SEQUENTIAL COVARIATE SHIFT DETECTION USING CLASSIFIER TWO-SAMPLE TESTS

Anonymous authors
Paper under double-blind review

ABSTRACT

A standard assumption in supervised learning is that the training data and test data are from the same distribution. However, this assumption often fails to hold in practice, which can cause the learned model to perform poorly. We consider the problem of detecting covariate shift, where the covariate distribution shifts but the conditional distribution of labels given covariates remains the same. This problem can naturally be solved using a two-sample test—i.e., test whether the current test distribution of covariates equals the training distribution of covariates. Our algorithm builds on classifier tests, which train a discriminator to distinguish train and test covariates, and then use the accuracy of this discriminator as a test statistic. A key challenge is that classifier tests assume given a fixed set of test covariates. In practice, test covariates often arrive sequentially over time—e.g., a self-driving car observes a stream of images while driving. Furthermore, covariate shift can occur multiple times—i.e., shift and then shift back later or gradually shift over time. To address these challenges, our algorithm trains the discriminator online. Furthermore, it evaluates test accuracy using each new covariate before taking a gradient step; this strategy avoids constructing a held-out test set, which can reduce sample efficiency. We prove that this optimization preserves the correctness—i.e., our algorithm achieves a desired bound on the false positive rate. In our experiments, we show that our algorithm efficiently detects covariate shifts on ImageNet.

1 INTRODUCTION

A key challenge facing deep neural networks is their sensitivity to changes in the data distribution. In particular, supervised learning traditionally assumes that the training and test data are from the same distribution (Vapnik, 1998), but this assumption often fails in practice. For example, an autonomous car using perception to identify obstacles needs to be robust to shifts such as changes in the weather and lighting conditions. We focus on covariate shift (Shimodaira, 2000), where there is a shift in the covariate distribution $p(x)$, and the conditional label distribution $p(y|x)$ remains unchanged. Covariate shift can reduce model performance (Sugiyama & Müller, 2005), invalidate uncertainty estimates (Ovadia et al., 2019; Park et al., 2020), and affect model selection (Sugiyama et al., 2007).

One strategy is to devise an algorithm to detect covariate shift; if detected, the algorithm can alert the user that predictions may be unreliable. Covariate shift detection can be formulated as two-sample hypothesis test (Gretton et al., 2012; Rabanser et al., 2018; Liu et al., 2020), where the goal is to determine whether two sets of examples are from the same distribution. To test for covariate shift, we choose the first sample to be the data used to train the model and the second sample to be recent test data given as input to the model. Then, the detector returns “covariate shift” if the hypothesis test indicates that the two samples are from different distributions and “no shift” otherwise.

We propose a detection algorithm based on classifier tests (Lopez-Paz & Oquab, 2017; Cheng & Cloninger, 2019; Kim et al., 2021), which use the accuracy of a classifier trained to distinguish the two samples as the test statistic. In particular, if the two samples are from the same distribution, then the accuracy should be $1/2$; otherwise, it should be $>1/2$. Since the test statistic follows a binomial distribution, we use the Clopper-Pearson interval (Clopper & Pearson, 1934) (an exact confidence interval for the unknown success probability of the Binomial distribution) to derive the cutoff. In contrast, prior work relies on asymptotics to derive the cutoff, which results in approximations.
A key challenge is that the test examples are typically obtained over time—e.g., an autonomous robot continuously perceives its environment, and we want to detect if its distribution of observations shifts at any time. There are two key challenges to leveraging classifier tests in this setting. First, they rely on training a classifier to distinguish training and test examples; doing so on every step would be computationally intractable. Second, they rely on a held-out test set to estimate the test statistic, but constructing such a set online would reduce sample efficiency.

Rather than train a classifier at each step, our proposed algorithm trains a model online using stochastic gradient descent. Then, rather than construct a held-out test set, our algorithm evaluates the accuracy of the model online using each example before taking a gradient step on that example. We prove that this strategy results in an unbiased estimate of the model accuracy; as a consequence, the finite-sample guarantees on the false positive rate provided by the sequential test continue to hold. In addition, we prove bounds on the false negative rate under mild conditions on the classifier (i.e., it achieves nontrivial accuracy distinguishing the two distributions).

We evaluate our approach on both synthetic and natural shifts on the ImageNet (Russakovsky et al., 2015) dataset. In particular, we demonstrate that our approach achieves better sample efficiency than baseline algorithms; furthermore, it uniformly satisfies the desired false positive rate. Thus, our algorithm is an effective strategy for sequential covariate shift detection.

Contributions. We formulate (sequential) covariate shift detection as a two-sample test, and propose a novel algorithm to solve this problem (Section 3). Then, we prove finite sample bounds on false positive rate and false negative rate achieved by our algorithm (Section 4). Finally, we empirically demonstrate that our algorithm effectively detects shifts on ImageNet (Section 5).

Sequential detection vs. sequential tests. While we consider the sequential setting, we deliberately choose not use a sequential hypothesis test, since the covariate shift may occur after a delay or gradually over time. A sequential test only applies if all of the test data is shifted. Furthermore, since we are not using sequential tests, the false positive rate bound only holds per-step, not uniformly across all steps. This is necessary: we cannot guarantee that we detect a covariate shift occurring at a later point in time if we constrain the false positive to be bounded uniformly across all steps. In our experiments, we show that the rate of false alarms remains manageable while enabling our algorithm to detect covariate shift in a number of interesting scenarios.

2 RELATED WORK

Covariate shift. There has been work on training models in the presence of covariate shift. In particular, in the unsupervised domain adaptation setting (Ben-David et al., 2007; Bickel et al., 2007; Ganin et al., 2016), the algorithm has access to labeled examples from the source domain but only unlabeled examples from the target domain, and the goal is to train a model that achieves good performance on the target domain. One strategy is to use importance weighting to upweight source examples that are more similar to target examples (Bickel et al., 2007). Another strategy is to first learn an invariant representation (Ganin et al., 2016), which is an embedding space where the source and target examples are similar, and then train a model on this embedding space using the source examples. If we detect covariate shift, one solution is to retrain the model using these techniques.

