Contrastive Identification of Covariate Shift in Image Data

Matthew L. Olson, Thuy-Vy Nguyen, Gaurav Dixit, Neale Ratzlaff, Weng-Keen Wong, and Minsuk Kahng*

Oregon State University

Abstract

Identifying covariate shift is crucial for making machine learning systems robust in the real world and for detecting training data biases that are not reflected in test data. However, detecting covariate shift is challenging, especially when the data consists of high-dimensional images, and when multiple types of localized covariate shift affect different subspaces of the data. Although automated techniques can be used to detect the existence of covariate shift, our goal is to help *human* users characterize the extent of covariate shift in large image datasets with interfaces that seamlessly integrate information obtained from the detection algorithms. In this paper, we design and evaluate a new visual interface that facilitates the comparison of the local distributions of training and test data. We conduct a quantitative user study on multi-attribute facial data to compare two different learned low-dimensional latent representations (pretrained ImageNet CNN vs. density ratio) and two user analytic workflows (nearest-neighbor vs. cluster-to-cluster). Our results indicate that the latent representation of our density ratio model, combined with a nearest-neighbor comparison, is the most effective at helping humans identify covariate shift.

1 Introduction

One of the common problems that plague deployed machine learning (ML) systems is covariate shift [16], which occurs when the input feature distribution $P(\mathbf{X})$ changes between training and testing phases, but the conditional distribution of the response given the features $P(Y|X)$, remains the same. For example, an image recognition system trained during sunny days may not be effective on cloudy days. By not accounting for covariate shift, ML systems can lack robustness when they encounter "unknown unknowns" [8] during deployment and are therefore vulnerable to bias in the training data.

Although automated algorithms can be effective at detecting covariate shift (e.g., Chapters 6-10 in [14]), humans still need to be involved in the process for several reasons: first, it is an important task for people to detect if the data distribution has changed enough to affect a ML system that has been deployed. If the data exhibits bias or shift, they need to know it, so that they can take further actions. Second, it is possible that multiple types of localized covariate shift are occurring in the dataset, with each type affecting a different subspace of the overall feature space. These localized covariate shifts can be challenging for an algorithm to identify and past work has shown that humans can sometimes be better than machines at detecting these problem areas [2]. Third, a human is often ultimately needed to identify the cause of the shift and to fix the problem.

Identifying covariate shift from image data, among the many types of data used in ML, is more challenging because the data is high-dimensional and the original (pixel) feature space is less human-interpretable. Can visualization help human users to identify and characterize how test set images are different from training set images (e.g., face images in the training set have no glasses while the

test set has some) [17]? One possible approach may be to visualize training and test distributions side-by-side (i.e., juxtaposition) using dimensionality reduction methods (e.g., t-SNE) [1] and show each data point as an image thumbnail [4, 20]. However, the scale of modern image datasets makes it difficult because we cannot easily show many images on the projected space [4,6]. Instead of visualizing the distributions of the entire training and test datasets *globally*, we aim to intelligently show only *local* regions of the space, where the locality is informed by the detection algorithm. For example, given a test set image highly ranked by a shift detection algorithm (i.e., deviated from training set distribution), a visualization may show that many of its similar test images (i.e., local neighborhood) share a characteristic (e.g., many faces with sunglasses) while the similar training images do not (e.g., no faces with sunglasses).

In this paper, we design and evaluate a new visual analysis interface for human users to identify covariate shift in image data. Although there exist several visualization work for detecting some types of dataset shift [4, 15, 19, 21], we advance beyond the existing work in two aspects. First, our interface is designed to facilitate *contrastive* analysis between two different distributions for local regions [1, 5], which is a key to covariate shift detection task. We design a novel *side-by-side histogram* view for comparing two sets of images in a selected local region and characterizing shifts. Second, while past work often uses the raw image features in embedding into two-dimensional (2D) space, we integrate the internal latent representation information of detection algorithms into computing similarities between images, which more accurately presents distribution differences.

