where label values with a high incidence of non-negative SHAP values (marked black) are likely to be represented in $\mathcal{S}$. This directly allows estimating the likelihood of test label $Y^j = l,\ Y^j \sim P_{\mathcal{T}}$ being represented in the set $\mathcal{S}$ as the proportion of test labels $\hat{Y}_i^j$, in a sufficiently small interval δ around l, whose SHAP values are non-negative.

$$\begin{gathered} P_{\mathcal{T}}^{+}(Y^j = l \mid Y^1 = k) = \frac{|\{\hat{Y}_i^j : \phi_i^j \geq 0,\ \hat{Y}_i^j \in Y^l\}|}{|\{\hat{Y}_i^j : \phi_i^j \geq 0\}|},\ j = 2...6,\ \hat{Y}_i \in \hat{Y} \\ Y^l = \{l - \delta, l + \delta\}, \qquad \hat{Y} = \{\hat{Y}_i : \hat{Y}_i^1 = k\},\ k \in K \end{gathered} \tag{8}$$

Assessing the explanation By expressing expected diversity $P_{\mathcal{T}}$ in terms of specified design concerns, the two-step process, using a simulated test set, identifies sample representation in each concern using non-negative influence on predictive certainty. From the original broadly spread expectations $P_{\mathcal{T}}$ (Figure 2), the process correctly eliminates a significant amount of outliers in each label dimension, producing $P_{\mathcal{T}}^{+}$ (Figure 7b). $P_{\mathcal{T}}^{+}$ shows label values likely to be observed in the dataset $\mathcal{S}$ and has a roughly similar spread as the actual distribution $P_{\mathcal{S}}$. Also, using a test set with M=10k samples, the process estimates sample representation in a much larger dataset with N=50k samples.

Introduced originally in Section 2 to quantify bias between expected and actual distributions of annotations, the overlap index V is also suitable for measuring similarity between $P^+_{\mathcal{T}}$ and $P_{\mathcal{S}}$. This helps quantify the effectiveness of estimating sample representation using simulation. The visual observation that P^+_T is a better estimate of true sample distribution, compared to the broad range of expectations $P_{\mathcal{T}}$, is confirmed by better a mean overlap score $V^j(P^+_{\mathcal{T}})$ (see Table 1), over all labels and shapes, compared to mean $V^j(P_{\mathcal{T}})$. While this holds true for both classifier instances shown in the table, the detector using F=VGG13 at $T = T^*$, which has the best AUROC score in detecting outliers, produces the closest estimate with a mean overlap score of 0.27. VGG05, with poorer AUROC, has a weaker average overlap score of 0.39. The close correlation between AUROC and V further confirms the plausibility of estimating marginal sample representation using SHAP scores. This shows that, while facing an expensive labeling process, with the right means of parametric simulation, one can conduct a campaign from a low-dimensional space of specified design concerns to estimate sample representation in a given dataset and comprehensibly explain sample selection bias.



(a) Sample-representation from SHAP scores **(b)** Marginal sample representation

Fig. 7: Explaining sample representation using simulation (F=VGG13, $T = T^*$, $S^T = 0.7$)

4 Discussion

Under-representation and outlier detection A good outlier detector $E_{\mathcal{S}}$ of under-represented samples must blur the distinction between simulated and real images while emphasizing the distinction between over and under-represented images. Figure 6b shows both conditions are jointly achievable, with classifiers that have a high test set accuracy, and therefore generalize well, also having better AUROC scores in detecting representation. However, as seen in Figure 8, using regularization measures like batch normalization layers after each convolutional block, while improving test accuracy, reduces AUROC scores for all classifier instances. This is probably because it tends to blur [23] both forms of distinction. The figure also shows that dropout increases the test accuracy without any major effect on AUROC scores, giving no special domain separation advantage in detecting under-representation. Among the classifier configurations investigated here, vanilla VGG, with the strongest correlation between AUROC and test set accuracy, is observed to best addresses both forms of domain distinction.