Two-sample tests. We focus on classifier two-sample tests (C2ST). In this approach, the idea is to train a binary classifier to distinguish source and target samples, compute a real-valued score based on this classifier as the test statistic, and then use a univariate two-sample test to determine the cutoff for rejecting the null hypothesis (Friedman, 2004). A natural test statistic is the classifier’s accuracy on a held-out test set (Kim et al., 2021; Lopez-Paz & Oquab, 2017), or the differences in the classifier’s logits (Cheng & Cloninger, 2019); in this work, we use the former. One way to compute the cutoff is to use the asymptotic distribution of the test statistic (Lopez-Paz & Oquab, 2017). Nonparametric tests such as permutation tests can also be used (Kim et al., 2021).

Another kind of two-sample test is called a kernel two-sample test. In this approach, the idea is to use the maximum mean discrepancy (MMD) between the two samples according to a given kernel embedding as the test statistic (Gretton et al., 2012a; Chwialkowski et al., 2015; Jitkrittum et al., 2016). The key design decision is the choice of kernel. One strategy is to use a nonparametric kernel such as Gaussian radial basis functions (Gretton et al., 2012a); alternatively, the kernel can also be optimized to minimize the false negative rate of the resulting test (Gretton et al., 2012b). In addition,
recent work has shown how to first learn a kernel function in the form of a deep neural network, and then evaluate the MMD distance on a held-out test set \cite{Liu et al., 2020}. The test statistic can be chosen based on finite sample bounds or based on its asymptotic distribution \cite{Gretton et al., 2012a}. Nonparametric permutation tests can also be used \cite{Liu et al., 2020} with this approach.

Concept Drift. In the context of concept drift \cite{Gama et al., 2014}, there has been work detecting shifts in $p(x, y)$ \cite{Gonçalves Jr et al., 2014}. However, this work assumes that ground truth labels are provided for test examples, whereas our approach only requires unlabeled test examples. The former is substantially easier, since it suffices to check for drift in the distribution of prediction errors, which is usually very simple (e.g., a Bernoulli distribution for the 0-1 loss), making it easy to test for drift. In contrast, our approach checks for drift in high-dimensional covariates distribution.

Sequential hypothesis testing. A closely related problem is sequential hypothesis testing, which adaptively decides whether to reject the null hypothesis as samples become available \cite{Wald, 1945}. These approaches can also applied to two-sample testing \cite{Balsubramani & Ramdas, 2015, Lhéritier & Cazals, 2018, 2019, Manole & Ramdas, 2021}. However, as discussed above, they assume that the each distribution of the two samples does not change over time. In contrast, we are interested in the setting where the test examples might initially be from the same distribution as the training examples, but then shift at a later point in time. Sequential tests are not applicable to this setting.

3 SEQUENTIAL COVARIATE SHIFT DETECTION

3.1 PROBLEM FORMULATION

Let $X$ be the covariate space, $S$ be the source distribution over $X$, and $T_{1:t_2} = (T_{11}, T_{1:t_2+1}, \ldots, T_{t_2})$ be a sequence of target distributions over $X$ from time steps $t_1$ to $t_2$. On time step $t$, we consider samples $x_t \sim S$ and $x_t' \sim T_t$; in practice, $S$ can be taken to be the uniform distribution over the training set. We let $S_{w,t} = (x_{t-w+1}, x_{t-w+2}, \ldots, x_t)$ and $T_{w,t} = (x_{t-w+1}', x_{t-w+2}', \ldots, x_t')$ denote the recent examples in a time window of a given size $w \in \mathbb{N}$.

Our goal is to detect covariate shift at any step $t$. More precisely, we want to determine whether $S \neq T_{w,t}$, where $T_{w,t} = \sum_{k=t-w+1}^{t} T_k / w$, i.e., whether the average target distributions over the previous $w$ steps is shifted compared to $S$. For a fixed step $t$, this problem is a two-sample hypothesis test \cite{Lehmann & Romano, 2006}, where the null hypothesis is $H_0 : S = T_{w,t}$, and the alternative hypothesis is $H_1 : S \neq T_{w,t}$. That is, a two-sample test $f$ is designed to compute

$$f(S_{w,t}, T_{w,t}) \approx \begin{cases} 1 & \text{if } S \neq T_{w,t} \\ 0 & \text{otherwise.} \end{cases}$$

Our goal is to design a two-sample test $f$ for detecting covariate shift with this data stream. While we can in principle use any two-sample test, our goal is to design one that is both sample and computationally efficient while achieving high accuracy for high-dimensional data such as images. In addition, we want the test $f$ to come with finite sample guarantees on the false positive rate. In particular, given $\alpha \in \mathbb{R}_{>0}$, if $S = T_{w,t}$, we want to ensure

$$\mathbb{P}_{S_{w,t} \sim S, T_{w,t} \sim T_{1:w+1}} \left[ f(S_{w,t}, T_{w,t}; \alpha) = 0 \right] \geq 1 - \alpha.$$ 

Ideally, we also want to provide finite sample bounds on the false negative rate; however, for classifier tests, we can only do so under additional assumptions about the model family used to try and distinguish $S$ and $T_{w,t}$. Intuitively, we assume that (i) the model family has bounded complexity (e.g., Rademacher complexity), and (ii) some model exists in the family that achieves nontrivial accuracy at distinguishing $S$ and $T_{w,t}$. Then, our goal is to ensure that if $S \neq T_{w,t}$, we have

$$\mathbb{P}_{S_{w,t} \sim S, T_{w,t} \sim T_{1:w+1}} \left[ f(S_{w,t}, T_{w,t}; \alpha) = 1 \right] \geq 1 - M(\alpha, w)$$

for some function $M(\alpha, w)$ that depends on the model family.