We address the following two key research questions which we investigate in our 2×2 quantitative user study:

(RQ1) Which learned lower-dimensional latent representation is the most useful for humans to detect covariate shift? Comparing two high-dimensional representations requires some form of dimensionality reduction. We thus compare two lower-dimensional latent representations learned by deep neural networks. The first representation is a commonly used but effective baseline obtained from a pre-trained ImageNet Convolutional Neural Net (CNN) [7]. For the second latent representation, we performed an empirical evaluation and found that the most effective latent representation for a ML algorithm to detect covariate shift is learned through a density ratio estimation (DRE) neural network [12]. We want to evaluate how effective it can be for humans.

(RQ2) Which analytic workflow is the most effective at identifying covariate shift? The side-by-side visualization is designed to work for analyzing the local regions of the feature space. We explore two different user workflows for selecting local regions in discovering covariate shift: (1) a nearest-neighbor approach: a user picks an image that a detection algorithm estimates to be a likely outlier and the user sees similar images; and (2) a cluster-to-cluster approach: a user is presented with a set of clusters and examines each cluster.

2 Related Work

Dataset shift [14] is a broad topic covering many ways test data can be different from training data. Schneider et al. [15] presented a visualization design space for dataset shift and a tool for comparing multi-dimensional feature distributions. In terms of specific types of dataset shift, *concept drift* is a topic that has garnered some attention [19, 21]. Concept drift occurs when the relationship

*e-mail: {olsomatt, nguythu2, dixitg, ratzlafn, wongwe, minsuk.kahng} @oregonstate.edu
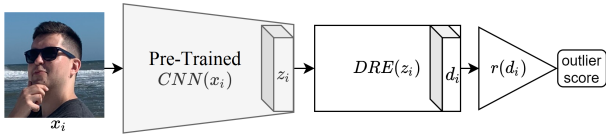
Figure 1: Our ML architecture for computing an outlier score for an image using a pre-trained CNN and our actively trained DRE-based model. We highlight the two vector latent representations of interest, z_i and d_i, which are compared in our user study.

between the response variable and the features (i.e., $P(Y|X)$) changes between training and testing. However, covariate shift addressed in our paper is fundamentally different from concept drift as it detects differences in the training and test feature distributions. Another category of approaches deal with detecting "*unknown unknowns*" (UUs) [8,9], which are data instances incorrectly classified with high confidence because the training data is missing entire subclasses, thereby causing blind spots [2]. Detecting UUs focuses on finding misclassified instances with high confidence. In contrast, our focus on covariate shift ignores classifier confidence and only looks at differences between training and test data distributions for image data. The most closely related work is OoDAnalyzer [4] for detecting out-of-distribution (OoD) images. It first detects OoD instances using a deep ensemble, then the data instances from the original feature space are mapped to a 2D space through a grid-based layout algorithm. Our approach instead aims to enable users to characterize localized covariate shifts affecting a subspace of the features and explores to use different lower dimensional latent spaces extracted from shift detection models rather than the original feature space.

3   LATENT REPRESENTATIONS OF SHIFT DETECTION MODELS

We investigated a variety of latent representations for our task (see supplemental material for more details) and found that the latent representation learned by a *density ratio estimation* (DRE) algorithm performed the best. In this section, we introduce this DRE algorithm, which assigns an *outlier* score to each test instance (lower value means it is unlikely to be drawn from the training set distribution).

Let x_i be the raw input features (i.e., the pixels of the image) of the i-th instance in a dataset. Let $z_i = \text{CNN}(x_i)$ be the learned latent representation from a pre-trained Convolutional Neural Network; for instance, this latent representation is the penultimate layer of the pretrained InceptionNet CNN model [18]. A superscript of tr or te denotes the training or test dataset respectively. For instance, x_i^{tr} indicates the i-th training data instance's features.