T	P	Y^1	$V^j(P)$					Mean $V^j(P)$
			j=2	3	4	5	6	
-	$P_{\mathcal{T}}$	0	0.60	0.47	0.59	0.47	0.75	0.57
		1	0.57	0.45	0.57	0.45	0.76	
T^* $S^T = 0.7$	$P^+_{\mathcal{T}}$ F=VGG13	0	0.49	0.14	0.17	0.35	0.55	0.27
		1	-0.19	0.26	0.16	0.17	0.56	
	$P^+_{\mathcal{T}}$ F=VGG05	0	0.47	0.33	0.29	0.44	0.60	0.39
		1	0.31	0.34	0.27	0.25	0.56	
1	$P^+_{\mathcal{T}}$ F=VGG13	0	0.49	0.30	0.40	0.15	0.69	0.36
		1	0.30	0.28	0.21	0.13	0.69	
	$P^+_{\mathcal{T}}$ F=VGG05	0	0.22	0.14	0.54	0.47	0.65	0.43
		1	0.57	0.29	0.56	0.15	0.70	

Table 1: Quantitative bias estimation

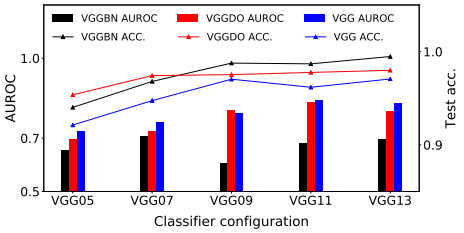


Fig. 8: Effect of regularization on AUROC

The importance of effective simulation It is crucial to note that high test accuracy reflects the combined effect of plausible simulation and good generalization. It is equally essential, therefore, that the simulator produces samples that are plausibly real. Ensuring effective simulation, while supporting a variety of parameters, is undoubtedly a challenge for realistic datasets with richer content. As noted earlier, while this is domain and problem dependent, for images at least, rapid advancements in the quality and range of graphics tools ([5],[9]), potentially makes effective simulation plausible. However, with notable progress in techniques that automate parts of the labeling process [7], it is also important to assess whether labeling is cheaper for the dataset in concern.

Improving estimation of representation Figure 7b shows that while the estimated sample representation $P^+_{\mathcal{T}}$ comes close, it does not overlap perfectly with the true label distribution $P_{\mathcal{S}}$. As quantified in Table 1, even the best detector (F=VGG13 at $T = T^*$) has a mean overlap index of 0.27 indicating relatively close, but only partial, overlap on average. At the individual label level, index values show varying accuracy in support-matching. The representation of pixel brightness $0.5 < V^6(P_{\mathcal{T}}) < 0.8$ is consistently underestimated, while those of bounding box coordinates are better estimated. It is however clear from Table 1 that temperature scaling ($T = T^*$ vs 1) and deeper classifiers (F =VGG13 vs VGG05) improve estimation, indicating that more sophisticated techniques of predictive outlier detection, like methods in [32], can improve estimation.

Balancing detail in specifying expectations The level of detail specified in the expectations $P_{\mathcal{T}}$ plays a key role in deciding the cost and benefit of explaining sample representation. An overly detailed breakdown of design factors involves significant engineering effort, degrades interpretability, and overlooks the remarkable benefits of generalization offered by deep learning. But well-balanced expectations can provide valuable insight into training data. Take an application like self-driving vehicles, where engineers actively seek a certain level of understanding of operational scenarios [15] to ensure safe operation. Such understanding can be exploited to systematically explain, analyze, and manage the data used to train models deployed in the system, thereby improving overall

confidence in its dependability. While balancing details in the specification may not always be easy, one advantage of this method is that it is semi-supervised. Annotations included in the analysis impacts only the simulated test set $\mathcal{T}$ and has no effect on the actual dataset $\mathcal{S}$.

Extension to other domains This method of explanation can conceivably be extended to a problem in another domain if (i) operational scenarios can be reasonably broken down and (ii) model-based parametric simulators that can generate data for this domain are available. For example, this method can use a simulator of vulnerable road user trajectories [16] to examine a sparsely labeled dataset of trajectories (e.g. [30]) and check whether it adequately represents trajectories of risk groups like elder pedestrians, electric bikes, etc.