3.2 ALGORITHM OVERVIEW

Next, we describe our two-sample test. We build on classifier two-sample test (C2ST) \cite{López-Paz & Oquab, 2017, Kim et al., 2021}. The idea is to train a classifier $\hat{g}_t$ to try and distinguish $S_{w,t}$
The key challenge is what data to use as the test dataset to estimate the accuracy of \( \hat{g}_t \) before using it to train \( \hat{g}_t \). In the next section, we prove that this strategy maintains the independence of our estimate of the accuracy of \( \hat{g}_t \) (Lemma 1), and that as a consequence, our algorithm satisfies the desired false positive rate (for a single step \( t \)).

3.3 Algorithm Details

**Sequential detection algorithm.** At each time step \( t \), we observe a source sample \( x_t \), \( x_t \sim S \) and a target sample \( x_t' \), \( x_t' \sim T_t \). By using these current samples and previous samples, we detect covariate shifts by updating the source-target classifier in online learning. In particular, our algorithm consists of three steps: (1) source-target prediction, (2) covariate shift detection, and (3) online source-target classifier update. The following and Algorithm 1 include details.
Step 1. Source-target prediction. We predict source-target labels on the current samples $x_t$ and $x'_t$ using the current source-target classifier $\hat{g}_t$. In particular, we denote prediction on the source sample $x_t$ by $\hat{y}_t = \hat{g}_t(x_t)$, and denote prediction on the target sample $x'_t$ by $\hat{y}'_t = \hat{g}_t(x'_t)$. These predictions and previous predictions are used in covariate shift detection in the following step.

Step 2. Calibrated covariate shift detection. Let $Q_{w,t}$ be a distribution over $\mathcal{X} \times \{0, 1\}$, where
\[
Q_{w,t}(x, y) := \frac{1}{2} \cdot \mathcal{S}(x) \cdot \mathbb{1}(y = 0) + \frac{1}{2} \cdot \mathcal{T}_{w,t}(x) \cdot \mathbb{1}(y = 1).
\]
Then, $z = \mathbb{1}(\hat{g}_t(x) = y)$ is a Bernoulli random variable with distribution $\text{Bernoulli}(\mu^*_w, t)$, where
\[
\mu^*_w = \mathbb{P}(x, y) \sim Q_{w,t}[\hat{g}_t(x) = y]
\]
is the accuracy of $\hat{g}_t$ at distinguishing whether an example $x$ is from distribution $\mathcal{S}$ or $\mathcal{T}_{w,t}$. The unbiased empirical estimate of this accuracy is denoted by
\[
\tilde{\mu}_{w,t} = \frac{1}{2w} \sum_{i=t-w+1}^{t} (\mathbb{1}(\hat{y}_i = y_i) + \mathbb{1}(\hat{y}'_i = y'_i)).
\]
In fact, $2w\tilde{\mu}_{w,t}$ is a Binomial random variable with Binomial$(2w, \mu^*_w)$; thus, the accuracy $\mu^*_w$ can be estimated by the Clopper-Pearson (CP) interval $\Theta_{\text{CP}}(2w\tilde{\mu}_{w,t}, 2w; \alpha)$ that includes the unknown parameter $\mu^*_w$, with high probability, i.e.,
\[
\mathbb{P}[\mu^* \in \Theta_{\text{CP}}(2w\tilde{\mu}_{w,t}, 2w; \alpha)] \geq 1 - \alpha.
\]
This property can be used for checking the accuracy of $\hat{g}_t$ might be $1/2$. In particular, our algorithm returns “covariate shift” if $1/2 \not\in \Theta_{\text{CP}}(2w\tilde{\mu}_{w,t}, 2w; \alpha)$, and “no covariate shift” otherwise, i.e.,
\[
\hat{f}(S_{w,t}, T_{w,t}; \alpha) = \mathbb{1}[\frac{1}{2} \not\in \Theta_{\text{CP}}(2w\tilde{\mu}_{w,t}, 2w; \alpha)].
\]
Here, the Clopper-Pearson interval calibrates the empirical accuracy $\tilde{\mu}_{w,t}$ using the property of the Binomial distribution.

Step 3. Online source-target classifier update. Finally, we update a binary classifier $\hat{g}_t$ using new training examples based on the source and target samples, i.e., $(x_t, 0)$ and $(x'_t, 1)$. In general, $\hat{g}_t$ can be any model; we consider it to be a neural network, in which case we can update its parameters using stochastic gradient descent with respect to the cross entropy loss.

4 Theoretical Guarantees

In this section, we describe our finite sample bounds on the false positive and false negative rates of our covariate shift detector $\hat{f}$; the key to have valid bounds is proving the independence on predictions $\hat{y}_1, \ldots, \hat{y}_t$ and $(\hat{y}'_1, \ldots, \hat{y}'_t)$ to have a valid Clopper-Pearson interval, since they are seemingly dependent through the online learned classifier $\hat{g}_t$. First, our key result shows that our estimate of the accuracy of $\hat{g}_t$ is valid—i.e., the predictions $\hat{y}_i, \ldots, \hat{y}_j$ are conditionally independent (see Appendix A.1 for a proof), thus the accuracy is the parameter of the Binomial distribution:

Lemma 1. If $x_1, \ldots, x_j$ are independent for any $i, j \in \mathbb{N}$ where $i < j$, $\hat{y}_i, \ldots, \hat{y}_j$ are conditionally independent given $\hat{y}_i, \ldots, \hat{y}_{j-1}$.

Our next result says that our algorithm ensures the desired bound $\alpha$ on the false positive rate (i.e., $\hat{f}$ says “covariate shift” when there is no covariate shift). To this end, we exploit the following observation that any source-target classifier makes the expected accuracy of $1/2$ if there is no covariate shift. Intuitively, if $S = T_{w,t}$, source-target classification is impossible (see Appendix A.2 for a proof):

Lemma 2. If $S = T_{w,t}$, we have $\mu^*_w = 1/2$ for any source-target classifier $\hat{g}_t$.