Next, in Fig. 1, $d_i = \text{DRE}(z_i)$ is the representation learned when training a DRE-based model, where the density ratio $r(d_i) = P^{\text{tr}}(d_i)/P^{\text{te}}(d_i)$ is the ratio of the training density divided by the test density. We use $r(d_i)$ for determining the outlier score, where the lower the value, more likely the instance will be an outlier. We use the Kullbeck-Liebler Importance Estimation Procedure (KLIEP) as the DRE method because it outperformed other DRE methods in our preliminary investigations. The KLIEP loss is defined as follows:

$$L_{\text{KLIEP}} = \frac{1}{n_{\text{te}}} \sum_{j=1}^{n_{\text{te}}} r\left(d_j^{\text{te}}\right) - \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \ln\left(r\left(d_i^{\text{tr}}\right)\right), \qquad (1)$$

where $r(d) = \log(\exp(Wd+b)+1)$ to ensure non-negativity.

4   VISUAL INTERFACE AND ANALYTIC WORKFLOW DESIGN

This section describes two versions of workflows and associated visual interfaces for covariate shift identification.

Typical ML-only approach. A typical way of detecting outliers or shifts from image data (without visualization) is by having a large list of test images sorted by outlier score, then walking through each image in the list one by one. This method falls short in enabling
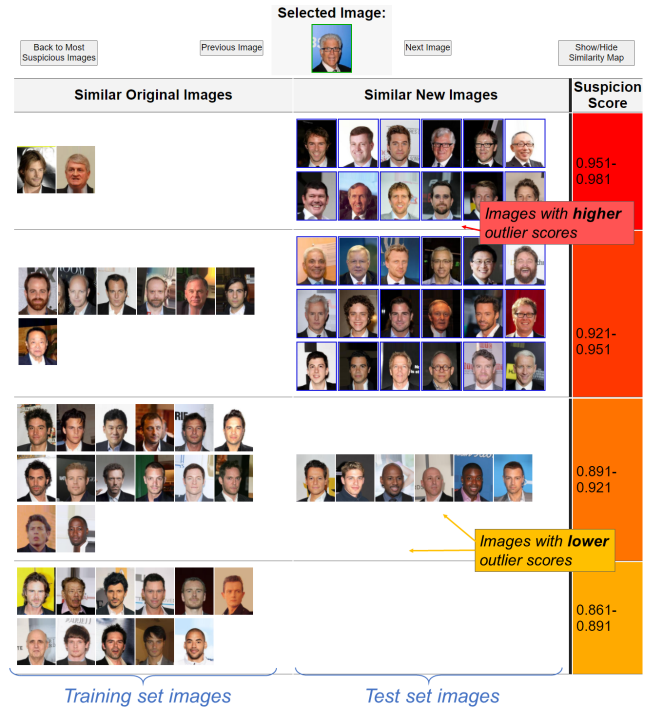

Figure 2: Our *side-by-side histogram* is designed to help users easily compare the distributions of training and test set images (here we used the term "original", "new", and "suspicion score"– instead of "training set", "test set", and "inverse density ratio"– for non-expert users). It shows both training and test set images (on the left and right side, respectively) that are close to a selected image (shown at top). In this example, the similar test set images on the right side have more images with high outlier scores compared to the training set on the left, and only the right side includes face images with smiles and neckties while the left side does not include such images.

users to compare a test set against a training set and finding patterns in test data for the way covariate shift may occur.

Typical VIS-only approach. On the other hand, one approach to visually comparing two distributions is using two 2D projected views side-by-side. However, it does not scale especially if we want to show individual images directly on the projected view.

We combine the ML and VIS approaches by allowing users to select local regions of the data space with the help of shift detection algorithms and visually compare the training and test set distributions for these regions. In selecting local regions, we consider two workflows: (1) *nearest-neighbor* and (2) *cluster-to-cluster*.

4.1   Nearest Neighbor User Workflow

In the first user workflow, a user begins with a list of images sorted by shift scores and examines images one by one, just like the typical ML-only without-visualization approach of analyzing results from outlier detection algorithms. The difference is that the user examines each image with a new visual contrastive interface that shows the neighborhood of the selected image both for training and test sets.