5 Related work

Sample selection bias Sample selection bias has been addressed in existing literature from the perspective of domain adaptation [19]. Previous methods to mitigate sample selection bias have mainly attempted to modify the training procedures or the model itself to yield classifiers that work well on the test distribution. Methods such as importance re-weighting [35], minimax optimization [24], kernel density estimation [8] and model averaging [10] all fall in this category. While these methods can yield classifiers that are able to generalize, the accuracy can suffer when the two distributions differ greatly in the overlap of their support or in the distribution of their mass. Our immediate goal, on the other hand, does not seek to obtain a classifier that generalizes, but instead we seek to obtain a high level *understanding* of the deficiencies of our training data and where the bias stems from. This goal does not necessarily require a full specification of $P(Y)$, instead we work with the weak proxy of $P_{\mathcal{T}}(Y)$ which attempts to match $P(Y)$ only through the support. However, by eschewing mass-modeling, we gain a few advantages, one of which is the reduced effort in defining expectations. More importantly, since several existing methods for correcting sample selection bias work only if the support of $P_{\mathcal{T}}$ is included in that of $P_{\mathcal{S}}$ and our method of explanation tests precisely for this condition. Overlap indices $V^j(P_{\mathcal{T}}) \leq 0$ guarantees that the support of the biased distribution includes that of the expectations and correction measures like importance re-weighting are applicable. If $0 < V^j(P_{\mathcal{T}}) \leq 1$, expanding the diversity of data collection is unavoidable. Thus seeking to understand and explain the data set can allow for an improved understanding of the validity for methods that directly impacts the generalization performance.

Understanding sample representation Besides clustering approaches [6] and feature projection methods such as t-SNE [26], previous research into providing a high level understanding of the training set has, for example, applied tree-based methods to detect regions of low point density in the input space [13].

High-dimensional explanations in the input space, however, adversely affects interpretation, and ways to extend these methods to yield explanations using an interpretable low-dimensional space of annotations are not immediately clear.

Bias estimation using simulation Closer to our purpose are the methods [28] and [27] which detect inherent biases in a trained model using parametric simulation and Bayesian optimization. While their goal is to find input samples where the model is locally weak, our goal is to ensure that a given dataset meets global expectations defined by a test set. This can verify that a system is dependable for all considered scenarios, like [34], which is a standardized set of tests. However, in reformulating bias detection as outlier detection, our method – unlike the aforementioned methods – trades-off the ability to detect unknown unknowns [21] in favor of a faster, global evaluation of bias. Combining our global and their local approaches may, therefore, help ensure better overall dependability.

Shapley-based outlier detection Previous work using Shapley values for outlier detection, such as [12] and [1], focus mainly on providing interpretable explanations for why a data point is considered to be an outlier. It may also be possible to extend their data-space explanations to the annotation-space, like we do, using parametric simulation. However, pixel-wise reconstruction error has well-known drawbacks in capturing structural aspects of data [22]. It is therefore not immediately clear whether their use of auto-encoder reconstruction error is as good at detecting structural under-representation as our technique of using predictive certainty, which is calculated from the feature space of a classifier.

6 Conclusions

With data playing a crucial role in deciding the behavior of trained models, evaluating whether training and validation sets meets design expectations would be a helpful step towards a better understanding of model properties. To aid this evaluation, we demonstrate a method to specify expectations on and evaluate sample representation in a dataset, in a human interpretable form, in terms of annotations. Using parametric simulation to map test annotations into a test set, the method exposes under-representation by measuring the uncertainty of a classifier, trained on the original dataset, in recognizing test set samples. Techniques of input attribution enable further conversion of predictive uncertainty into a comprehensible low-dimensional estimate of sample representation in the dataset. While refinements in estimation are possible, the core quantitative and qualitative methods shown here are valuable aids in understanding a dataset and, consequently, the properties of a model trained using this data.