Since the expected accuracy of $\hat{g}_t$ is $1/2$ regardless of how we design and learn $\hat{g}_t$, and how many samples are used to learn $\hat{g}_t$, the Clopper-Pearson interval includes the true accuracy with high probability; thus the false positive rate of the proposed covariate shift detector $\hat{f}$ is effectively controlled by the confidence level of the Clopper-Pearson interval, as follows (see Appendix A.3 for a proof):

\[
\mathbb{E}(\hat{f}(S_{w,t}, T_{w,t}; \alpha)) = \alpha.
\]
Theorem 3 (Bound on false positive rate). If \( S = \overline{T}_{w,t} \), then for any \( \hat{g}_t \), we have
\[
P(\hat{f}(S_{w,t}, T_{w,t}; \alpha) = 0) \geq 1 - \alpha.
\] (2)

Our next result provides a bound on the false negative rate; we first observe that the Clopper-Pearson interval is included in the interval by the Hoeffding’s bound. Intuitively, the Clopper-Pearson interval represents a lower and upper bound of the expected accuracy given an empirical accuracy tailored to a Bernoulli random variable; the Hoeffding’s bound can similarly bound the mean but in a more general setup. Thus, the Clopper-Pearson interval can be smaller (see Appendix A.3 for a proof).

Lemma 4. Let \( s \sim \text{Binomial}(n, p) \) and \( F(s; n, p) \) is the CDF of \( \text{Binomial}(n, p) \); we have
\[
\frac{s - \sqrt{\frac{\ln \frac{2}{P(\varepsilon)}}{2n}}}{\sqrt{n}} \leq \inf \left\{ \theta \mid F(n - s; n, 1 - \theta) \geq \frac{\alpha}{2} \right\} \quad \text{and} \quad \sup \left\{ \theta \mid F(s; n, \theta) \geq \frac{\alpha}{2} \right\} \leq \frac{s}{n} + \sqrt{\frac{\ln \frac{2}{P(\varepsilon)}}{2n}}.
\]

Leveraging this, we have the following bound on false negative rate (see Appendix A.3 for a proof).

Theorem 5 (Bound on false negative rate). Assume \( \hat{g}_t \) achieves nontrivial accuracy, i.e., \( \mu_{w,t} \geq 1/2 + \epsilon \), where \( \epsilon \in (0, 1/2] \), is the accuracy at distinguishing \( S \) and \( \overline{T}_{w,t} \). Let \( a(w, \alpha) := 2w(1/2 + \sqrt{\log(2)/4w}) \) and \( b(w, \alpha) := 2w(1/2 - \sqrt{\log(2)/4w}) \). If \( S \neq \overline{T}_{w,t} \) and \( w - 1 - \left[ \sqrt{w \log(2/\alpha)} \right] \geq 0 \), then we have
\[
P(\hat{f}(S_{w,t}, T_{w,t}; \alpha) = 1) \geq F\left(2w - a(w, \alpha) + 1; 2w, 1/2 - \epsilon\right) + F\left(b(w, \alpha) - 1; 2w, 1\right).
\] (3)

In the false negative bound, the first term is dominant and increases as \( w \) increases, which implies the sample size needs to be increased to have a powerful shift detector; the condition on \( w \) suggests that the bound is valid when \( w \geq 201 \) given \( \alpha = 0.01 \). We note that the assumption \( L(\hat{g}_t) := 1 - \mu^* \leq 1/2 - \epsilon \) can be achieved under standard conditions. For instance, assume that the model family \( \mathcal{G} \) of source-target classifiers has finite VC dimension (i.e., \( \text{VC}(\mathcal{G}) < \infty \)), and that the optimal model \( g^* \in \mathcal{G} \) has nontrivial inaccuracy \( L(g^*) = 1/2 - \xi \) for some \( \xi \in \mathbb{R}_{>0} \); then, with probability at least \( 1 - \delta \) with respect to \( S_{w,t} \) and \( T_{w,t} \) and letting \( m = 2w \), we have
\[
L(\hat{g}_t) \leq L(g^*) + 4\sqrt{\frac{\text{VC}(\mathcal{G}) \left( \log(2m) + 1 \right)}{m}} + \sqrt{\frac{\log(2/\delta)}{m}} \leq \frac{1}{2} - \left( \xi - 4\sqrt{\frac{\text{VC}(\mathcal{G}) \left( \log(2m) + 1 \right)}{m}} - \sqrt{\frac{\log(2/\delta)}{m}} \right)_{\geq \xi},
\]
where the second term (which we have taken to be \( \epsilon \)) satisfies \( \epsilon > 0 \) for sufficiently large \( m \).

5 Experiments

We evaluate the effectiveness of our algorithm at detecting both natural and synthetic covariate shifts of varying forms (e.g., gradual shifts and multiple shifts back and forth), showing that it significantly outperforms natural baselines.

5.1 Experiment Setup

Baselines. We compare our algorithm to three baselines: two of them differ in the way they use the samples at each time step, the third uses Wald’s sequential likelihood test [Wald, 1945]. For the first two baselines, while our approach uses all samples to construct the CP interval around the accuracy of the source-target classifier \( \hat{g}_t \) as well as to train \( \hat{g}_t \), the baseline instead constructs a held-out test set using every \( H^{th} \) sample. Then, only this held-out test set is used to compute the CP interval, and only the remaining samples are used to train \( \hat{g}_t \). In our experiments, we used values of \( H \in \{2, 5\} \), denoted H2, H5, respectively. For Wald’s test, we consider the Bernoulli random variable with a probability \( p \) indicating whether the prediction of the source-target classifier is correct for the given
Table 1: Scenario description for experiments. (a) “M-shift” is Multiple shift, (b) “GI-shift” is gradually increasing shift, and (c) “GID-shift” is gradually increasing-then-decreasing shift.

<table>
<thead>
<tr>
<th>Start position</th>
<th>Description</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>No shift</td>
<td>0.0</td>
</tr>
<tr>
<td>25%</td>
<td>Shift</td>
<td>1.0</td>
</tr>
<tr>
<td>50%</td>
<td>No shift</td>
<td>0.0</td>
</tr>
<tr>
<td>75%</td>
<td>Shift</td>
<td>1.0</td>
</tr>
<tr>
<td>80%</td>
<td>Shift</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Sample. The hypothesis test is $H_0 : p = 0.5$ vs. $H_1 : p = 0.5 + \epsilon$, where $\epsilon = 0.2$ in our experiments; we restart the test each time it makes a decision.