New contrastive visualization for shift identification. Once a user selects an image, they are provided with our new *side-by-side histogram* visualization (shown in Fig. 2). For a selected image, shown at the top, the visualization displays two vertical histograms: one for training set (shown on the left) and the other for test set (on the right). Its vertical bins are computed using a shift detection model's prediction of covariate shift, normalized and sorted over all data. We found sorting to be important for drawing a user's attention
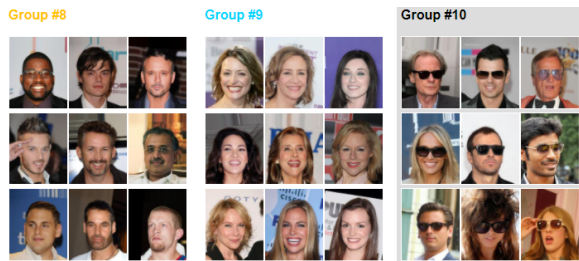
Figure 3: Front page of the interface for the cluster-to-cluster exploration. Images in Cluster #10 include face images with sunglasses. We replaces the term "clusters" with "groups" for non-expert users.
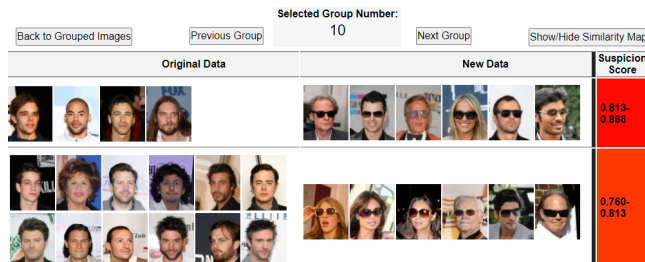

Figure 4: The *side-by-side histogram* with the cluster workflow, showing only the top portion of the interface.

to images most likely to contain a shift. This histogram of images not only shows an outlier score distribution of images, but also displays individual examples of images, motivated by the *unit visualization* technique [13] which represents individual data points within the context of aggregate statistics. For example, in Fig. 2, users can see that test data (on the right side) has more high scored outliers (more images on the top rows) compared to training data (on the left), and faces with glasses, smiles, and neckties appear only on the right side—while none of these attributes occur on the left.

Detailed setup used in user study. For our study, we set the number of histogram bins to be 5 and programatically search for a distance where at least 100 images are chosen for half of the selected images, and set a minimum threshold of number of images to ten for the other half. Distance is calculated by the Frobenius norm in the latent space between the selected image and all others, and neighbors are instances with a small distance. We additionally provide participants with a 2D projection of the latent representation of the test data using UMAP [3] with default parameters, to show a global picture of the feature space for all images. The interface also supports interactions. For example, when users interacting with the histogram view, the 2D projection view is updated to highlight the selected image and the associated test images to help them see their location in a global picture. A participant can navigate to a new histogram by clicking an image in the histogram view or a data point on the projected view.

4.2 Cluster-to-Cluster User Workflow

We design the other workflow, *cluster-to-cluster*, to help users analyze a large number of images without having to select each image one by one. Instead of presenting a sorted list at the beginning, this version of the interface presents a set of clusters with representative images, as shown in Fig. 3. Users can choose one of the clusters to see the corresponding side-by-side histogram for the selected cluster (an example shown in Fig. 4). The visualization looks very similar to that for the nearest neighbor workflow. The main difference being the nearest neighbor shows a selected image at the top while the cluster-to-cluster view simply shows a cluster ID.

Detailed setup used in user study. A participant is first shown with 10 clusters that potentially have more outliers, each with its top nine

outlier images as its representatives (as in Fig. 3). To determine the 10 clusters, we first create 100 clusters from the test data by using an agglomerative clustering algorithm [11] over the latent representation of test set. We then compute the average outlier score for each cluster and select the 10 clusters with highest outlier score.