References

1. Antwarg, L., Shapira, B., Rokach, L.: Explaining anomalies detected by autoencoders using SHAP. CoRR **abs/1903.02407** (2019), http://arxiv.org/abs/1903.02407
2. Berman, D.S., Buczak, A.L., Chavis, J.S., Corbett, C.L.: A survey of deep learning methods for cyber security. Information **10**(4), 122 (2019). https://doi.org/10.3390/info10040122
3. Birch, J., Rivett, R., Habli, I., Bradshaw, B., Botham, J., Higham, D., Jesty, P., Monkhouse, H., Palin, R.: Safety cases and their role in ISO 26262 functional safety assessment. In: Bitsch, F., Guiochet, J., Kaâniche, M. (eds.) Computer Safety, Reliability, and Security - 32nd International Conference, SAFECOMP 2013, Toulouse, France, September 24-27, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8153, pp. 154–165. Springer (2013). https://doi.org/10.1007/978-3-642-40793-2_15
4. Borg, M., Englund, C., Wnuk, K., Durán, B., Levandowski, C., Gao, S., Tan, Y., Kaijser, H., Lönn, H., Törnqvist, J.: Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. CoRR **abs/1812.05389** (2018), http://arxiv.org/abs/1812.05389
5. Chao, Q., Bi, H., Li, W., Mao, T., Wang, Z., Lin, M.C., Deng, Z.: A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. Comput. Graph. Forum **39**(1), 287–308 (2020). https://doi.org/10.1111/cgf.13803
6. Chen, J., Chang, Y., Hobbs, B., Castaldi, P.J., Cho, M.H., Silverman, E.K., Dy, J.G.: Interpretable clustering via discriminative rectangle mixture model. In: Bonchi, F., Domingo-Ferrer, J., Baeza-Yates, R., Zhou, Z., Wu, X. (eds.) IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain. pp. 823–828. IEEE Computer Society (2016). https://doi.org/10.1109/ICDM.2016.0097
7. Cheng, Q., Zhang, Q., Fu, P., Tu, C., Li, S.: A survey and analysis on automatic image annotation. Pattern Recognit. **79**, 242–259 (2018). https://doi.org/10.1016/j.patcog.2018.02.017
8. Dudík, M., Schapire, R.E., Phillips, S.J.: Correcting sample selection bias in maximum entropy density estimation. In: Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]. pp. 323–330 (2005), http://papers.nips.cc/paper/2929-correcting-sample-selection-bias-in-maximum-entropy-density-estimation
9. Ersotelos, N., Dong, F.: Building highly realistic facial modeling and animation: a survey. The Visual Computer **24**(1), 13–30 (2008). https://doi.org/10.1007/s00371-007-0175-y
10. Fan, W., Davidson, I.: On sample selection bias and its efficient correction via model averaging and unlabeled examples. In: Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA. pp. 320–331. SIAM (2007). https://doi.org/10.1137/1.9781611972771.29, https://doi.org/10.1137/1.9781611972771.29
11. Gardner, A., Kanno, J., Duncan, C.A., Selmic, R.R.: Measuring distance between unordered sets of different sizes. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014. pp. 137–143. IEEE Computer Society (2014). https://doi.org/10.1109/CVPR.2014.25