This baseline is essentially the online version of an existing classifier two-sample test (C2ST) (Kim et al., 2021; Lopez-Paz & Oquab, 2017), which splits the (fixed) training dataset into a training set to train $g_t$ and a held-out test set to estimate the accuracy of $g_t$; thus, $H$ controls the tradeoff between the number of examples in the training set and held-out test set.

**Source-target classifier.** We use a fully-connected neural network with a single hidden layer (with 128 hidden units) and with the ReLU activation functions as the source-target classifier $g_t$. We use a binary cross-entropy loss for training in conjunction with an SGD optimizer with a learning rate of 0.01 (for natural shift experiments) and 0.001 (for synthetic shift experiments). Finally, since the inputs are ImageNet images (Russakovsky et al., 2015), we use a 2048-dimensional feature vector generated by first running a pretrained ResNet152 model (He et al., 2016) on the images, and then using these features vectors for the covariates of $S_{w,t}$ and $T_{w,t}$.

**Scenarios.** We run each algorithm to test whether the target distribution in the given window is shifted with three different scenarios: multiple shift (“M-shift”), gradually increasing shift (“GI-shift”), and gradually increasing-then-decreasing shift (“GID-shift”). Table 1 describes each scenario. For example, the multiple shift scenario proceeds as follows: (i) it starts with no covariate shift at the beginning; (ii) after observing 25% target samples (i.e., $250^{th}$ samples for natural shift experiments and $2500^{th}$ samples for synthetic shift experiments), covariate shift is applied to all target samples (with probability 1) by adding random perturbations for synthetic shift and by drawing samples from a target distribution for natural shift; (iii) after 50% of target samples, it reverts to no covariate shift; and (iv) finally after observing 75% target samples, the covariate shift is applied to all target samples. Gradually increasing shift and gradually increasing-then-decreasing shift scenarios start with no covariate shift for the first 20% of target samples; then, covariate shift is applied with some probability $0 < p < 1$ by gradually changing $p$ over time.

**Stream data generation.** For each shift (i.e., natural shift and synthetic shift), we have a source dataset $S$ and target datasets $T_t$, from which we randomly draw source and target samples for each time step $t$. In particular, we consider a batch of samples for computational efficiency, where we denote the batch size by $B$; we use $B = 10$ for our experiments. That is, we wait for $B$ samples to be collected from the target distribution before checking for covariate shift and the updating the source-target classifier; then, we begin collecting the next batch. Finally, we evaluate each approach using multiple random repetitions, which we denote by $R$ (the value of $R$ depends on each experiment).

### 5.2 Dataset.

**Dataset.** First, we consider a natural shift on ImageNet. To construct such a shift, we consider the subset of dog classes; in particular, 120 of the 1000 of the ImageNet classes are of dogs (Khosla et al., 2011). Then, we randomly select half (i.e., 60) of these classes to be the source dataset, and the other half to be the target dataset; thus, the number of source and target images is 2997 each (after removing duplicated images). As a consequence, the source and target datasets correspond to different dog breeds, which is a kind of natural distribution shift.

**Results.** Figure 1 and Table 2 show results for the natural shift experiment with $w = 10$ and $\alpha = 1%$. Figure 1 illustrates detection rates of the four algorithms with $R = 100$ repetitions (i.e., the fraction of repetitions that reported “covariate shift” at each step). Table 2a shows the number
of shifted samples required to reach at least 80% of covariate shift detection rate under the shift. Table 2a shows false positive rate (FPR) after 50, 100, 150, and 200 samples with $R = 20000$ repetitions.

**Discussion.** Figure 1 shows the detection rate of each algorithm as each scenario progresses. In multiple shift (Figure 1a) and gradually increasing-then-decreasing shift (Figure 1c), covariate shift disappears after a certain point, and all algorithms correctly detect this change. However, as shown in Table 2a, our approach always requires fewer samples to detect the shift. While H5 does not achieve 80% detection and H2 and Wald reach 80% only for multiple shift change scenario, our approach always detects covariate shift at a rate higher than 80%. Furthermore, for multiple shift, our algorithm requires fewer than half the number of samples compared to H2. In summary, our algorithm is significantly more sample efficient at detecting covariate shift compared to the baselines, most likely since it utilizes all samples for both training the source-target classifier and constructing the CP interval. For FPR, all algorithms always satisfy the FPR bound (i.e., $\text{FPR} \leq \alpha$).

### 5.3 Synthetic Shift

**Dataset.** Next, we consider a synthetic shift on ImageNet. In particular, we split the original ImageNet validation set into equal sized source and target datasets. To construct the target dataset, we add synthetic perturbations on original images. We (separately) consider five perturbation types from [Hendrycks & Dietterich, 2019]—in particular, Contrast, Defocus Blur, Elastic Transform, Gaussian Blur, and Gaussian Noise, with five different severity levels.

**Results.** The experiment results are shown in Figure 2 and Table 3 for the experiments with the perturbation severity of 2, window size $w = 10$, and significance level $\alpha = 1\%$. Table 3a shows the number of target samples required by each algorithm to detect the first covariate shift in the detection rate of at least 80%. Table 3b shows the false positive rate (FPR) after 500, 1000, 1500 and 2000 samples for each of the three scenarios. Figure 2 shows the detection rates over multiple repetitions.
Figure 2: Detection rate for synthetic shifts with Gaussian noise perturbation, Severity = 2, R = 100, w = 10, $\alpha = 1\%$. The black dashed line indicates shifted sample ratio, i.e., the degree (or probability) of covariate shift. The red dotted line shows the accuracy of ResNet152 on the source and target samples in the given window.

Table 3: Synthetic shift results with (a) Severity = 2, w = 10, $\alpha = 1\%$, and R = 100, and (b) R = 20000. In (a), we bold the best algorithm. In (b), we bold values that exceed the desired $\alpha = 1\%$.