For the side-by-side histogram for a cluster, a maximum of 50 images in the cluster are shown for the test set, and an equal number of training set images are selected for effective comparison of two distributions, where the training images are the closest images to the cluster's centroid in the latent representation space.

5 User Study

We conducted an online human subject study with a 2×2 design, to answer the two research questions we asked earlier in Section 1:

RQ1. Which learned lower-dimensional latent representation is more effective for humans to detect covariate shift? (i.e., pre-trained ImageNet CNN vs. density ratio) and

RQ2. Which analytic workflow is more effective at identifying covariate shift? (i.e., nearest-neighbor vs. cluster-to-cluster).

5.1 Study Design

Participants. We recruited 60 unique participants using university mailing lists. The average age was 24 with a standard deviation of 6. There were 42 male, 17 female, and 1 gender non-conforming participants. 7 took no Computer Science classes; 17 took 1-3; 15 took 4-6; 13 took 7-12, and 8 took over 13 classes. 11 took at least one class in Artificial Intelligence. Participants were compensated via emailed $10 Amazon gift card upon study completion. We did not reject anyone who applied, as our criteria only included being an adult, color differentiation, and ability to use a computer.

Study Conditions. We used a 2×2 partial within-subject, partial between-subject design, to study the effects of the two variables (i.e., nearest neighbors (NN) vs. cluster-to-cluster (CL) workflows; ImageNet (IM) vs. Density Ratio (DR)). We randomly assign participants to the condition of using the ImageNet features (IM) or those learned by our Density Ratio (DR) covariate shift model (between-subjects). Both conditions used the same underlying CNN, density ratio model, and outlier scores. Then, each participant performed two shift identification tasks, one with the NN workflow and the other with the CL workflow (within-subjects). For example, a participant performed the first shift identification task (e.g., glasses, smile, and necktie) with the NN workflow that uses the ImageNet features (i.e., NN-IM), and then performed the second task (e.g., hats and facial hair) with the CL workflow that uses the same ImageNet features (i.e., CL-IM). The condition orders were counter-balanced.

Dataset. We used a subset of images from the CelebA faces dataset [10], as non-experts can understand changing attributes on a face and would not need guidance on this part of the task. We selected 6 attributes from the 40 potential attributes: eyeglasses, smiling, wearing necktie, wearing a hat, having a beard, and having a mustache. We separated these attributes into two sets of shifts: (1) glasses, smiles, and neckties; (2) hats and facial hair (beards/mustache combined). We selected 5,000 images as a training set, 9,000 as an unshifted test set, and 1,000 as a test set containing a shift.

Study Procedure. Our study was conducted completely online, where participants performed on their own once provided website URL and login information. The study begins with a video tutorial with an example that uses a toy dataset of flowers. Participants were asked attention check questions to ensure their understanding. Participants are then directed to perform the tasks with the interface for a minimum of 10 minutes and a maximum of 20 minutes for each shift set before submitting a Google form answer sheet of what they believe to be the sources of covariate shift (up to 5 responses). The interfaces used in the study are shown in the supplemental material.

5.2 Study Data Collection and Analysis

We coded participant responses to compare the number of participants who find each specific shift (e.g., eyeglasses) between conditions. For all statistical analysis we use a one-tailed 2×2 Fisher's exact test. The contingency table of the Fisher's test is comprised of "found specific shift" vs. "did not find it" and condition "a" vs. "b". For example, we perform a test on the CL workflow with DR features vs. the same workflow with the IM features on the eyeglasses attribute.

6 Results

Our results indicate that the *nearest neighbor* workflow with *density ratio* latent representation (NN-DR) is generally the best combination for identifying covariate shift. Table 1 shows the count of the number of participants who find a specific shift. Table 2 shows the p-values from Fisher's test statistic for the comparison of each condition to another. We find NN-DR is significantly better than CL-DR as well as CL-IM at finding smiles. Other than for detecting eyeglasses, no method performs better than NN-DR; even on eyeglasses, the improvement by the cluster view is not statistically significant.

