Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift

Stephan Rabanser [1 2] **Stephan Günnemann** [1] **Zachary C. Lipton** [2]

Abstract

We might hope that when faced with unexpected inputs, well-designed software systems would fire off warnings. Machine learning (ML) systems, however, which depend strongly on properties of their inputs (e.g. the i.i.d. assumption), tend to fail silently. This paper explores the problem of building ML systems that fail loudly, investigating methods for detecting dataset shift and identifying exemplars that most typify the shift. We focus on several datasets and various perturbations to both covariates and label distributions with varying magnitudes and fractions of data affected. Interestingly, we show that while classifier-based methods designed to explicitly discriminate between source and target domains perform well in high-data settings, they perform poorly in low-data settings. Moreover, across the dataset shifts that we explore, a two-sample-testing-based approach using pre-trained classifiers for dimensionality reduction performs best.

1. Introduction

Software systems employing deep neural networks are now applied widely in industry, powering the vision systems in social networks and self-driving cars, providing assistance to radiologists, underpinning recommendation engines used by online retailers, enabling the best-performing commercial speech recognition software, and automating translation between languages. In each of these systems, predictive models are integrated into conventional human-interacting software systems, which leverage their predictions to drive consequential real-world decisions.

The reliable functioning of software depends crucially on tests. Many classic software bugs can be caught when software is compiled, e.g., that a function receives input of the wrong type, while other problems are detected only at runtime, triggering warnings or exceptions. In the worst case, if the errors are never caught, software may behave incorrectly without alerting anyone to the problem.

Unfortunately, software systems based on machine learning are notoriously hard to test and maintain (Sculley et al., 2014). Despite their power, modern machine learning models are brittle. Seemingly subtle changes in the data distribution can destroy the performance of otherwise state-of-the-art classifiers, a phenomenon exemplified by adversarial examples (Szegedy et al., 2013; Zügner et al., 2018). When decisions are made under uncertainty, even shifts in the label distribution can significantly compromise accuracy (Zhang et al., 2013; Lipton et al., 2018). Unfortunately, in practice, ML pipelines rarely inspect incoming data for signs of distribution shift. Moreover, best practices detecting shift in high-dimensional real-world data have not yet been established[1]. The first indications that something has gone awry might come when customers complain.

In this paper, we investigate methods for detecting and characterizing distribution shift, with the hope of removing a critical stumbling block obstructing the safe and responsible deployment of machine learning in high-stakes applications. Faced with distribution shift, our goals are three-fold: (i) detect when distribution shift occurs from as few examples as possible; (ii) quantify the amount of shift; and (iii) characterize the shift in distribution, e.g. by identifying those samples from the test set that appear over-represented in the target data. This paper focuses principally on goal (i) and, to a lesser extent, (iii).

We investigate shift detection through the lens of statistical two-sample testing. We wish to test the equivalence of the *source* distribution (from which training data is sampled) and *target* distribution (from which real-world data is sampled). For simple univariate distributions, such hypothesis testing is a mature science. However, best practices for two sample tests with high-dimensional (e.g., image) data remain an open question. While off-the-shelf methods for kernel-based multivariate two-sample tests are appealing, they scale badly with dataset size and their statistical power is known to decay badly with high ambient dimension (Ramdas et al., 2015).

[1]Technical University of Munich, Germany [2]Carnegie Mellon University, Pittsburgh, PA. Correspondence to: Stephan Rabanser <rabanser@cs.tum.edu>.

[1]TensorFlow's data validation tools compare only summary statistics of source vs target data: https://tensorflow.org/tfx/data_validation/get_started#checking_data_skew_and_drift

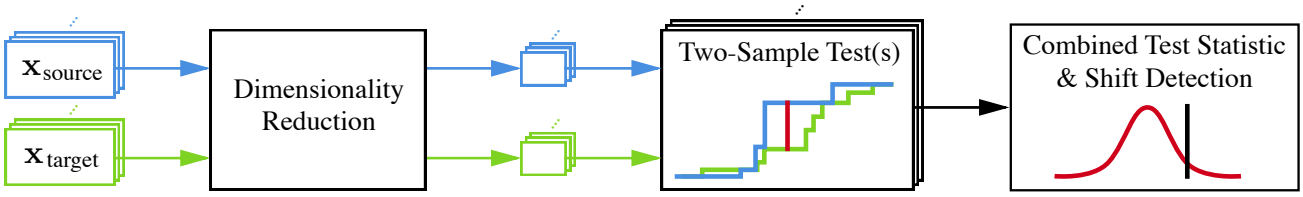


Figure 1. Our pipeline for detecting dataset shift: source and target data is fed through a dimensionality reduction process and subsequently analyzed through statistical testing. We consider various choices for how to represent the data and how to perform two-sample tests.

One natural approach (to ML practitioners) might be to train a classifier to distinguish between source and target examples. Given class-balanced holdout samples, we can pronounce the data shifted if our classifier can recognizes the domain with significantly greater than 50% accuracy. Analyzing the simple case where one wishes to test the means of two Gaussians, Ramdas et al. (2016) recently showed that the power of a classification-based two-sample test using Fisher's Linear Discriminant Analysis classifier achieves minimax rate-optimal performance. However, the performance of classifier-based approaches has not been characterized (either theoretically or empirically) for the complex high-dimensional data distributions on which modern machine learning is routinely deployed. Providing this empirical analysis is the key contribution of this paper. Throughout, to avoid confusion, we denote any source-vs-target classifier a *domain classifier* and refer to classifiers trained (on source data) for the original classification task as a *label classifier*.

One benefit of the domain-classifier approach is that the *domain classifier* reduces dimensionality to a single dimension, learned precisely for the purpose of discriminating between source and target data. However, learning such a classifier from scratch may require large amounts of training data. Adding to the problem, the domain-classifier approach requires partitioning our (scarce) target data using, e.g., using half for training and leaving the remainder for two-sample testing. Alternatively we also explore the *black box shift detection (BBSD)* approach due to Lipton et al. (2018), which addresses shift detection under the label shift assumption. They show that if one possesses an off-the-shelf label classifier $f(x)$ with an invertible confusion matrix (verifiable on training data), then detecting that the source distribution p differs from the target distribution q requires only detecting that $p(f(x)) \neq q(f(x))$. This insight enables efficient shift detection, using a pre-trained (label) classifier for dimensionality reduction. Building on these ideas of combining black-box dimensionality reduction with subsequent two-sample testing, we explore a range of dimensionality reduction techniques and compare them under a wide variety of shifts (Figure 1 illustrates our general framework). We show (empirically) that BBSD works surprisingly well under a broad set of shifts, even when the label shift assumption is not met.

2. Related work

Given just one example from the test data, our problem simplifies to *anomaly detection*, surveyed thoroughly by Chandola et al. (2009) and Markou & Singh (2003). Popular approaches to anomaly detection include density estimation (Breunig et al., 2000), margin-based approaches such as one-class SVMs (Schölkopf et al., 2000), and the tree-based isolation forest method due to (Liu et al., 2008). Recently, GANs have been explored for this task (Schlegl et al., 2017). Given simple streams of data arriving in a time-dependent fashion where the signal is piece-wise stationary with abrupt changes, this is the classic time series problem of change point detection, surveyed comprehensively by Truong et al. (2018). An extensive literature addresses dataset shift in the context of domain adaptation. Owing to the impossibility of correcting for shift absent assumptions (Ben-David et al., 2010), these papers often assume either covariate shift $q(x, y) = q(x)p(y|x)$ (Shimodaira, 2000; Sugiyama et al., 2008; Gretton et al., 2009) or label shift $q(x, y) = q(y)p(x|y)$ (Saerens et al., 2002; Chan & Ng, 2005; Storkey, 2009; Zhang et al., 2013; Lipton et al., 2018), where p and q denote the target and source distributions, respectively. Schölkopf et al. (2012) provides a unifying view of these shifts, associating assumed invariances with the corresponding causal assumptions.

Several recent papers have proposed outlier detection mechanisms dubbing the task *out-of-distribution (OOD) sample detection*. Hendrycks & Gimpel (2017) proposes to simply threshold the maximum softmax entry of a neural network classifier which already seems to contain a relevant signal. Liang et al. (2018) and Lee et al. (2018) extend this idea by either adding temperature scaling and adversarial-like perturbations on the input or by explicitly adapting the loss to aid OOD detection. Choi & Jang (2018) and Shalev et al. (2018) employ model ensembling to further improve detection reliability. Alemi et al. (2018) motivate use of the variational information bottleneck. Hendrycks et al. (2019) expose the model to OOD samples, exploring heuristics for discriminating between in-distribution and out-of-distribution samples. Shafaei et al. (2018) survey and compare numerous OOD detection techniques.

3. Shift Detection Techniques

Given labeled data $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n) \sim p$ and unlabeled data $\boldsymbol{x}'_1, ..., \boldsymbol{x}'_m \sim q$, our task is to determine whether $p(\boldsymbol{x})$ equals $q(\boldsymbol{x})$. Formally, $H_0 : p(\boldsymbol{x}) = q(\boldsymbol{x})$ vs $H_A : p(\boldsymbol{x}) \neq q(\boldsymbol{x})$. Chiefly, we explore the following design considerations: (i) what **representation** to run the test on; (ii) which **two-sample test** to run; (iii) when the representation is multidimensional; whether to run **multivariate or multiple univariate two-sample tests**; and (iv) **how to combine** their results.

3.1. Dimensionality Reduction

We now introduce the multiple dimensionality reduction (DR) techniques that we compare vis-a-vis their effectiveness in shift detection (in concert with two-sample testing). Note that absent assumptions on the data, these mappings, which reduce the data dimensionality from D to K (with $K << D$), are in general surjective, with many different inputs mapping to the same output. Thus, it is trivially to construct pathological cases where the distribution of inputs shifts while the distribution of low-dimensional latent representations remains fixed, yielding false negatives. However, we speculate that in a non-adversarial setting, such shifts may be exceedingly unlikely. Thus our approach is (i) empirically motivated, and (ii) not put forth as a defense against worst-case adversarial attacks.

No Reduction (*NoRed*) To justify the use of any DR technique, our default baseline is to run tests on the original raw features.

Principal Components Analysis Principal components analysis (PCA) is a standard tool that finds an optimal orthogonal transformation matrix $\boldsymbol{R} \in \mathbb{R}^{D \times K}$ such that points are linearly uncorrelated after transformation. This transformation is learned in such a way that the first principal component accounts for as much of the variability in the dataset as possible, and that each succeeding principal component captures as much of the remaining variance as possible subject to the constraint that it be orthogonal to the preceding components. Formally, we wish to learn $\boldsymbol{R}$ given $\boldsymbol{X}$ under the mentioned constraints such that $\hat{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{R}$ yields a more compact data representation.

Sparse Random Projection (*SRP*) Since computing the optimal transformation might be expensive in high dimensions, random projections are a popular DR technique which trade a controlled amount of accuracy for faster processing times. Specifically, we make use of sparse random projections, a more memory- and computationally-efficient modification of standard Gaussian random projections. Formally, we generate a random projection matrix $\boldsymbol{R}$, using it to reduce the dimensionality of a given data matrix $\boldsymbol{X}$, such that $\hat{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{R}$. The elements of $\boldsymbol{R}$ are generated using the following rule set

$$R_{ij} = \begin{cases} +\sqrt{\frac{v}{K}} & \text{with probability } \frac{1}{2v} \\ 0 & \text{with probability } 1 - \frac{1}{v} \\ -\sqrt{\frac{v}{K}} & \text{with probability } \frac{1}{2v} \end{cases} \quad (1)$$

where $v = \frac{1}{\sqrt{D}}$ (Achlioptas, 2003; Li et al., 2006).

Autoencoders (*TAE* and *UAE*) We compare the above-mentioned linear models to non-linear reduced-dimension representations learned by both *trained* (TAE) and *untrained* autoencoders (UAE), Formally, an autoencoder consists of an encoder function $\phi : \mathcal{X} \to \mathcal{L}$ and a decoder function $\psi : \mathcal{L} \to \mathcal{X}$ where the latent space $\mathcal{L}$ has lower dimensionality than the input space $\mathcal{X}$. As part of the training process, both the encoding function ϕ and the decoding function ψ are learned jointly to reduce the reconstruction loss: $\phi, \psi = \arg\min_{\phi,\psi} \|\boldsymbol{X} - (\psi \circ \phi)\boldsymbol{X}\|^2$.

Label Classifiers (*BBSDs* and *BBSDh*) Motivated by recent results achieved by black box shift detection (BBSD) (Lipton et al., 2018), we also propose to use the outputs of a (deep network) *label classifier* trained on source data as our dimensionality-reduced representation. We explore variants using either the softmax outputs (BBSDs) or the hard-thresholded predictions (BBSDh) for subsequent two-sample testing. Since both variants provide differently sized output (with BBSDs providing an entire softmax vector and BBSDh providing a one-dimensional class prediction), different statistical tests are carried out on these representations.

Domain Classifier (*Classif*) Here, we attempt to detecting shift by explicitly training a *domain classifier* to discriminate between data from source and target domains. To this end, we partition both the source data and target data into two halves, using the first to train a domain classifier to distinguish source (class 0) from target (class 1) data. We then apply this model to the second half conducting a significance test to determine if the classifier's performance is different from random chance.