(a) Number of samples for detection

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Alg.</th>
<th>Contrast</th>
<th>Blur</th>
<th>Elastic Transform</th>
<th>Gaussian Blur</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-shift</td>
<td>Ours</td>
<td>230</td>
<td>200</td>
<td>220</td>
<td>210</td>
<td>180</td>
</tr>
<tr>
<td>H2</td>
<td></td>
<td>470</td>
<td>450</td>
<td>450</td>
<td>490</td>
<td>350</td>
</tr>
<tr>
<td>H5</td>
<td></td>
<td>410</td>
<td>410</td>
<td>460</td>
<td>510</td>
<td>-</td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GI-shift</td>
<td>Ours</td>
<td>2100</td>
<td>2060</td>
<td>2090</td>
<td>2070</td>
<td>2080</td>
</tr>
<tr>
<td>H2</td>
<td></td>
<td>4050</td>
<td>3690</td>
<td>4050</td>
<td>4010</td>
<td>3670</td>
</tr>
<tr>
<td>H5</td>
<td></td>
<td>4360</td>
<td>4110</td>
<td>6010</td>
<td>4110</td>
<td>4110</td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GID-shift</td>
<td>Ours</td>
<td>880</td>
<td>560</td>
<td>900</td>
<td>720</td>
<td>610</td>
</tr>
<tr>
<td>H2</td>
<td></td>
<td>2030</td>
<td>2010</td>
<td>2050</td>
<td>2010</td>
<td>2010</td>
</tr>
<tr>
<td>H5</td>
<td></td>
<td>2060</td>
<td>2060</td>
<td>2060</td>
<td>2060</td>
<td>2060</td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(b) FPR (%) at selected time

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Alg.</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-shift</td>
<td>Ours</td>
<td>0.73</td>
<td>0.89</td>
<td>0.90</td>
<td>1.00</td>
</tr>
<tr>
<td>H2</td>
<td></td>
<td>0.40</td>
<td>0.46</td>
<td>0.46</td>
<td>0.50</td>
</tr>
<tr>
<td>H5</td>
<td></td>
<td>0.80</td>
<td>0.73</td>
<td>0.84</td>
<td>0.69</td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>0.07</td>
<td>0.11</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>GI-shift</td>
<td>Ours</td>
<td>0.86</td>
<td>0.89</td>
<td>0.92</td>
<td>0.85</td>
</tr>
<tr>
<td>H2</td>
<td></td>
<td>0.62</td>
<td>0.54</td>
<td>0.60</td>
<td>0.50</td>
</tr>
<tr>
<td>H5</td>
<td></td>
<td>0.74</td>
<td>0.71</td>
<td>0.77</td>
<td>0.73</td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>0.10</td>
<td>0.13</td>
<td>0.07</td>
<td>0.03</td>
</tr>
<tr>
<td>GID-shift</td>
<td>Ours</td>
<td>0.87</td>
<td>0.95</td>
<td>0.92</td>
<td>0.89</td>
</tr>
<tr>
<td>H2</td>
<td></td>
<td>0.51</td>
<td>0.57</td>
<td>0.51</td>
<td>0.53</td>
</tr>
<tr>
<td>H5</td>
<td></td>
<td>0.73</td>
<td>0.75</td>
<td>0.85</td>
<td>0.89</td>
</tr>
<tr>
<td>Wald</td>
<td></td>
<td>0.09</td>
<td>0.08</td>
<td>0.04</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Discussion. As can be seen, our approach outperforms the baselines in terms of sample efficiency for the covariate shift detection as was the case of the natural shift. Our algorithm requires about half as many samples before detecting covariate shift compared to the baselines. In terms of FPR, our approach always satisfies the FPR bound. Finally, Figure 2 shows the accuracy drop with the shifted samples. In particular, the red dotted line shows the accuracy of ResNet152 on the examples in the source and target samples of the given window; as can be seen, the accuracy decreases as the degree of the shift increases. Covariate shift detection can be successfully used to notify a user that an accuracy drop may have occurred.

6 Conclusion

We have proposed a novel covariate shift detection algorithm, which uses a classifier two-sample test to check whether the current test examples differ in distribution compared to the training examples. Our approach ensures sample efficiency by avoiding the need to split the dataset into a training set and a held-out test set, and instead using all the data to both train the source-target discriminator and to evaluate its accuracy. We prove that even with this optimization, our approach provides finite sample guarantees on the false positive rate at a desired level; we also prove bounds on the false negative rate under a mild conditions on the trained classifier. Finally, we empirically demonstrate that our proposed algorithm is significantly more sample efficient compared to a natural baseline that uses a held-out test set in terms of detecting both natural and synthetic shifts on ImageNet.
**Reproducibility Statement.** For our empirical results, we stated our algorithm in Algorithm 1, hyperparameters in Section 5.1, and dataset setups in Section 5.2 and 5.3. We have included the source code in the supplement for reproducing the experimental results. For our theory, we have included all proofs in Appendix A.

**REFERENCES**


A PROOFS

A.1 PROOF OF LEMMA [1]

Figure 3 represents the graphical model over random variables, where observed random variables are colored in gray. We prove the conditional independence using the d-separation (also called the Bayes ball algorithm) [Bishop 2006], which is a set of rules that can determine the conditional dependency between two random variables based on the graphical model and observed random variables. In particular, \( \hat{y}_{i+2} \) is conditionally independent to \( \hat{y}_k \) for all \( k \leq i + 1 \) since the path to \( \hat{y}_k \) is blocked by \( \hat{g}_{i+1} \) (i.e., \( \hat{g}_{i+1} \) is observed). Similarly, \( \hat{y}_{i+2} \) is conditionally independent to \( \hat{y}_k \) for all \( k \geq i + 3 \). This proves the claim.