12. Giurgiu, I., Schumann, A.: Additive explanations for anomalies detected from multivariate temporal data. In: Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E.A., Carmel, D., He, Q., Yu, J.X. (eds.) Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019. pp. 2245–2248. ACM (2019). https://doi.org/10.1145/3357384.3358121
13. Gu, X., Easwaran, A.: Towards safe machine learning for CPS: infer uncertainty from training data. In: Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2019, Montreal, QC, Canada, April 16-18, 2019. pp. 249–258. ACM (2019). https://doi.org/10.1145/3302509.3311038
14. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 1321–1330 (2017), http://proceedings.mlr.press/v70/guo17a.html
15. Gyllenhammar, M., Johansson, R., Warg, F., Chen, D., Heyn, H.M., Sanfridson, M., Söderberg, J., Thorsén, A., Ursing, S.: Towards an Operational Design Domain That Supports the Safety Argumentation of an Automated Driving System. In: 10th European Congress on Embedded Real Time Software and Systems (ERTS 2020). TOULOUSE, France (Jan 2020), https://hal.archives-ouvertes.fr/hal-02456077
16. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Physical Review E **51**(5), 4282–4286 (May 1995). https://doi.org/10.1103/physreve.51.4282, http://dx.doi.org/10.1103/PhysRevE.51.4282
17. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. CoRR **abs/1610.02136** (2016), http://arxiv.org/abs/1610.02136
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
19. Kouw, W.M.: An introduction to domain adaptation and transfer learning. CoRR **abs/1812.11806** (2018), http://arxiv.org/abs/1812.11806
20. Kuutti, S., Bowden, R., Jin, Y., Barber, P., Fallah, S.: A survey of deep learning applications to autonomous vehicle control. CoRR **abs/1912.10773** (2019), http://arxiv.org/abs/1912.10773
21. Lakkaraju, H., Kamar, E., Caruana, R., Horvitz, E.: In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. pp. 2124–2132. AAAI Press (2017), http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14434
22. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. pp. 1558–1566 (2016), http://proceedings.mlr.press/v48/larsen16.html
23. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=Hk6dkJQFx
24. Liu, A., Ziebart, B.D.: Robust classification under sample selection bias. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014,

Montreal, Quebec, Canada. pp. 37–45 (2014), http://papers.nips.cc/paper/5458-robust-classification-under-sample-selection-bias
25. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. pp. 4765–4774 (2017), http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions
26. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008), http://www.jmlr.org/papers/v9/vandermaaten08a.html
27. McDuff, D.J., Cheng, R., Kapoor, A.: Identifying bias in AI using simulation. CoRR **abs/1810.00471** (2018), http://arxiv.org/abs/1810.00471
28. McDuff, D.J., Ma, S., Song, Y., Kapoor, A.: Characterizing bias in classifiers using generative models (2019), http://papers.nips.cc/paper/8780-characterizing-bias-in-classifiers-using-generative-models
29. Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., Mordvintsev, A.: The building blocks of interpretability. Distill (2018), https://distill.pub/2018/building-blocks/
30. Rasouli, A., Kotseruba, I., Kunic, T., Tsotsos, J.K.: PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 6261–6270. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00636, https://doi.org/10.1109/ICCV.2019.00636
31. Seshia, S.A., Sadigh, D.: Towards verified artificial intelligence. CoRR **abs/1606.08514** (2016), http://arxiv.org/abs/1606.08514
32. Shafaei, A., Schmidt, M., Little, J.J.: A less biased evaluation of out-of-distribution sample detectors. In: 30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019. p. 3. BMVA Press (2019), https://bmvc2019.org/wp-content/uploads/papers/0333-paper.pdf
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1409.1556
34. Thorn, E., Kimmel, S.C., Chaka, M.: chap. A Framework for Automated Driving System Testable Cases and Scenarios (Sep 2018), https://rosap.ntl.bts.gov/view/dot/38824, tech Report
35. Tran, V.: Selection Bias Correction in Supervised Learning with Importance Weight. (L'apprentissage des modèles graphiques probabilistes et la correction de biais sélection). Ph.D. thesis, University of Lyon, France (2017), https://tel.archives-ouvertes.fr/tel-01661470
36. Vogelsang, A., Borg, M.: Requirements engineering for machine learning: Perspectives from data scientists. In: 27th IEEE International Requirements Engineering Conference Workshops, RE 2019 Workshops, Jeju Island, Korea (South), September 23-27, 2019. pp. 245–251. IEEE (2019). https://doi.org/10.1109/REW.2019.00050
37. Zadrozny, B.: Learning and evaluating classifiers under sample selection bias. In: Brodley, C.E. (ed.) Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004. ACM International Conference Proceeding Series, vol. 69. ACM (2004). https://doi.org/10.1145/1015330.1015425