3.2. Statistical Hypothesis Testing

The DR techniques each yield a representation, either uni- or multi-dimensional, and either continuous or discrete depending on the method. The next step is to choose a suitable statistical hypothesis test for each of these representations.

Multivariate kernel two-sample tests For all multi-dimensional representations, we evaluate the *Maximum Mean Discrepancy (MMD)*, a popular kernel-based technique for multivariate two-sample testing. MMD allows us to distinguish between two probability distributions p, q

based on the mean embeddings μ_p, μ_q of the distributions in a reproducing kernel Hilbert space (RKHS) $\mathcal{F}$, formally

$$\text{MMD}(\mathcal{F}, p, q) = ||\mu_p - \mu_q||^2_{\mathcal{F}}. \tag{2}$$

Given samples from both distributions, we can calculate an unbiased estimate of the squared MMD statistic as follows

$$\begin{aligned}\text{MMD}^2 = &\frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} \kappa(x_i, x_j) \\ &+ \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \kappa(y_i, y_j) \\ &- \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \kappa(x_i, y_j)\end{aligned} \tag{3}$$

where we use a squared exponential kernel $\kappa(x, x') = e^{-\frac{1}{2}\|x - x'\|^2}$. A p-value can then be obtained by carrying out a permutation test on the resulting kernel matrix.

Multiple Univariate Testing: Kolmogorov-Smirnov (KS) Test + Bonferroni correction As a simple baseline alternative to MMD, we consider the approach consisting of testing each of the K dimensions separately (instead testing over all dimensions jointly). Here, for continuous data, we adopt the *Kolmogorov-Smirnov (KS) test*, a non-parametric test whose statistic is calculated by computing the largest difference S of the cumulative density functions (CDFs) over all values z as follows:

$$S = \sup_z |F_s(z) - F_t(z)| \tag{4}$$

where F_s and F_t are the empirical CDFs of the source and target data, respectively. Under the null hypothesis, S follows the Kolmogorov distribution.

Since we carry out a KS test on each of the K components, we must subsequently combine the p-values from each test, raising the issue of multiple hypothesis testing. Since we cannot make strong assumptions about the (in)dependence among the tests, we rely on a conservative aggregation method, notably the Bonferroni correction (Bland & Altman, 1995), which rejects the null hypothesis if the minimum p-value among all tests is less than α/K (where α is the significance level of the test). While several less conservative aggregations methods have been proposed (Simes, 1986; Zaykin et al., 2002; Loughin, 2004; Heard & Rubin-Delanchy, 2018; Vovk & Wang, 2018), they typically require assumptions on the dependencies among the tests. Moreover, even using the conservative test, in our experiments, the univariate approach generally outperformed kernel two-sample testing given identical representations (Section 5).

Categorical Testing: Chi-Squared Test For the hard-thresholded label classifier (BBSDh), we employ Pearson's chi-squared test, a parametric tests designed to evaluate whether the frequency distribution of certain events observed in a sample is consistent with a particular theoretical distribution. Specifically, we use a test of independence between the class distributions (expressed in a contingency table) of source and target data. The testing problem can be formalized as follows: given a contingency table with 2 rows (one for absolute source and one for absolute target class frequencies) and K columns containing observed counts O_{ij}, the expected frequency under the independence hypothesis for a particular cell is $E_{ij} = N_{\text{sum}} p_{i\cdot} p_{\cdot j}$ with N_{sum} being the sum of all cells in the table, $p_{i\cdot} = \frac{O_{i\cdot}}{N} = \sum_{j=1}^{c} \frac{O_{ij}}{N}$ being the fraction of row totals, and $p_{\cdot j} = \frac{O_{\cdot j}}{N} = \sum_{i=1}^{r} \frac{O_{ij}}{N}$ being the fraction of column totals. The relevant test statistic X^2 can be computed as

$$X^2 = \sum_{i=1}^{2} \sum_{j=1}^{K} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{5}$$

which, under the null hypothesis, follows a chi-squared distribution with $k - 1$ degrees of freedom: $X^2 \sim \chi^2_{k-1}$.

Binomial Testing For the domain classifier, we simply compare its accuracy (acc) on held-out data to random chance via a binomial test. Formally, we set up a testing problem H_0 : acc $= 0.5$ vs H_A : acc $\neq 0.5$. Under the null hypothesis, the accuracy of the classifier follows a binomial distribution: acc $\sim$ Bin$(N_{\text{samp}}, 0.5)$, where N_{samp} corresponds to the number of held-out samples used.

3.3. Obtaining most anomalous samples

As our detection framework does not detect outliers but rather aims at capturing top-level shift dynamics, it is not possible for us to decide whether any given sample is in-distribution or out-of-distribution. However, we can still provide an indication of what typical samples from the shifted distribution look like by harnessing domain assignments from the domain classifier. Specifically, we can identify the exemplars which the classifier was most confident in assigning to the target domain. Hence, whenever the binomial test signals a statistically significant accuracy deviation from chance, we can use use the domain classifier to obtain the most anomalous samples and present them to the user.

In contrast to the domain classifier, the other shift detectors we propose do not base their shift detection potential on explicitly deciding which domain a single sample belongs to, instead comparing entire distributions against each other. While we did explore initial ideas on identifying samples which if removed would lead to a large increase in the overall p-value, the results we obtained were unremarkable.

4. Experiments

Our experiments were carried out on the MNIST ($N_{\text{tr}} = 50000$; $N_{\text{val}} = 10000$; $N_{\text{te}} = 10000$; $D = 28 \times 28 \times 1$; $C = 10$ classes) (LeCun et al., 1998) and CIFAR-10 ($N_{\text{tr}} = 40000$; $N_{\text{val}} = 10000$; $N_{\text{te}} = 10000$; $D = 32 \times 32 \times 3$; $C = 10$ classes) (Krizhevsky & Hinton, 2009) image datasets. For the autoencoder (UAE & TAE) experiments, we employ a convolutional architecture with 3 convolutional layers and 1 fully-connected layer. For both the label and the domain classifier we use a ResNet-18 (He et al., 2016). We train all networks (TAE, BBSDs, BBSDh, Classif) using stochastic gradient descent with momentum batches of 128 examples, decaying the learning rate with $1/\sqrt{t}$) over 200 epochs with early stopping.

For PCA, SRP, UAE, and TAE, we reduce dimensionality to $K = 32$ latent dimensions, which for PCA explains roughly 80% of the variance in the CIFAR-10 dataset. The label classifier BBSDs reduces dimensionality to the number of classes C. Both the hard label classifier BBSDh and the domain classifier Classif reduce dimensionality to a one-dimensional class prediction, where BBSDh predicts label assignments and Classif predicts domain assignments.

To challenge our detection methods, we simulate a variety of shifts, affecting both the covariates and the label proportions. For all shifts, we evaluate the various methods' abilities to detect shift at a significance level of $\alpha = 0.05$. We also include the no-shift case to check against false positives. We randomly split all of the data into training, validation, and test sets according to the indicated proportions N_{tr}, N_{val}, and N_{te} and then apply a particular shift to the test set only. In order to qualitatively quantify the robustness of our findings, shift detection performance is averaged over a total of 5 random splits, which ensures that we apply the same type of shift to different subsets of the data. The selected training data used to fit the DR methods is kept constant across experiments with only the splits between validation and test changing across the random runs. Note that DR methods are learned using training data, while shift detection is being performed on dimensionality-reduced representations of the validation and the test set. We evaluate the models with various amounts of samples from the test set $s \in \{10, 20, 50, 100, 200, 500, 1000, 10000\}$. Because of the unfavorable dependence of kernel methods on the dataset size we, run these methods only up until 1000 target samples have been acquired.

For each shift type (as appropriate) we explored three levels of shift intensity (e.g. the magnitude of added noise) and various percentages of affected data $\delta \in \{0.1, 0.5, 1.0\}$. Specifically, we explore the following types of shifts:

(i) **Adversarial shift (*adv_shift*)**: We turn a given percentage δ of test samples into adversarial examples via the fast gradient sign method (Goodfellow et al., 2014);

(ii) **Knock-out shift (*ko_shift*)**: We remove a fraction δ of data points from class $c = 0$, creating class imbalance (Lipton et al., 2018);

(iii) **Gaussian noise shift (*gn_shift*)**: We corrupt covariates of a fraction δ of test set samples by Gaussian noise centered in the datapoint with standard deviation $\sigma \in \{1, 10, 100\}$ (denoted *small_gn_shift*, *medium_gn_shift*, and *large_gn_shift*);

(iv) **Image shift (*img_shift*)**: More natural shifts to images, modifying a fraction δ of images with combinations of random amounts of rotations $\{10, 40, 90\}$, (x, y)-axis-translation percentages $\{0.05, 0.2, 0.4\}$, as well as zoom-in percentages $\{0.1, 0.2, 0.4\}$ (denoted *small_img_shift*, *medium_img_shift*, and *large_img_shift*);

(v) **Image shift + knock-out shift (*medium_img_shift+ ko_shift*)**: Test sets which are affected by both label distribution and covariate shifts—we apply a fixed medium image shift with $\delta_1 = 0.5$ and a variable knock-out shift δ;

(vi) **Only-zero shift + image shift (*only_zero_shift+ medium_img_shift*)**: Here, we only include images from class $c = 0$ in combination with a variable medium image shift affecting only a fraction δ of the data;

(vii) **Original splits**: As a sanity check, we evaluate our detectors on the original source/target splits provided by the creators of MNIST, CIFAR-10, Fashion MNIST, and SVHN datasets (typically assumed to be i.i.d.);

(viii) **Domain adaptation datasets**: Data from the domain adaptation task transferring from MNIST (source) to USPS (target) ($N_{\text{tr}} = 1000$; $N_{\text{val}} = 1000$; $N_{\text{te}} = 1000$; $D = 16 \times 16 \times 1$; $C = 10$ classes) (Long et al., 2013).

5. Discussion

We now discuss the salient findings from our empirical investigation:

Univariate vs multivariate tests We first evaluate whether we can detect shifts more easily using multiple univariate tests and aggregating their results via the Bonferroni correction or by using multivariate kernel tests. We were surprised to find that across DR methods, aggregated univariate tests outperformed multivariate tests (see tables 1, 2, 3, and 4).

Table 1. Detection accuracy of different dimensionality reduction techniques across all simulated shifts on MNIST and CIFAR-10. **Green bold** entries indicate the best DR method at a given sample size, *red italic* the worst. Underlined entries indicate accuracy values larger than 0.5.

Test	DR	Number of samples from test							
		10	20	50	100	200	500	1,000	10,000
Univ. tests	NoRed	0.18	0.25	0.36	0.39	0.45	0.49	0.57	0.70
	PCA	*0.13*	0.22	*0.25*	*0.30*	*0.35*	*0.41*	*0.46*	*0.58*
	SRP	0.18	*0.21*	0.27	0.32	0.37	0.46	0.53	0.61
	UAE	0.20	0.25	0.33	0.42	0.48	0.54	0.65	**0.74**
	TAE	0.20	0.26	0.37	0.45	0.44	0.53	0.58	0.67
	BBSDs	**0.29**	**0.38**	**0.43**	**0.49**	**0.55**	**0.61**	**0.66**	0.72
χ^2	BBSDh	0.14	0.18	0.23	0.26	0.32	0.41	0.47	0.48
Bin	Classif	0.04	0.09	0.10	0.10	0.28	0.38	0.47	0.66
Multiv. tests	*NoRed*	*0.00*	*0.00*	*0.00*	*0.04*	0.15	0.15	*0.18*	–
	PCA	0.01	0.05	0.09	0.11	0.15	0.22	0.28	–
	SRP	*0.00*	*0.00*	0.03	0.08	*0.13*	*0.14*	0.19	–
	UAE	**0.19**	**0.26**	**0.36**	0.36	0.42	0.49	**0.59**	–
	TAE	**0.19**	0.22	**0.36**	**0.44**	**0.46**	**0.50**	0.58	–
	BBSDs	0.16	0.19	0.23	0.31	0.30	0.43	0.48	–

Dimensionality reduction methods For each testing method and experimental setting, we evaluate which DR technique is best suited to shift detection. In the multiple-univariate-testing case (and thus overall), BBSDs was the best-performing DR method. In the multivariate-testing case, the autoencoders (UAE and TAE) performed best. In both cases, these methods consistently outperformed others across sample sizes. The domain classifier performs badly in the low-sample regime (≤ 100 samples), but quickly catches up as more samples are obtained. Noticeably, the multivariate test performs poorly in the no reduction case, especially on CIFAR-10, perhaps owing to the high dimensionality of the dataset. Table 1 summarizes these results.

Shift types Table 2 lists shift detection accuracy values for each distinct shift as an increasing amount of samples is obtained from the target domain. Specifically, we see that *large_gn_shift*, *medium_gn_shift*, *large_img_shift*, *medium_img_shift+ko_shift*, and *only_zero_shift+medium_img_shift* are easily detectable even with few samples, while *small_gn_shift*, *medium_gn_shift*, *adv_shift*, and *ko_shift* are hard to detect even with many samples. With a few exceptions, the best DR technique (BBDSs for multiple univariate tests, UAE & TAE for multivariate tests) is significantly faster and more accurate at detecting shift than the average of all dimensionality reduction methods.