A.2 PROOF OF LEMMA [2]

For any source-target classifier \( \hat{g}_t \), if \( S = \bar{T}_{w,t} \), the following holds:

\[
\mu_{w,t}^* = \mathbb{P}_{(x,y) \sim Q_{w,t}} [\hat{g}_t(x) = y] \\
= \int \sum_{y \in \{0,1\}} \mathbb{1}(\hat{g}_t(x) = y) Q_{w,t}(x, y) dx \\
= \int \sum_{y \in \{0,1\}} \mathbb{1}(\hat{g}_t(x) = y) \left( \frac{1}{2} \cdot S(x) \cdot \mathbb{1}(y = 0) + \frac{1}{2} \cdot \bar{T}_{w,t}(x) \cdot \mathbb{1}(y = 1) \right) dx \\
= \frac{1}{2} \int \sum_{y \in \{0,1\}} \mathbb{1}(\hat{g}_t(x) = y) S(x) \mathbb{1}(y = 0) + \sum_{y \in \{0,1\}} \mathbb{1}(\hat{g}_t(x) = y) \bar{T}_{w,t}(x) \mathbb{1}(y = 1) dx \\
= \frac{1}{2} \int \mathbb{1}(\hat{g}_t(x) = 0) S(x) + \mathbb{1}(\hat{g}_t(x) = 1) \bar{T}_{w,t}(x) dx \\
= \frac{1}{2} \int \mathbb{1}(\hat{g}_t(x) = 0) S(x) + \mathbb{1}(\hat{g}_t(x) = 1) S(x) dx \\
= \frac{1}{2} \int (\mathbb{1}(\hat{g}_t(x) = 0) + \mathbb{1}(\hat{g}_t(x) = 1)) S(x) dx \\
= \frac{1}{2} \int S(x) dx \\
= \frac{1}{2},
\]

where the sixth equality holds since \( S = \bar{T}_{w,t} \); the claim follows. \( \square \)
A.3 PROOF OF THEOREM[3]

Denote the event that \( P_{(x,y) \sim Q_{w,t}} \{ \hat{g}_t(x) = y \} = 1/2 \) by \( E \), and let \( s_{w,t} = 2w \hat{\mu}_{w,t} \). Then, we have

\[
\begin{align*}
\mathbb{P}_{S_{w,t},T_{w,t}} \left[ f(S_{w,t}, T_{w,t}; \alpha) = 0 \right] &= \mathbb{P}_{S_{w,t},T_{w,t}} \left[ \left( \frac{1}{2} \in \Theta_{CP}(s_{w,t}, 2w; \alpha) \right) \land \left( \mathbb{P}_{x,y} \{ \hat{g}_t(x) = y \} = \frac{1}{2} \right) \right] \\
&= \mathbb{P}_{S_{w,t},T_{w,t}} \left[ E \right] \mathbb{P}_{S_{w,t},T_{w,t}} \left[ \frac{1}{2} \in \Theta_{CP}(s_{w,t}, 2w; \alpha) \ \bigg| \ E \right] \\
&= \mathbb{P}_{S_{w,t},T_{w,t}} \left[ \frac{1}{2} \in \Theta_{CP}(s_{w,t}, 2w; \alpha) \ \bigg| \ E \right] \\
&\geq 1 - \alpha,
\end{align*}
\]

where the first equality holds since \( S = T_{w,t} \) and by Lemma 2, the third equality holds by Lemma 2, and the last inequality holds by the property of the Clopper-Pearson interval and Lemma 1.

A.4 PROOF OF LEMMA[4]

We use the tail bound of the binomial distribution using the Hoeffding’s inequality—\textit{i.e.}

\[
F(s; n, p) \leq \exp \left\{ -2n \left( p - \frac{s}{n} \right)^2 \right\}.
\]

For the upper bound of the upper Clopper-Pearson interval, we have

\[
\sup \left\{ \theta \ \bigg| \ F(s; n, \theta) \geq \frac{\alpha}{2} \right\} \leq \sup \left\{ \theta \ \bigg| \ \exp \left\{ -2n \left( \theta - \frac{s}{n} \right)^2 \right\} \geq \frac{\alpha}{2} \right\} \\
= \sup \left\{ \theta \ \bigg| \ \frac{s}{n} - \sqrt{\frac{\ln \frac{2}{\alpha}}{2n}} \leq \theta \leq \frac{s}{n} + \sqrt{\frac{\ln \frac{2}{\alpha}}{2n}} \right\} \\
= \frac{s}{n} + \sqrt{\frac{\ln \frac{2}{\alpha}}{2n}}. \tag{4}
\]

For the lower bound of the lower Clopper-Pearson interval, we have

\[
\inf \left\{ \theta \ \bigg| \ F(n-s; n, 1-\theta) \geq \frac{\alpha}{2} \right\} \geq \inf \left\{ \theta \ \bigg| \ \exp \left\{ -2n \left( \theta - \frac{s}{n} \right)^2 \right\} \geq \frac{\alpha}{2} \right\} \\
= \inf \left\{ \theta \ \bigg| \ \frac{s}{n} - \sqrt{\frac{\ln \frac{2}{\alpha}}{2n}} \leq \theta \leq \frac{s}{n} + \sqrt{\frac{\ln \frac{2}{\alpha}}{2n}} \right\} \\
= \frac{s}{n} - \sqrt{\frac{\ln \frac{2}{\alpha}}{2n}}. \tag{5}
\]

Finally, (4) and (5) imply the claim.