When comparing the conditions by representation, the density ratio representation is always equivalent to or better at identifying covariate shift than the original ImageNet space. Some shifts lent themselves to a specific workflow rather than a latent space. Finding neckties and facial hair is difficult for cluster workflow users, with only two and three participants finding it, respectively. However, using the nearest neighbor workflow is nearly statistically significant for neckties and is significant for finding facial hair (one-tailed Fisher's Exact Test, $p < 0.05$). We believe the primary benefit to users is the ability to focus on a single image to compare and contrast both the train and the test sets. In the one instance where NN-DR is not the best (for detecting eyeglasses), the CL-DR is not statistically

Condition	Glasses	Smile	Necktie	Hats	Facial Hair
NN-IM	4	6	3	14	6
CL-IM	4	7	1	14	1
NN-DR	7	**13**	**4**	15	**6**
CL-DR	**11**	7	1	**15**	2

Table 1: The number of participants discovering each shift for all conditions, higher is better. The best performing method for each condition are highlighted in bold.

Comparison	Glasses	Smile	Necktie	Hats	Facial Hair
DR>IM	**0.01**	0.06	0.50	0.25	0.50
NN-DR>NN-IM	0.22	**0.01**	0.50	0.50	0.64
CL-DR>CL-IM	**0.01**	0.64	0.76	0.50	0.50
NN>CL	0.22	0.15	0.07	0.75	**0.01**
NN-IM>CL-IM	0.66	0.77	0.30	0.76	**0.04**
NN-DR>CL-DR	0.97	**0.03**	0.16	1.00	0.11

Table 2: A table of p-values for a comparison between a pair of conditions from the user study. Bolded numbers are statistically significant at $p < 0.05$ (one-tailed Fisher's Exact Test).

	Shift Set 1			Shift Set 2		
	P ($\uparrow$)	R ($\uparrow$)	FP ($\downarrow$)	P ($\uparrow$)	R ($\uparrow$)	FP ($\downarrow$)
NN-IM	0.42	0.29	1.40	0.64	0.67	1.00
CL-IM	0.35	0.27	1.80	0.49	0.50	1.40
NN-DR	**0.85**	**0.53**	**0.53**	**0.70**	**0.70**	**0.93**
CL-DR	0.48	0.42	1.47	0.57	0.57	1.40

Table 3: Average precision (P), recall (R) and false positive rate (FP) by condition. The arrows indicate whether higher or lower is better.
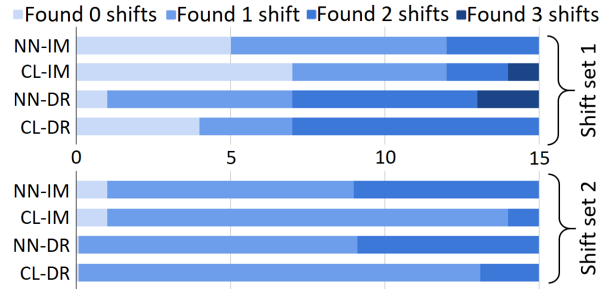

Figure 5: Number of participants on how many shifts they found among three (for shift set #1) or two different shifts (for shift set #2). For example, for the NN-DR condition for shift set #1, 14 out of 15 participants found at least one shift and two found all the three shifts.

significantly superior (one-tailed Fisher's Exact Test, $p = 0.132$).

The cluster workflow is also more detrimental to participants finding shifts that do not exist. Table 3 shows the average precision, recall, and false positive rate for each condition. NN-DR is the highest in precision and recall and the lowest in false positive rate. In terms of user workflows, the precision for the nearest neighbor workflow is at least 0.37 higher for shift set 1 and 0.13 higher for shift set 2 when compared to the cluster workflow. These results suggest that by having a focal image, participants are better able to compare and contrast the training and test sets.

Lastly, participants in the nearest neighbor workflow are consistently better at finding more shifts (e.g., finding one, two, and all three shifts from shift set 1). Breakdowns are shown in Fig. 5.