Shift intensity Based on the results in Table 3, we can conclude that the small shifts (*small_gn_shift*, *small_img_shift*, and *ko_shift*) are harder to detect than medium shifts (*medium_gn_shift*, *medium_img_shift*, and *adv_shift*) which in turn are harder to detect than large shifts (*large_gn_shift*, *large_img_shift*, *medium_img_shift+ko_shift*, and *only_zero_shift+medium_img_shift*). Specifically, we see that large shifts can on average already be detected with better than chance accuracy at only 10 samples in the multiple univariate testing setting. Medium and small shifts require orders of magnitude more samples in order to achieve similar accuracy.

Test sample size As we can clearly see from the results in tables 1, 2, 3, and 4, the more samples we obtain from the target domain, the better we can detect shifts.

Identifying exemplars of shift While full individual results are presented in the supplementary material, we briefly discuss two exemplary results in detail: adversarial shift on MNIST (see Figure 2) and medium image shift on CIFAR-10 (see Figure 3). Sub-figures (a)-(c) show the p-value evolution of the different DR methods, while sub-figures (d) and (e) show the *most different* and *most similar* exemplars returned by the domain classifier.

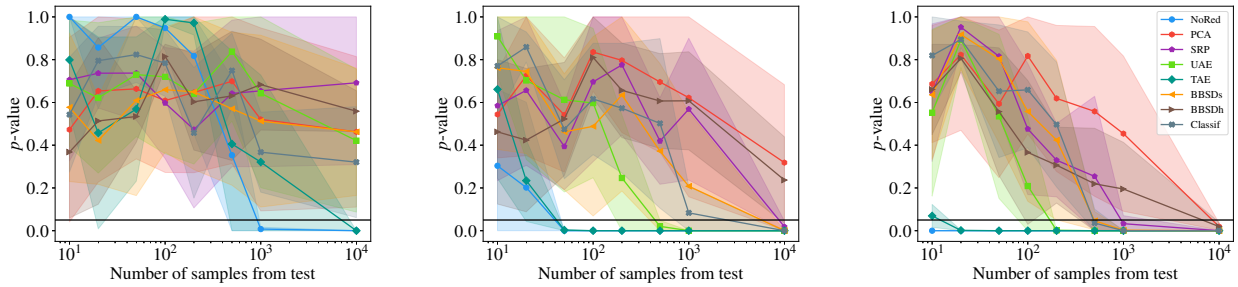
In the adversarial shift case on MNIST, we can clearly see the influence of the percentage of adversarial examples in the test set. Both the no-reduction baseline and the autoencoders appear well-suited to detecting this shift, requiring only 10% adversarial samples for detection. Since MNIST images contain a large amount of zero-values (depicted as black), not only the domain classifier but also the human eye can easily detect anomalous input images and distinguish them from images with perfectly black backgrounds.

Looking at the medium image shift example on CIFAR-10, we see that BBSDs is in the lead here, already detecting a shift with only 10% perturbed samples, while other methods required a larger percentage of shifted images in order to detect a shift within the given sample sizes. We can see that the top-different-samples clearly show the effects of rotation, (x, y)-translations, and zooms. In contrast, top-similar-samples show well-centered images (some of them even showing picture frames which clearly indicate that the picture is centered).

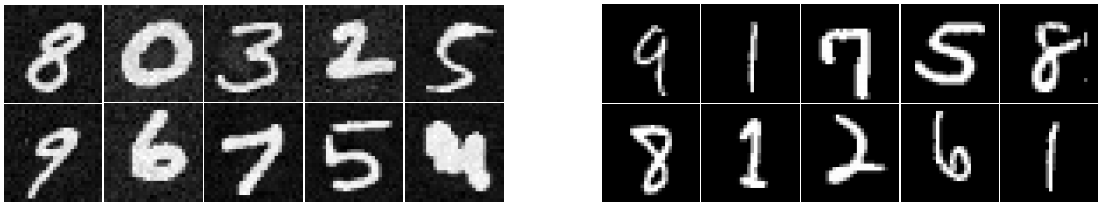
Original splits According to our tests, the original splits from the MNIST dataset appear to exhibit a dataset shift. After inspecting the most anomalous samples returned by the difference classifier, we observed that many of these samples depicted the digit 6. A mean-difference plot between sixes from the training set and sixes from the test set revealed that the training instances are rotated slightly to the right, while the test samples are drawn more open and centered. To back up this claim even further, we also carried out a two-sample KS test between the two sets of sixes in the input space and found that the two sets can conclusively be regarded as different with a p-value of $2.7 \cdot 10^{-10}$, significantly undercutting the respective Bonferroni threshold of $6.3 \cdot 10^{-5}$. While this particular shift does not look

Table 2. Detection accuracy of different shifts on MNIST and CIFAR-10. The first entry in each cell shows to the accuracy of the best DR technique (univariate: BBSDs, multivariate: average of UAE and TAE), while the value in parentheses second entry shows the accuracy across all dimensionality reduction techniques. Underlined entries indicate accuracy values larger than 0.5.

Test type	Simulated shift type	Number of samples from test							
		10	20	50	100	200	500	1,000	10,000
Univariate tests	small_gn_shift	0.00 (0.07)	0.03 (0.07)	0.07 (0.07)	0.10 (0.12)	0.10 (0.12)	0.10 (0.16)	0.10 (0.19)	0.10 (0.23)
	medium_gn_shift	0.00 (0.09)	0.03 (0.09)	0.10 (0.13)	0.10 (0.16)	0.14 (0.19)	0.24 (0.25)	0.24 (0.31)	0.38 (0.41)
	large_gn_shift	0.45 (0.34)	0.52 (0.37)	0.59 (0.51)	0.72 (0.53)	0.83 (0.64)	0.86 (0.72)	0.97 (0.79)	1.00 (0.87)
	small_img_shift	0.14 (0.05)	0.21 (0.08)	0.31 (0.13)	0.45 (0.19)	0.59 (0.26)	0.59 (0.30)	0.69 (0.38)	0.97 (0.58)
	medium_img_shift	0.34 (0.13)	0.55 (0.25)	0.66 (0.31)	0.79 (0.36)	0.83 (0.41)	0.90 (0.54)	0.93 (0.60)	1.00 (0.83)
	large_img_shift	0.48 (0.23)	0.66 (0.32)	0.72 (0.40)	0.83 (0.49)	0.83 (0.62)	0.93 (0.73)	1.00(0.82)	1.00 (0.91)
	adv_shift	0.07 (0.08)	0.10 (0.11)	0.10 (0.14)	0.10 (0.13)	0.17 (0.17)	0.21 (0.20)	0.34 (0.31)	0.45 (0.38)
	ko_shift	0.00 (0.04)	0.00 (0.04)	0.00 (0.04)	0.07 (0.08)	0.10 (0.12)	0.34 (0.22)	0.48 (0.34)	0.72 (0.60)
	medium_img_shift+ko_shift	0.45 (0.14)	0.66 (0.18)	0.79 (0.31)	1.00 (0.46)	1.00 (0.55)	1.00 (0.67)	1.00 (0.76)	1.00 (0.97)
	only_zero_shift+medium_img_shift	1.00 (0.52)	1.00 (0.79)	1.00 (0.90)	1.00 (0.91)	1.00 (0.98)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
Multivariate kernel tests	small_gn_shift	0.02 (0.01)	0.02 (0.02)	0.05 (0.06)	0.05 (0.08)	0.08 (0.08)	0.10 (0.09)	0.12 (0.09)	–
	medium_gn_shift	0.03 (0.01)	0.08 (0.04)	0.17 (0.08)	0.19 (0.12)	0.29 (0.13)	0.32 (0.18)	0.39 (0.21)	–
	large_gn_shift	0.51 (0.20)	0.53 (0.22)	0.71 (0.36)	0.75 (0.37)	0.81 (0.44)	0.88 (0.51)	0.97 (0.58)	–
	small_img_shift	0.12 (0.04)	0.15 (0.06)	0.20 (0.08)	0.29 (0.12)	0.39 (0.16)	0.41 (0.21)	0.51 (0.28)	–
	medium_img_shift	0.29 (0.12)	0.31 (0.12)	0.36 (0.15)	0.39 (0.21)	0.44 (0.31)	0.54 (0.36)	0.66 (0.45)	–
	large_img_shift	0.29 (0.12)	0.32 (0.16)	0.36 (0.25)	0.46 (0.30)	0.59 (0.42)	0.75 (0.47)	0.81 (0.53)	–
	adv_shift	0.03 (0.03)	0.08 (0.03)	0.20 (0.08)	0.22 (0.08)	0.22 (0.09)	0.25 (0.11)	0.32 (0.14)	–
	ko_shift	0.05 (0.06)	0.10 (0.06)	0.17 (0.08)	0.19 (0.12)	0.20 (0.12)	0.24 (0.16)	0.34 (0.23)	–
	medium_img_shift+ko_shift	0.17 (0.08)	0.22 (0.10)	0.39 (0.16)	0.51 (0.22)	0.54 (0.24)	0.58 (0.37)	0.78 (0.50)	–
	only_zero_shift+medium_img_shift	0.36 (0.25)	0.75 (0.40)	0.86 (0.50)	1.00 (0.63)	1.00 (0.75)	1.00 (0.76)	1.00 (0.82)	–



(a) 10% adversarial data in test set. (b) 50% adversarial data in test set. (c) 100% adversarial data in test set.



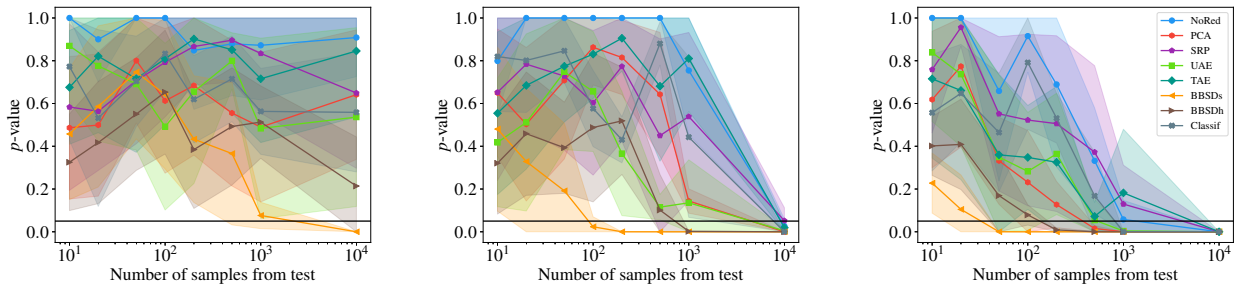
(d) Top different samples. (e) Top similar samples.

Figure 2. Shift detection results for adversarial shift on MNIST.

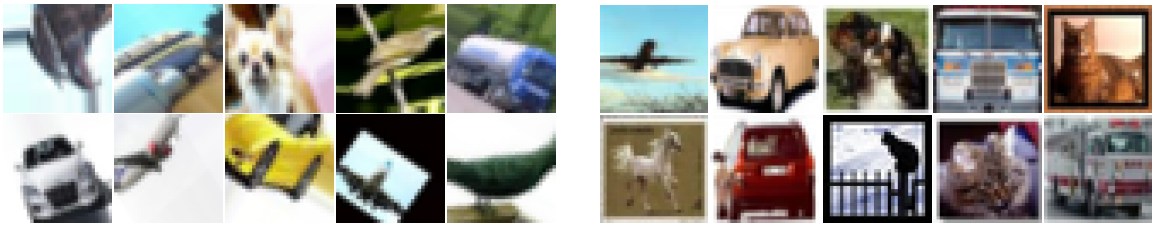
practically significant to the human eye, this result however still shows that the original MNIST split is not truly i.i.d. This result raises the significant question of how to quantify the malignancy of a shift, i.e. when a shift is harmful to a given machine learning model. After all, trained models generalize to the original MNIST test set with ease.

6. Conclusions

In this paper we put forth a comprehensive empirical investigation, examining the ways in which dimensionality reduction and two-sample testing might be combined to produce a practical pipeline for detecting distribution shift in real-life machine learning systems. Our results yielded the surprising insights that the (i) black box shift detection with soft predictions works well across a wide variety of scenarios, even when the underlying assumption of invariant class conditional distributions does not hold; and (ii) that given a suitable low-dimensional representation for shift detection, aggregated univariate tests performed separately on each latent dimension outperform multivariate two-sample tests, even when aggregated conservatively. Moreover, we produced the surprising observation that the MNIST dataset, despite ostensibly representing a random split, exhibits a



(a) 10% perturbed data in test set. (b) 50% perturbed data in test set. (c) 100% perturbed data in test set.



(d) Top different samples. (e) Top similar samples.

Figure 3. Shift detection results for medium image shift on CIFAR-10.

Table 3. Detection accuracy for small, medium, and large simulated shifts on MNIST and CIFAR-10 using univariate tests + Bonferroni correction on BBSDs. Reported accuracy values are results of the best DR technique (univariate: BBSDs, multivariate: average of UAE and TAE). Underlined entries indicate accuracy values larger than 0.5.

Test	Intensity	Number of samples from test							
		10	20	50	100	200	500	1,000	10,000
Univ.	Small	0.06	0.11	0.13	0.15	0.23	0.3	0.38	0.52
	Medium	0.17	0.29	0.37	0.42	0.47	0.56	0.58	0.69
	Large	0.59	0.7	0.76	0.88	0.91	0.94	0.99	1.00
Multiv.	Small	0.07	0.08	0.18	0.20	0.21	0.24	0.32	–
	Medium	0.17	0.20	0.27	0.28	0.37	0.44	0.53	–
	Large	0.34	0.44	0.59	0.67	0.73	0.80	0.89	–

Table 4. Detection accuracy for low (10%), medium (50%), and high (100%) percentages of perturbed target samples across all shifts on MNIST and CIFAR-10. Reported accuracy values are results of the best dimensionality reduction technique (univariate: BBSDs, multivariate: average of UAE and TAE). Underlined entries indicate accuracy values larger than 0.5.

Test	Percentage	Number of samples from test							
		10	20	50	100	200	500	1,000	10,000
Univ.	10%	0.19	0.20	0.25	0.34	0.36	0.41	0.48	0.56
	50%	0.28	0.42	0.49	0.56	0.60	0.61	0.70	0.80
	100%	0.41	0.54	0.55	0.60	0.70	0.81	0.82	0.82
Multiv.	10%	0.14	0.15	0.24	0.27	0.29	0.31	0.41	–
	50%	0.20	0.22	0.39	0.41	0.45	0.54	0.61	–
	100%	0.26	0.38	0.46	0.54	0.60	0.68	0.76	–

significant (although, perhaps not worrisome) distribution shift. Our work suggests several open questions that might offer promising paths for future work:

(i) *Under which conditions are shifts meaningful?* Since even the smallest of shifts will be detectable by the methods we proposed given a sufficient number of test samples, we are left with the question of when the detected shift warrants action. Recall that the detected MNIST shift on the original split did not have any observable impact on the label classifier's performance in our tests. Deciding when practitioners should be alarmed and what actions they should take remains an open problem.

(ii) *How do we detect shifts in online data?* Since data often arrives in a continuous stream, adapting our detection scheme to deal with online data would be an important addition. By doing so, we would need to account for and exploit the high degree of correlation between adjacent time steps, known as multiple-hypothesis-testing over time. Recently, Howard et al. (2018) provided some interesting insights on how to design nonparametric, time-evolving confidence intervals, which are correct at every single time-step.

(iii) *How does the proposed detection scheme work in other domains?* Since we have mostly explored a standard image classification setting for our experiments, it would be interesting to see how our method performs on problem classes in other machine learning domains, such as natural language processing or graphs.

Acknowledgements

We thank the Center for Machine Learning and Health, a joint venture of Carnegie Mellon University, UPMC, and the University of Pittsburgh for supporting our collaboration with Abridge AI to develop robust models for machine learning in healthcare. We are also grateful to Salesforce Research for a faculty award supporting our research on robust deep learning under distribution shift.

References

Achlioptas, D. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66, 2003.

Alemi, A. A., Fischer, I., and Dillon, J. V. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

Ben-David, S., Lu, T., Luu, T., and Pál, D. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Bland, J. M. and Altman, D. G. Multiple significance tests: the bonferroni method. *BMJ*, 1995.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof: identifying density-based local outliers. In *ACM sigmod record*, 2000.

Chan, Y. S. and Ng, H. T. Word sense disambiguation with distribution estimation. In *International Joint Conference on Artificial intelligence (IJCAI)*, 2005.

Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 2009.

Choi, H. and Jang, E. Generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2014.

Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., and Schölkopf, B. Covariate shift by kernel mean matching. *Journal of Machine Learning Research (JMLR)*, 2009.

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

Heard, N. A. and Rubin-Delanchy, P. Choosing between methods of combining-values. *Biometrika*, 2018.

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

Hendrycks, D., Mazeika, M., and Dietterich, T. G. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations (ICLR)*, 2019.

Howard, S. R., Ramdas, A., McAuliffe, J., and Sekhon, J. Uniform, nonparametric, non-asymptotic confidence sequences. *arXiv preprint arXiv:1810.08240*, 2018.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 1998.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations (ICLR)*, 2018.

Li, P., Hastie, T. J., and Church, K. W. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006.

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

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

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In *International Conference on Data Mining (ICDM)*, 2008.

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. Transfer feature learning with joint distribution adaptation. In *International conference on computer vision (ICCV)*, 2013.

Loughin, T. M. A systematic comparison of methods for combining p-values from independent tests. *Computational statistics & data analysis*, 2004.

Markou, M. and Singh, S. Novelty detection: a reviewpart 1: statistical approaches. *Signal processing*, 2003.

Ramdas, A., Reddi, S. J., Póczos, B., Singh, A., and Wasserman, L. A. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2015.

Ramdas, A., Singh, A., and Wasserman, L. Classification accuracy as a proxy for two sample testing. *arXiv preprint arXiv:1602.02210*, 2016.

Saerens, M., Latinne, P., and Decaestecker, C. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 2002.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, 2017.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. Support vector method for novelty detection. In *Advances in neural information processing systems (NIPS)*, 2000.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. In *International Conference on Machine Learning (ICML)*, 2012.

Sculley, D., Phillips, T., Ebner, D., Chaudhary, V., and Young, M. Machine learning: The high-interest credit card of technical debt. In *SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop)*, 2014.

Shafaei, A., Schmidt, M., and Little, J. J. Does your model know the digit 6 is not a cat? a less biased evaluation of" outlier" detectors. *arXiv preprint arXiv:1809.04729*, 2018.

Shalev, G., Adi, Y., and Keshet, J. Out-of-distribution detection using multiple semantic label representations. In *Advances in Neural Information Processing Systems*, 2018.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 2000.

Simes, R. J. An improved bonferroni procedure for multiple tests of significance. *Biometrika*, 1986.

Storkey, A. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 2009.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems (NIPS)*, 2008.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Truong, C., Oudre, L., and Vayatis, N. A review of change point detection methods. *arXiv preprint arXiv:1801.00718*, 2018.

Vovk, V. and Wang, R. Combining p-values via averaging. *arXiv preprint arXiv:1212.4966*, 2018.

Zaykin, D. V., Zhivotovsky, L. A., Westfall, P. H., and Weir, B. S. Truncated product method for combining p-values. *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society*, 2002.

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

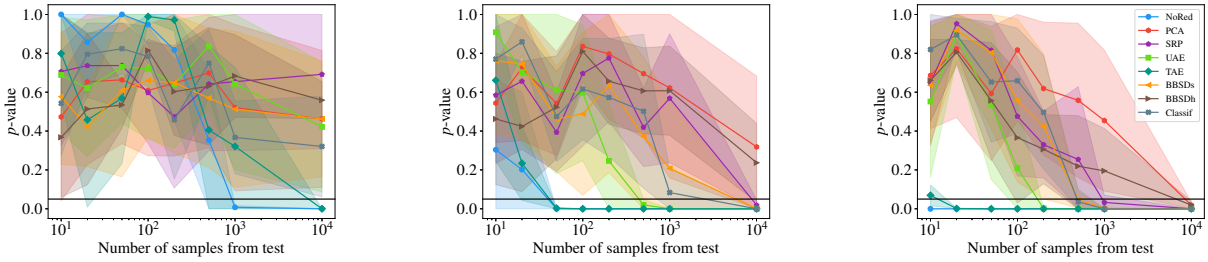
Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial attacks on neural networks for graph data. In *International Conference on Knowledge Discovery & Data Mining (KDD)*, 2018.

A. Detailed Shift Detection Results

Our complete shift detection results in which we evaluate different kinds of target shifts on MNIST and CIFAR-10 using the proposed methods are documented below. In addition to our artificially generated shifts, we also evaluated our testing procedure on the original splits provided by MNIST, Fashion MNIST, CIFAR-10, and SVHN.

A.1. Artificially Generated Shifts

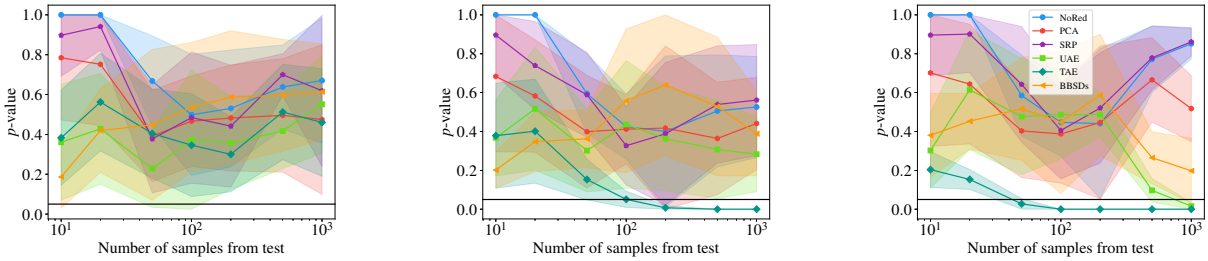
A.1.1. MNIST



(a) 10% adversarial samples. (b) 50% adversarial samples. (c) 100% adversarial samples.

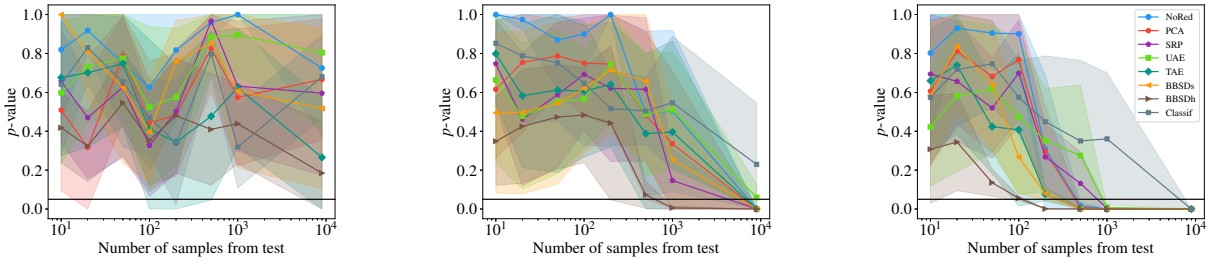
(d) Top different samples. (e) Top similar samples.

Figure 4. MNIST adversarial shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% adversarial samples. (b) 50% adversarial samples. (c) 100% adversarial samples.

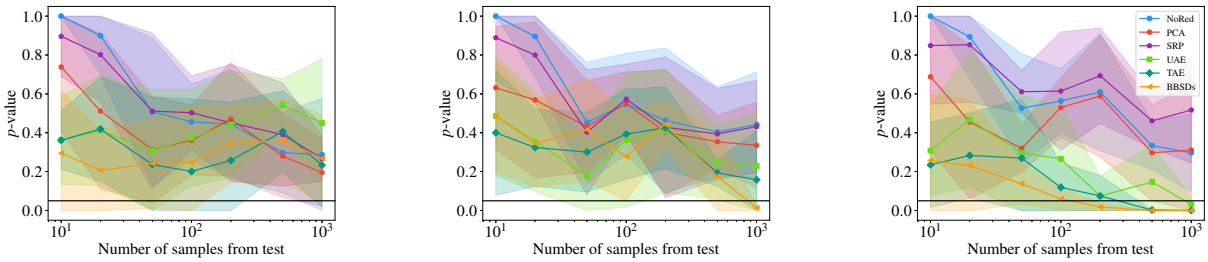
Figure 5. MNIST adversarial shift, multivariate two-sample tests.



(a) Knock out 10% of class 0. (b) Knock out 50% of class 0. (c) Knock out 100% of class 0.

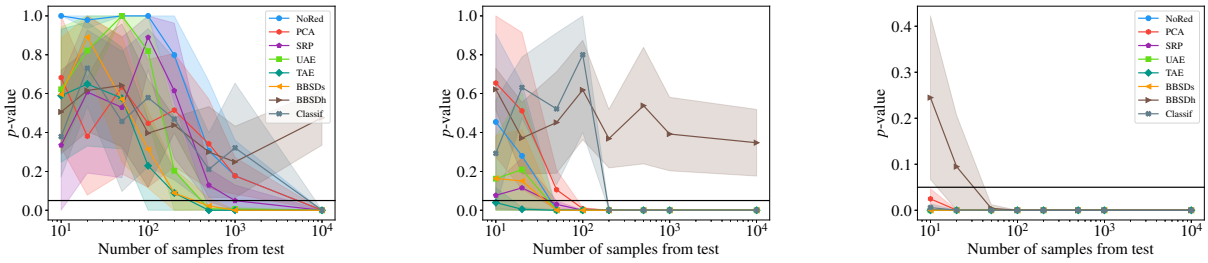
(d) Top different samples. (e) Top similar samples.

Figure 6. MNIST knock-out shift, univariate two-sample tests + Bonferroni aggregation.