7 Discussion

Our results clearly point toward two outcomes: the importance of selecting an appropriate latent representation and a user's need to have a focal image for which to compare against a group. Our cluster workflow generally performed worse than we originally expected, and as Participant #20 said, "I believe having a selected image helps to understand the process [of finding the shift] better". It is clear that not having a central selected image makes the shift detection task much harder, as a user must compare all test images in a given cluster to each other and also to all the images in the training set.

The nearest neighbor workflow was not always able to outperform the cluster workflow, such as for detecting eyeglasses. We speculate if participants had more time, and thus more examples of outliers than the limited few in the top 100 images, they may have been better able to identify that shift. The density ratio space cluster view (CL-DR) created a side-by-side histogram of all face images wearing sunglasses in the test set, which no other condition generated.

A final result to note is that at least one user from all conditions found each shift. This finding validates our experimental setup was not biased in favor of or against a particular condition.

8 Conclusion and Future Work

This work is one of the first to investigate analytic workflows for detecting covariate shifts in image data and to investigate the effect of latent representations on how well human users detect them. Our results indicate that using a nearest neighbor approach combined with a density ratio latent representation enabled participants to accurately discover and characterize different types of localized covariate shift.

While our results are promising, we want to note the limitations of this work. The main caveat is naturally the limitations of the data itself. We used a relatively small dataset that exhibits covariate shift. We leave it for future work to examine cases in extremely large data settings, or in settings where no covariate shift has occurred.

REFERENCES

[1] D. L. Arendt, N. Nur, Z. Huang, G. Fair, and W. Dou. Parallel embeddings: a visualization technique for contrasting learned representations. In *Proceedings of the 25th International Conference on Intelligent User Interfaces (IUI)*, pp. 259–274, 2020.

[2] J. Attenberg, P. Ipeirotis, and F. Provost. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *J. Data and Information Quality*, 6(1), Mar. 2015.

[3] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 2019.

[4] C. Chen, J. Yuan, Y. Lu, Y. Liu, H. Su, S. Yuan, and S. Liu. OoD-Analyzer: Interactive analysis of out-of-distribution samples. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3335–3349, 2021.

[5] M. Gleicher. Considerations for visualizing comparison. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):413–423, 2017.

[6] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2674–2693, 2019.

[7] M. Huh, P. Agrawal, and A. A. Efros. What makes ImageNet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

[8] H. Lakkaraju, E. Kamar, R. Caruana, and E. Horvitz. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, p. 2124–2132, 2017.

[9] A. Liu, S. Guerra, I. Fung, G. Matute, E. Kamar, and W. Lasecki. Towards hybrid human-AI workflows for unknown unknown detection. In *Proceedings of The Web Conference 2020 (WWW)*, pp. 2432–2442, 2020.

[10] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[11] D. Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

[12] H. Nam and M. Sugiyama. Direct density ratio estimation with convolutional neural networks with application in outlier detection. *IEICE Transactions on Information and Systems*, E98.D(5):1073–1079, 2015.

[13] D. Park, S. M. Drucker, R. Fernandez, and N. Elmqvist. Atom: A grammar for unit visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3032–3043, 2018.

[14] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[15] B. Schneider, D. A. Keim, and M. El-Assady. Datashiftexplorer: Visualizing and comparing change in multidimensional data for supervised learning. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: IVAPP*, pp. 141–148, 2020.

[16] 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.

[17] T. Spinner, U. Schlegel, H. Schäfer, and M. El-Assady. explAIner: A visual analytics framework for interactive and explainable machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1064–1074, 2020.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[19] X. Wang, W. Chen, J. Xia, Z. Chen, D. Xu, X. Wu, M. Xu, and T. Schreck. ConceptExplorer: Visual analysis of concept drifts in multi-source time-series data. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 1–11. IEEE, 2020.

[20] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2020.

[21] W. Yang, Z. Li, M. Liu, Y. Lu, K. Cao, R. Maciejewski, and S. Liu. Diagnosing concept drift with visual analytics. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 12–23. IEEE, 2020.