(a) Knock out 10% of class 0. (b) Knock out 50% of class 0. (c) Knock out 100% of class 0.

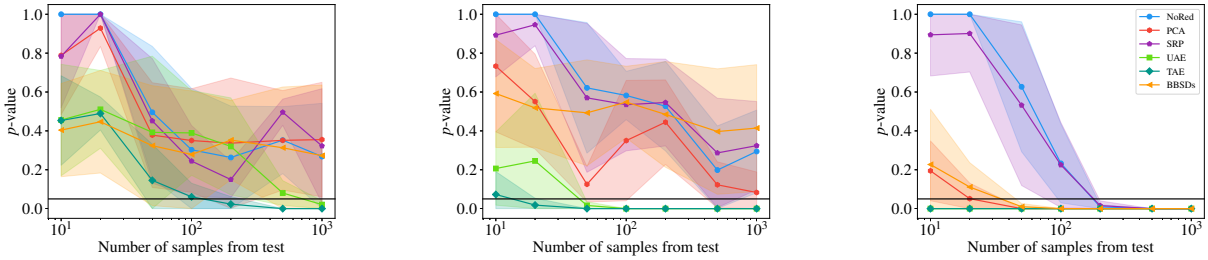
Figure 7. MNIST knock-out shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

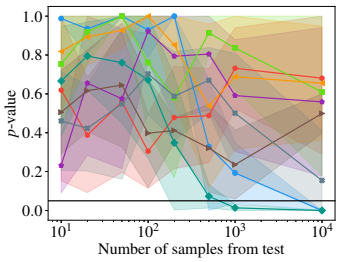
(d) Top different samples. (e) Top similar samples.

Figure 8. MNIST large Gaussian noise shift, univariate two-sample tests + Bonferroni aggregration.

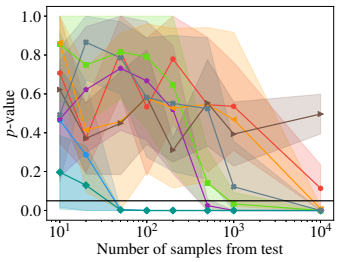


(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

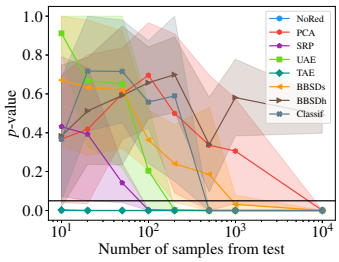
Figure 9. MNIST large Gaussian noise shift, multivariate two-sample tests.



(a) 10% perturbed samples.



(b) 50% perturbed samples.

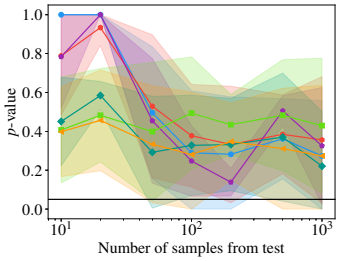


(c) 100% perturbed samples.

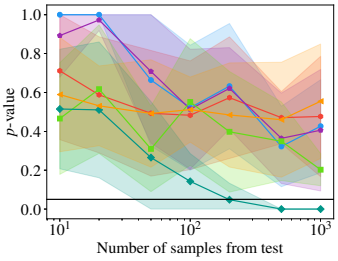
(d) Top different samples.

(e) Top similar samples.

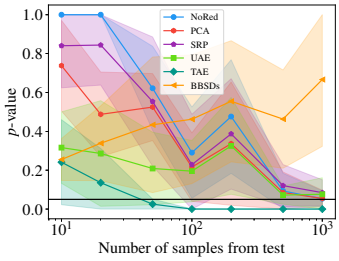
Figure 10. MNIST medium Gaussian noise shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% perturbed samples.

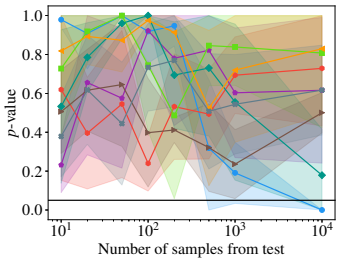


(b) 50% perturbed samples.

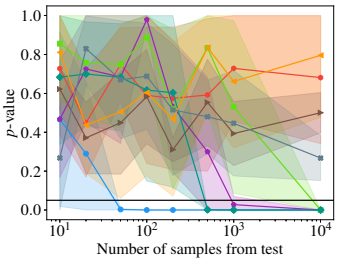


(c) 100% perturbed samples.

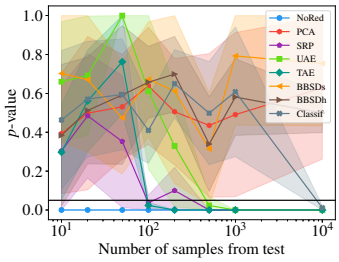
Figure 11. MNIST medium Gaussian noise shift, multivariate two-sample tests.



(a) 10% perturbed samples.



(b) 50% perturbed samples.

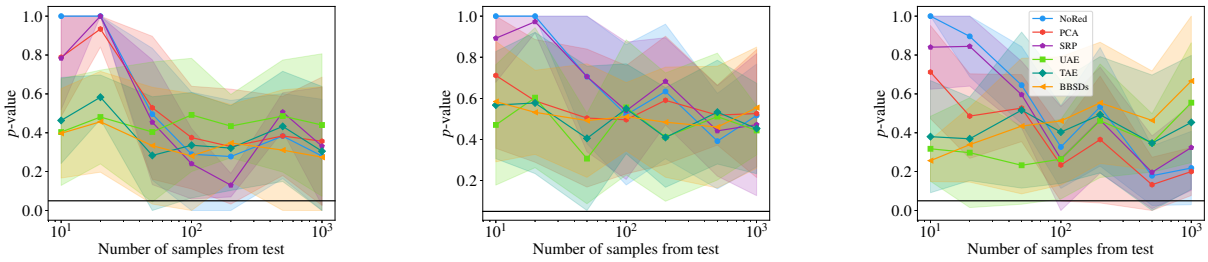


(c) 100% perturbed samples.

(d) Top different samples.

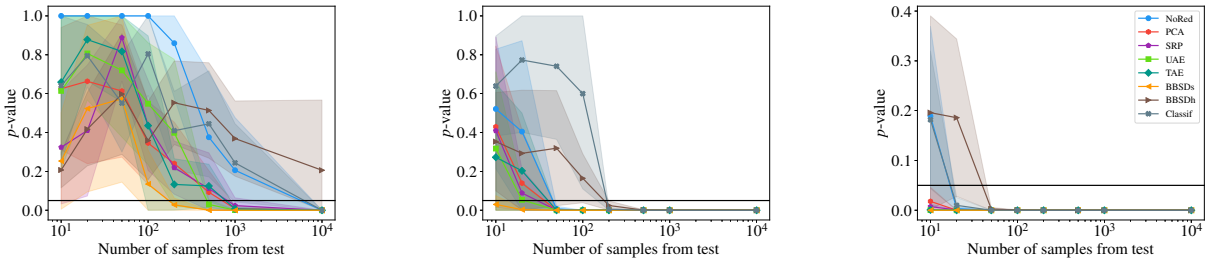
(e) Top similar samples.

Figure 12. MNIST small Gaussian noise shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

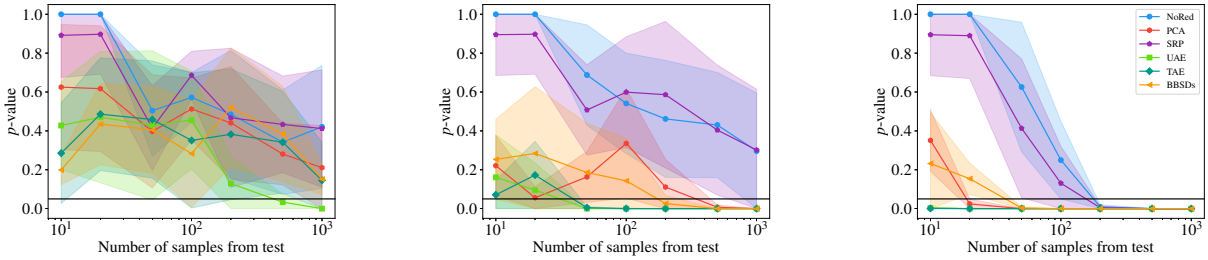
Figure 13. MNIST small Gaussian noise shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

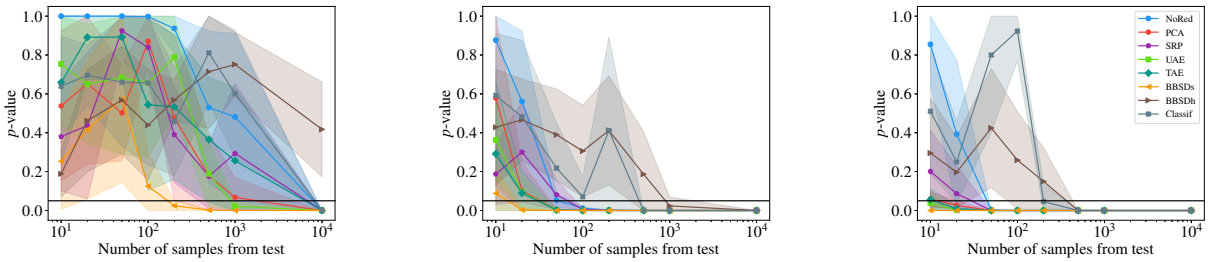
(d) Top different samples. (e) Top similar samples.

Figure 14. MNIST large image shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

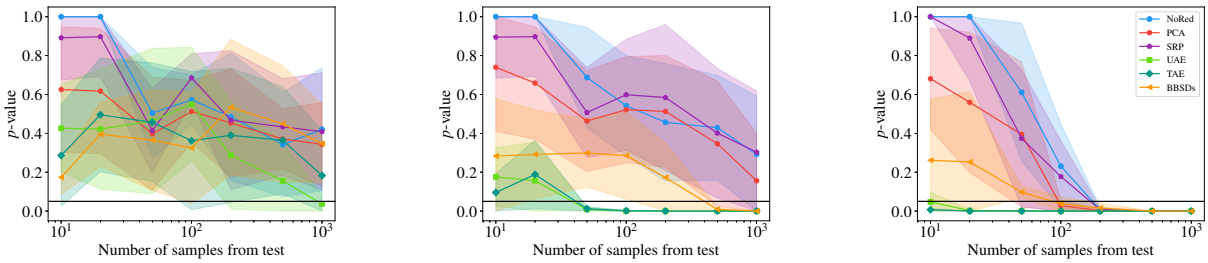
Figure 15. MNIST large image shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

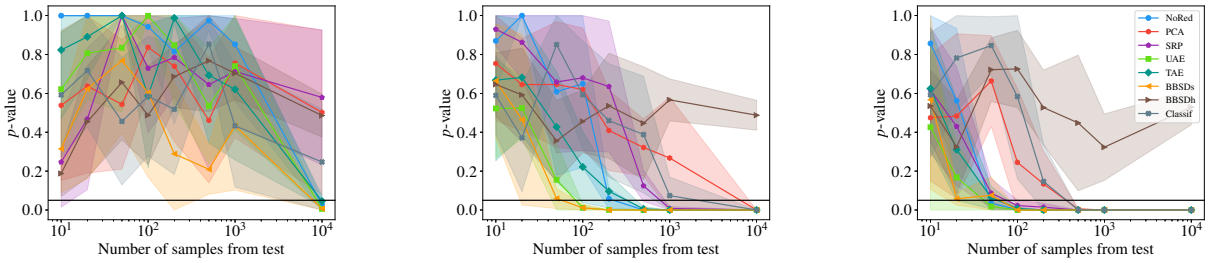
(d) Top different samples. (e) Top similar samples.

Figure 16. MNIST medium image shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

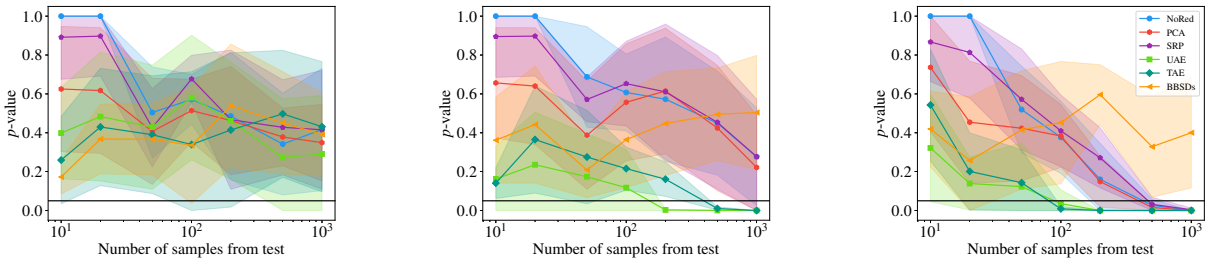
Figure 17. MNIST medium image shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

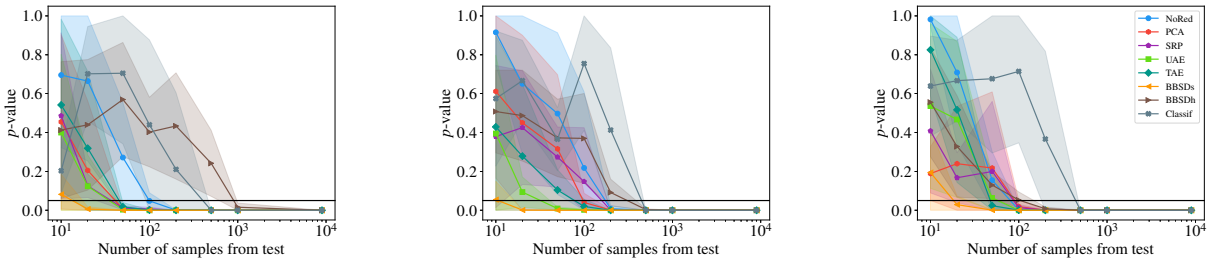
(d) Top different samples. (e) Top similar samples.

Figure 18. MNIST small image shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

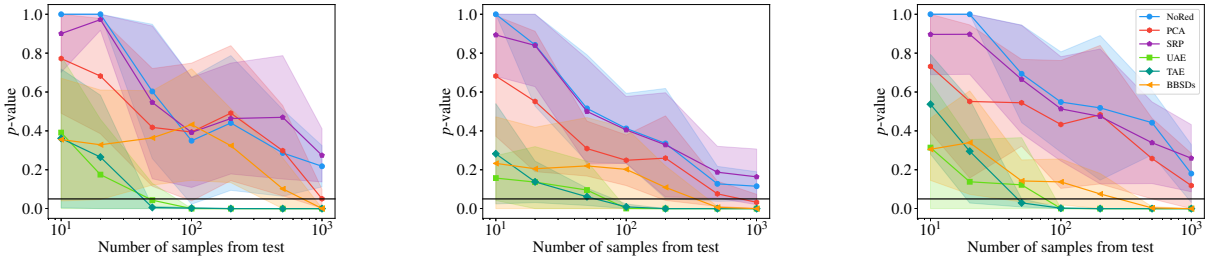
Figure 19. MNIST small image shift, multivariate two-sample tests.



(a) Knock out 10% of class 0. (b) Knock out 50% of class 0. (c) Knock out 100% of class 0.

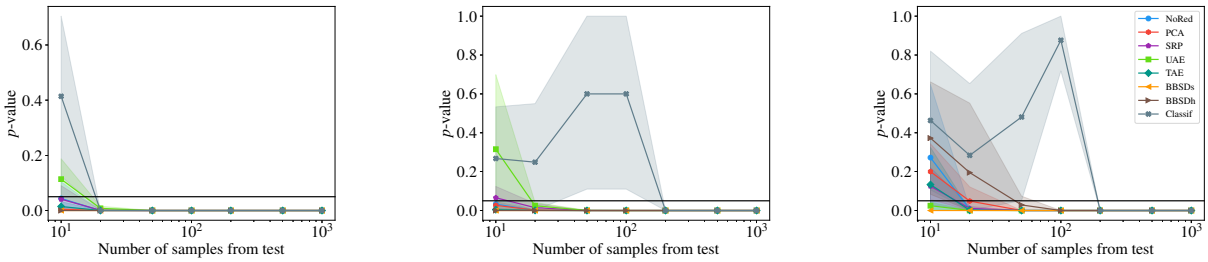
(d) Top different samples. (e) Top similar samples.

Figure 20. MNIST medium image shift (50%, fixed) plus knock-out shift (variable), univariate two-sample tests + Bonferroni aggregration.



(a) Knock out 10% of class 0. (b) Knock out 50% of class 0. (c) Knock out 100% of class 0.

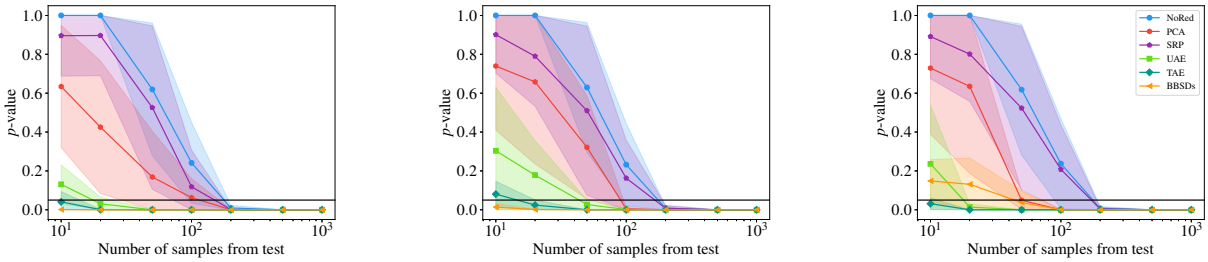
Figure 21. MNIST medium image shift (50%, fixed) plus knock-out shift (variable), multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

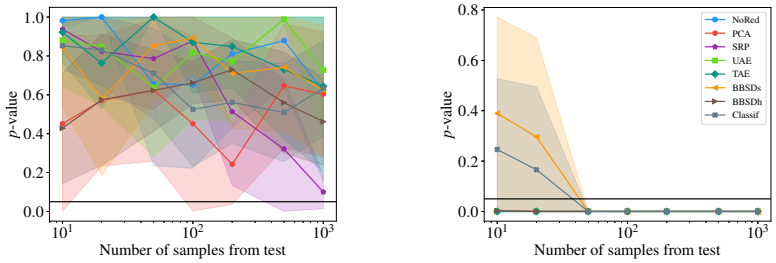
(d) Top different samples. (e) Top similar samples.

Figure 22. MNIST only-zero shift (fixed) plus medium image shift (variable), univariate two-sample tests + Bonferroni aggregration.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

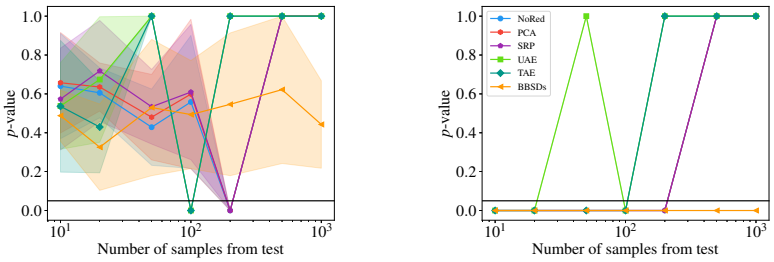
Figure 23. MNIST only-zero shift (fixed) plus medium image shift (variable), multivariate two-sample tests.



(a) Randomly shuffled dataset with same split proportions as original dataset. (b) Original split.

(c) Top different samples. (d) Top similar samples.

Figure 24. MNIST to USPS domain adaptation, univariate two-sample tests + Bonferroni aggregration.

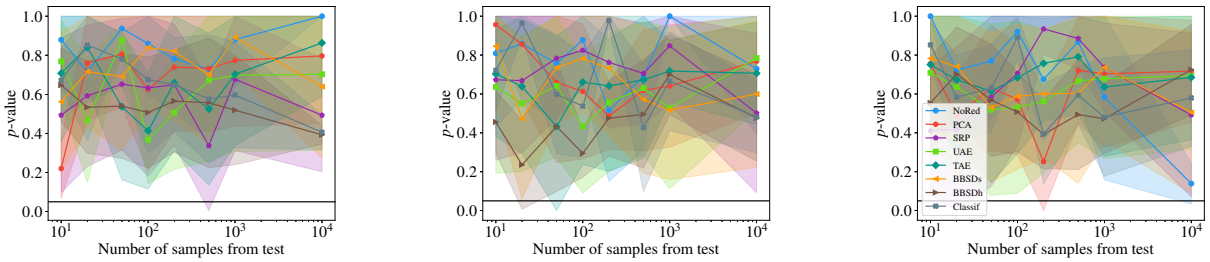


(a) Randomly shuffled dataset with same split proportions as original dataset.

(b) Original split.

Figure 25. MNIST to USPS domain adaptation, multivariate two-sample tests.

A.1.2. CIFAR-10



(a) 10% adversarial samples.

(b) 50% adversarial samples.

(c) 100% adversarial samples.

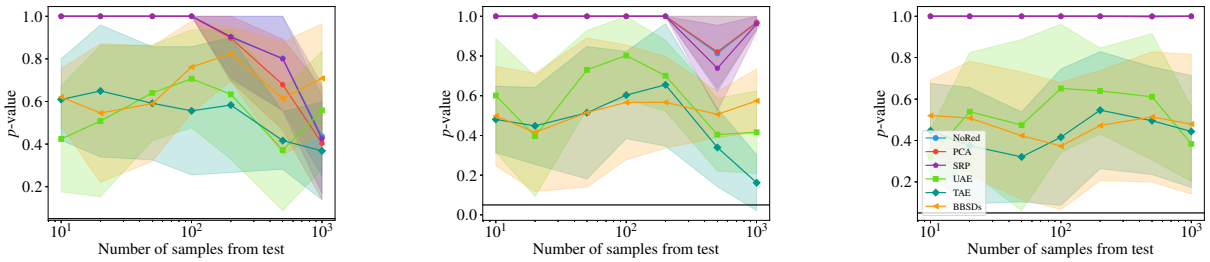
No samples available as *Classif* did not detect a shift.

(d) Top different samples.

No samples available as *Classif* did not detect a shift.

(e) Top similar samples.

Figure 26. CIFAR-10 adversarial shift, univariate two-sample tests + Bonferroni aggregration.

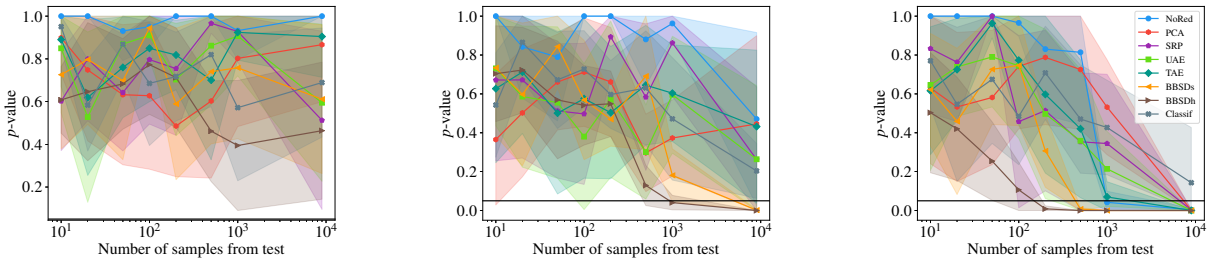


(a) 10% adversarial samples.

(b) 50% adversarial samples.

(c) 100% adversarial samples.

Figure 27. CIFAR-10 adversarial shift, multivariate two-sample tests.



(a) Knock out 10% of class 0. (b) Knock out 50% of class 0. (c) Knock out 100% of class 0.

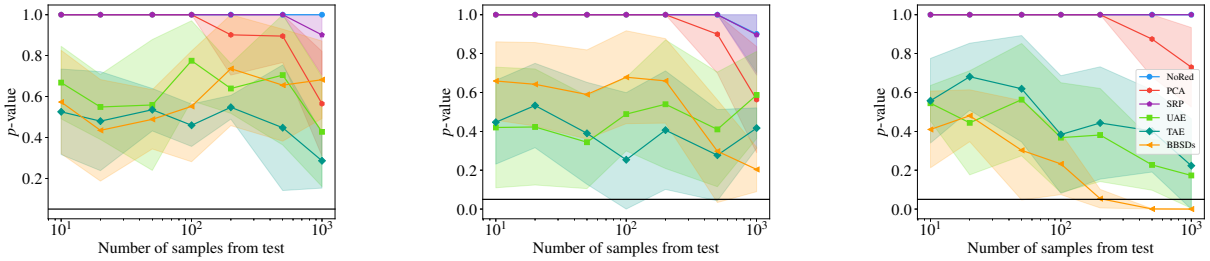
No samples available as *Classif* did not detect a shift.

(d) Top different samples.

No samples available as *Classif* did not detect a shift.

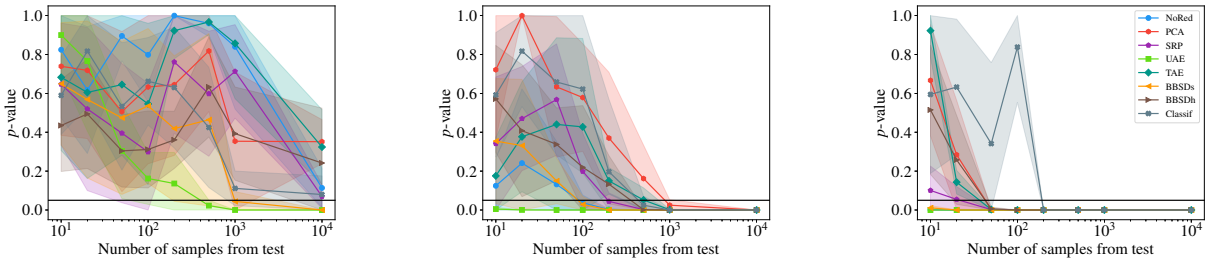
(e) Top similar samples.

Figure 28. CIFAR-10 knock-out shift, univariate two-sample tests + Bonferroni aggregration.



(a) Knock out 10% of class 0. (b) Knock out 50% of class 0. (c) Knock out 100% of class 0.

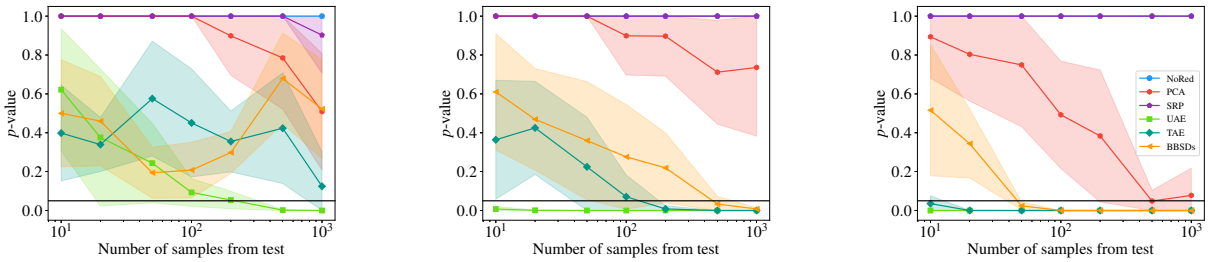
Figure 29. CIFAR-10 knock-out shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

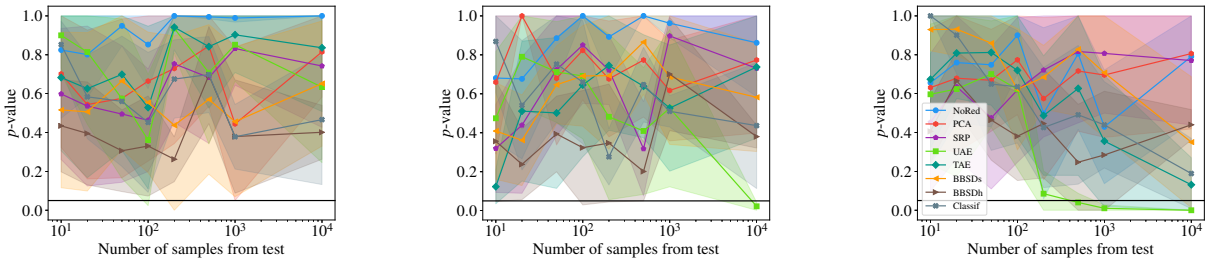
(d) Top different samples. (e) Top similar samples.

Figure 30. CIFAR-10 large Gaussian noise shift, univariate two-sample tests + Bonferroni aggregration.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

Figure 31. CIFAR-10 large Gaussian noise shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

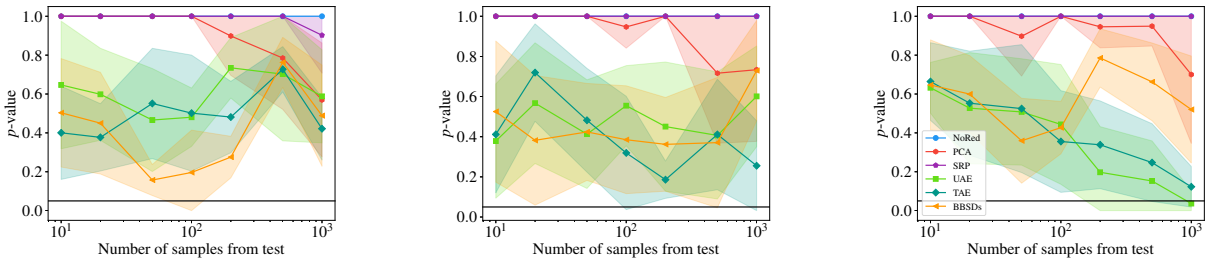
No samples available as *Classif* did not detect a shift.

(d) Top different samples.

No samples available as *Classif* did not detect a shift.

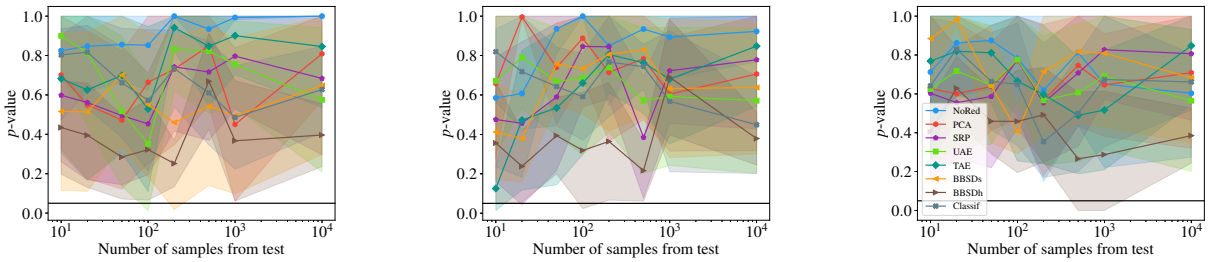
(e) Top similar samples.

Figure 32. CIFAR-10 medium Gaussian noise shift, univariate two-sample tests + Bonferroni aggregration.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

Figure 33. CIFAR-10 medium Gaussian noise shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

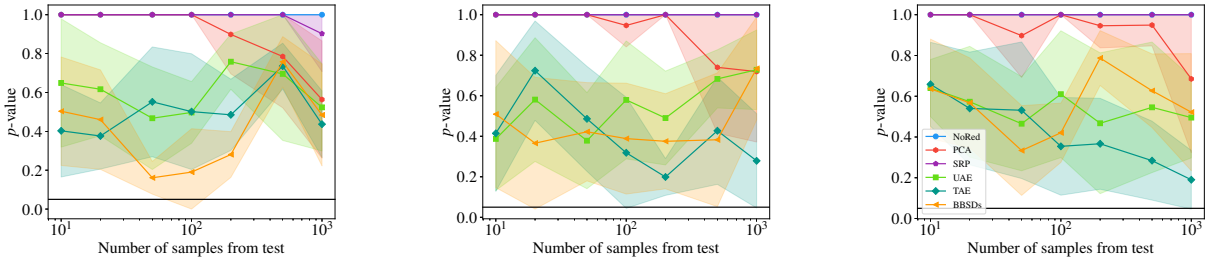
No samples available as *Classif* did not detect a shift.

(d) Top different samples.

No samples available as *Classif* did not detect a shift.

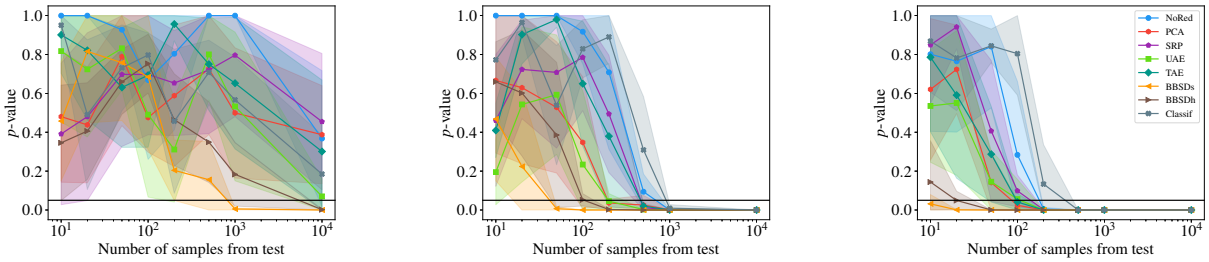
(e) Top similar samples.

Figure 34. CIFAR-10 small Gaussian noise shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

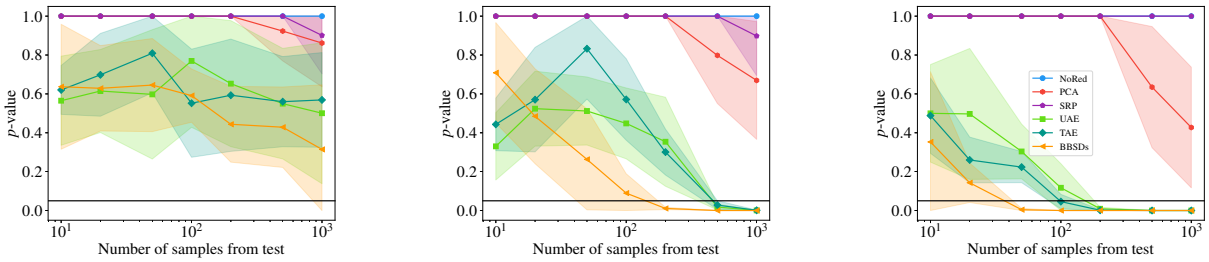
Figure 35. CIFAR-10 small Gaussian noise shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

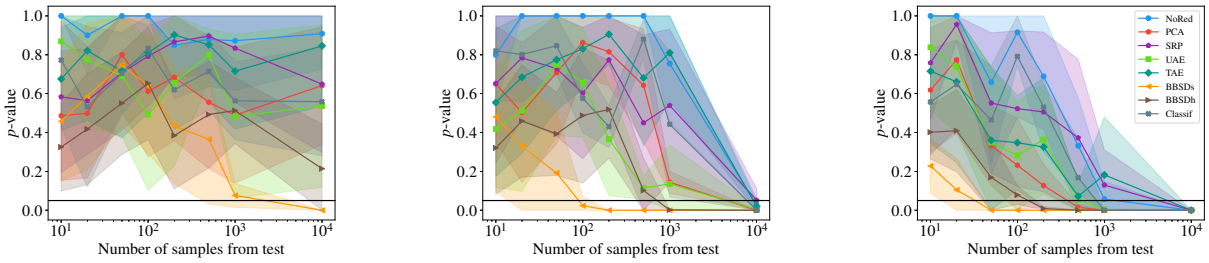
(d) Top different samples. (e) Top similar samples.

Figure 36. CIFAR-10 large image shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

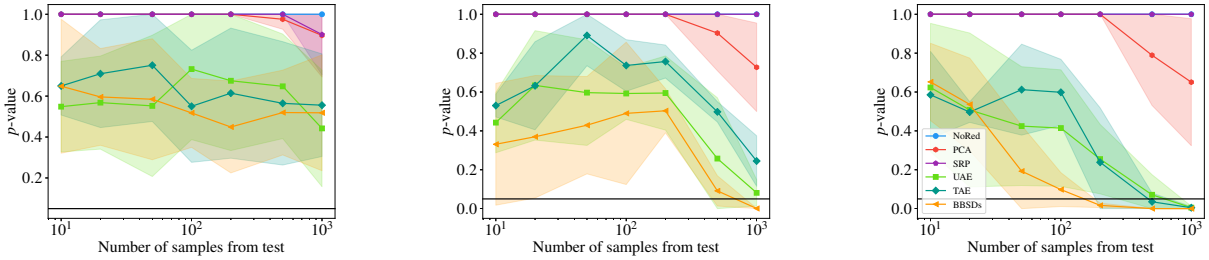
Figure 37. CIFAR-10 large image shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

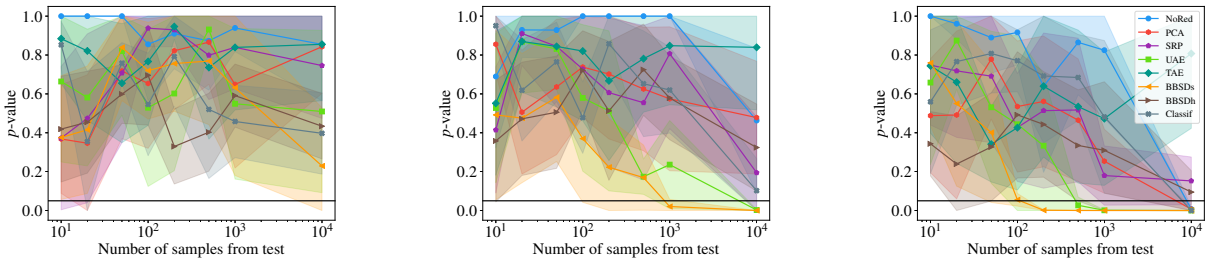
(d) Top different samples. (e) Top similar samples.

Figure 38. CIFAR-10 medium image shift, univariate two-sample tests + Bonferroni aggregation.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

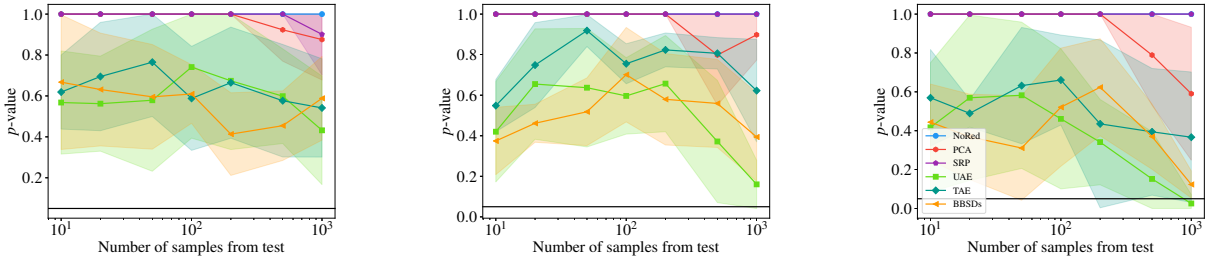
Figure 39. CIFAR-10 medium image shift, multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

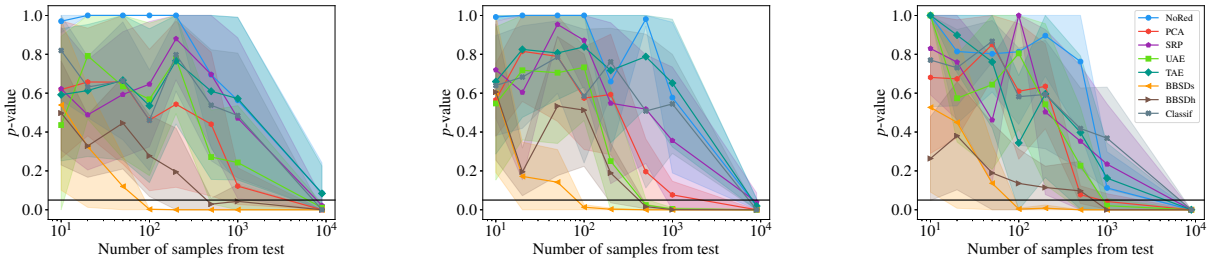
(d) Top different samples. (e) Top similar samples.

Figure 40. CIFAR-10 small image shift, univariate two-sample tests + Bonferroni aggregration.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

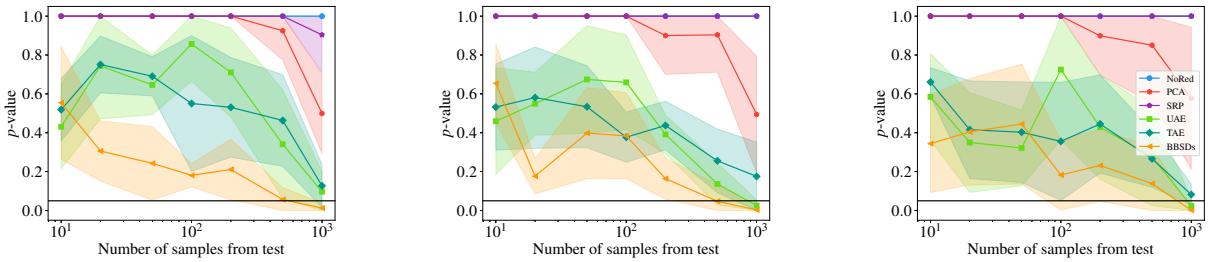
Figure 41. CIFAR-10 small image shift, multivariate two-sample tests.



(a) Knock out 10% of class 0. (b) Knock out 50% of class 0. (c) Knock out 100% of class 0.

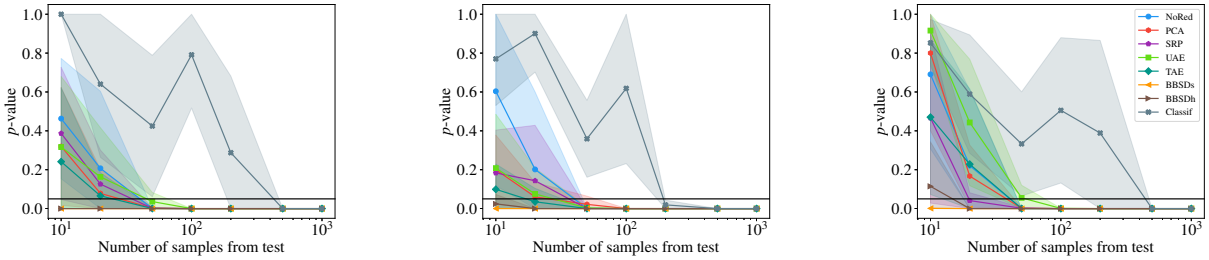
(d) Top different samples. (e) Top similar samples.

Figure 42. CIFAR-10 medium image shift (50%, fixed) plus knock-out shift (variable), univariate two-sample tests + Bonferroni aggregration.



(a) Knock out 10% of class 0. (b) Knock out 50% of class 0. (c) Knock out 100% of class 0.

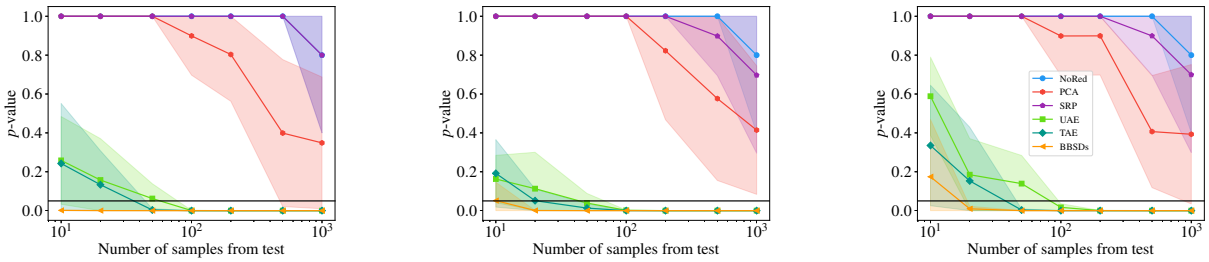
Figure 43. CIFAR-10 medium image shift (50%, fixed) plus knock-out shift (variable), multivariate two-sample tests.



(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

(d) Top different samples. (e) Top similar samples.

Figure 44. CIFAR-10 only-zero shift (fixed) plus medium image shift (variable), univariate two-sample tests + Bonferroni aggregration.

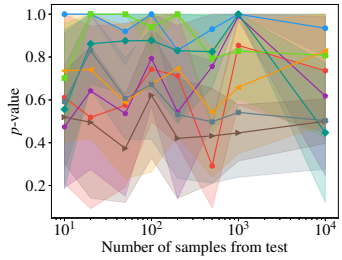


(a) 10% perturbed samples. (b) 50% perturbed samples. (c) 100% perturbed samples.

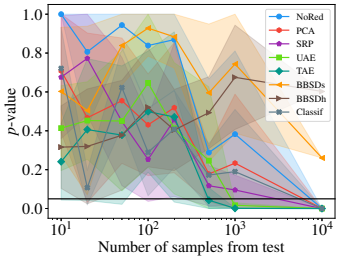
Figure 45. CIFAR-10 only-zero shift (fixed) plus medium image shift (variable), multivariate two-sample tests.

A.2. Original Splits

A.2.1. MNIST



(a) Randomly shuffled dataset with same split proportions as original dataset.

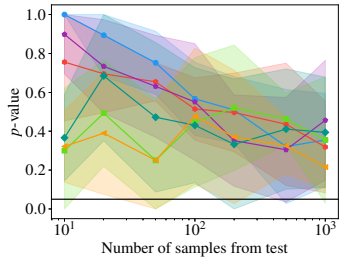


(b) Original split.

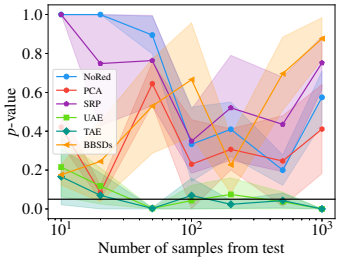
(c) Top different samples.

(d) Top similar samples.

Figure 46. MNIST randomized and original split, univariate two-sample tests + Bonferroni aggregation.



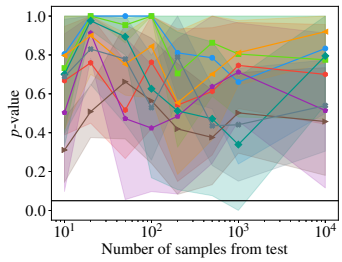
(a) Randomly shuffled dataset with same split proportions as original dataset.



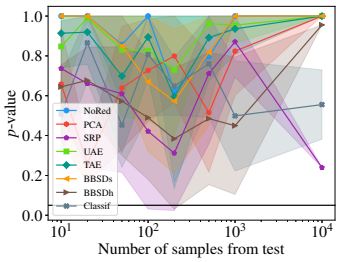
(b) Original split.

Figure 47. MNIST randomized and original split, multivariate two-sample tests.

A.2.2. FASHION MNIST



(a) Randomly shuffled dataset with same split proportions as original dataset.



(b) Original split.

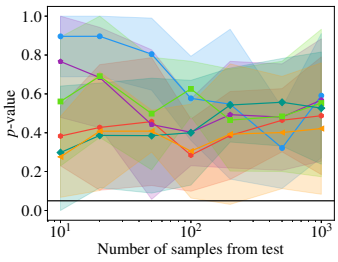
No samples available as *Classif* did not detect a shift.

(c) Top different samples.

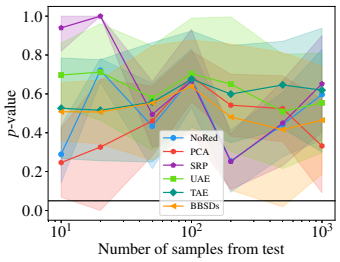
No samples available as *Classif* did not detect a shift.

(d) Top similar samples.

Figure 48. Fashion MNIST randomized and original split, univariate two-sample tests + Bonferroni aggregation.



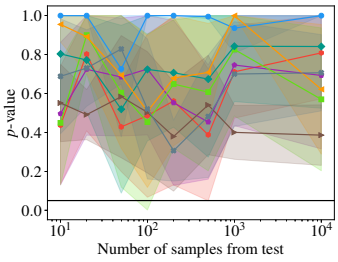
(a) Randomly shuffled dataset with same split proportions as original dataset.



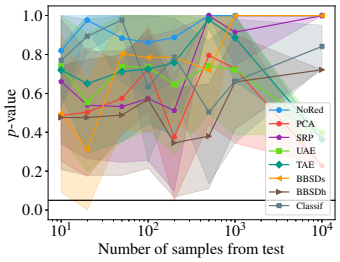
(b) Original split.

Figure 49. Fashion MNIST randomized and original split, multivariate two-sample tests.

A.2.3. CIFAR-10



(a) Randomly shuffled dataset with same split proportions as original dataset.



(b) Original split.

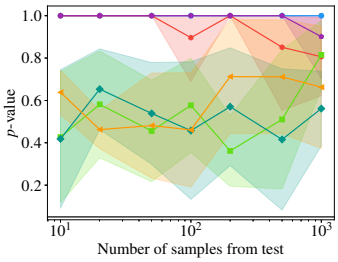
No samples available as *Classif* did not detect a shift.

(c) Top different samples.

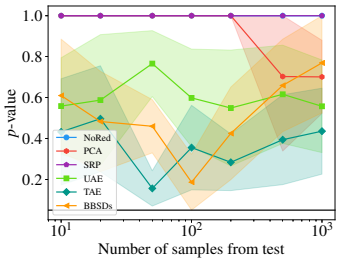
No samples available as *Classif* did not detect a shift.

(d) Top similar samples.

Figure 50. CIFAR-10 randomized and original split, univariate two-sample tests + Bonferroni aggregration.



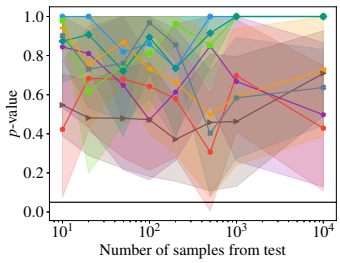
(a) Randomly shuffled dataset with same split proportions as original dataset.



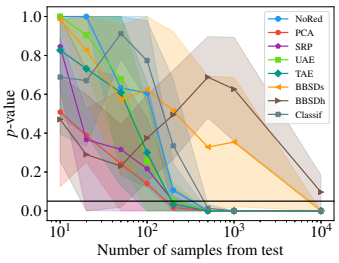
(b) Original split.

Figure 51. CIFAR-10 randomized and original split, multivariate two-sample tests.

A.2.4. SVHN



(a) Randomly shuffled dataset with same split proportions as original dataset.

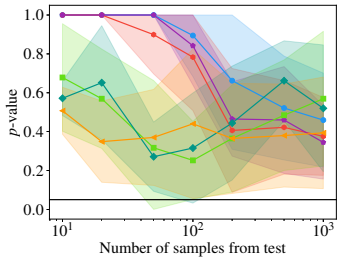


(b) Original split.

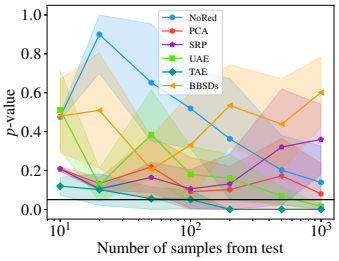
(c) Top different samples.

(d) Top similar samples.

Figure 52. SVHN randomized and original split, univariate two-sample tests + Bonferroni aggregation.



(a) Randomly shuffled dataset with same split proportions as original dataset.



(b) Original split.

Figure 53. SVHN randomized and original split, multivariate two-sample tests.