A.5 PROOF OF THEOREM[5]

Let the lower and upper bound of the Clopper-Pearson interval \( \Theta_{CP} \) be \( \underline{\Theta}_{CP} \) and \( \overline{\Theta}_{CP} \), respectively. Recall that we denote the CDF of a binomial distribution \( \text{binomial}(n, p) \) by \( F(s; n, p) \). Then, we
have
\[
\mathbb{P}_{S_w,T,w} \left[ \hat{f}(S_{w,t},T_{w,t}; \alpha) = 1 \right]
= \mathbb{P}_{S_w,T,w} \left[ \frac{1}{2} \notin \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) \right]
= \mathbb{P}_{S_w,T,w} \left[ \left( \mu^{*}_{w,t} < \frac{1}{2} + \epsilon \lor \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right) \land \left( \frac{1}{2} \notin \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) \right) \right]
= \mathbb{P}_{S_w,T,w} \left[ \left( \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right) \land \left( \frac{1}{2} \notin \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) \right) \right]
= \mathbb{P}_{S_w,T,w} \left[ \mu^{*}_{w,t} \leq \frac{1}{2} + \epsilon \right] \mathbb{P}_{S_w,T,w} \left[ \frac{1}{2} \notin \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
= \mathbb{P}_{S_w,T,w} \left[ \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) > \frac{1}{2} \lor \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) < \frac{1}{2} \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
+ \mathbb{P}_{S_w,T,w} \left[ \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) < \frac{1}{2} \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right],
\]
where (6) and (7) hold due to \( \mathbb{P}_{S_w,T,w} [\mu^{*}_{w,t} < 1/2 + \epsilon] = 0 \) and \( \mathbb{P}_{S_w,T,w} [\mu^{*}_{w,t} \geq 1/2 + \epsilon] = 1 \) from the assumption on \( g_t \) and \( S \neq T \), respectively.

By Lemma 4, the first term is lower bounded as follows:
\[
\mathbb{P}_{S_w,T,w} \left[ \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) > \frac{1}{2} \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
\geq \mathbb{P}_{S_w,T,w} \left[ \hat{\mu}_{w,t} - \sqrt{\frac{\ln \frac{\alpha}{2}}{4w}} > \frac{1}{2} \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
= \mathbb{P}_{S_w,T,w} \left[ 2w\hat{\mu}_{w,t} > a(w, \alpha) \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
= \mathbb{P}_{S_w,T,w} \left[ 2w\hat{\mu}_{w,t} \geq \lceil a(w, \alpha) \rceil + 1 \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
\geq F \left( 2w - \lceil a(w, \alpha) \rceil + 1; 2w, \frac{1}{2} - \epsilon \right),
\]
where the last inequality holds since the binomial parameter \( \frac{1}{2} - \epsilon \) makes the CDF \( F \) smallest.

Similarly, the second term is lower bounded as follows:
\[
\mathbb{P}_{S_w,T,w} \left[ \Theta_{CP}(2w\mu^{*}_{w,t}, 2w; \alpha) < \frac{1}{2} \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
\geq \mathbb{P}_{S_w,T,w} \left[ \hat{\mu}_{w,t} + \sqrt{\frac{\ln \frac{\alpha}{2}}{4w}} < \frac{1}{2} \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
= \mathbb{P}_{S_w,T,w} \left[ 2w\hat{\mu}_{w,t} < b(w, \alpha) \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
= \mathbb{P}_{S_w,T,w} \left[ 2w\hat{\mu}_{w,t} \leq \lceil b(w, \alpha) \rceil - 1 \left| \mu^{*}_{w,t} \geq \frac{1}{2} + \epsilon \right. \right]
\geq F \left( \lceil b(w, \alpha) \rceil - 1; 2w, 1 \right),
\]
where the last inequality holds since the binomial parameter \( \mu^{*}_{w,t} = 1 \) makes the CDF \( F \) smallest.

The claim follows by combining (8) and (9).
B Additional Baseline

We include one additional baseline, adapted version of Deep Kernel MMD (Liu et al., 2020). This Deep Kernel (DK) requires training of the kernel parameters and the network for extracting features. We use half of samples for this training process and conduct a test using the rest of samples as H2. We run this algorithm for Natural shift as in Section 5.2.

Results. The experimental results with DK are shown in Figure 4 and Table 4. As described in Section 5.2, Figure 4 shows the detection rates of each algorithms, and Table 4 presents FPR at selected points. Our approach outperforms this adapted version of Deep kernel MMD approach. First, in terms of the false positive rate (FPR), DK violates the desired FPR bound, while ours satisfies the bound (Table 4b). Furthermore, Table 4c shows that our approach generally requires a smaller number of samples for detection. In the M-shift case, ours needs ten more samples. In the other two shifts, however, ours shows superior performance compared to DK. In the GI-shift, our approach requires 150 fewer samples, and in GID-shift, our approach uses 310 samples for the detection, while DK cannot achieve 80% detection.

C Additional Discussion

C.1 Multiple Epochs in Training

As shown in Algorithm 1, each example is used only once in updating the source-target classifier, baselines also follow this setting in all experiments. We consider this single epoch update anticipating that our algorithms being used in the online setting, where it is infeasible to take multiple passes.
over the training data. However, without consideration of the online setting, each example can be used multiple times during training with the restriction that the example can be used only once in the CP interval, which can improve the performance. As this strategy is orthogonal to our approach, it can be applied to both ours and other baselines expecting the performance improvement.

D ADDITIONAL EXPERIMENTAL RESULTS

D.1 DETECTION RATE

This section shows the additional detection rate plots for the different perturbations, severities, and window sizes (\(w\)) including figures in the main paper.

D.1.1 M-SHIFT

Figure 5 - Figure 19 display the detection rate plot for M-shift scenario with different settings. These all different settings show the similar pattern with the figures in the main paper.
Figure 5: Contrast with $R = 100, w = 10, \alpha = 1\%$. 
Figure 6: Defocus blur with $R = 100, w = 10, \alpha = 1\%$. 
Figure 7: Elastic transform with $R = 100$, $w = 10$, $\alpha = 1\%$. 
Figure 8: Gaussian blur with $R = 100$, $w = 10$, $\alpha = 1\%$. 
Figure 9: Gaussian noise with $R = 100$, $w = 10$, $\alpha = 1\%$. 
Figure 10: Contrast with $R = 100$, $w = 20$, $\alpha = 1\%$. 
Figure 11: Defocus blur with $R = 100$, $w = 20$, $\alpha = 1\%$. 
Figure 12: Elastic transform with $R = 100$, $w = 20$, $\alpha = 1\%$. 
Figure 13: Gaussian blur with $R = 100$, $w = 20$, $\alpha = 1\%$. 
Figure 14: Gaussian noise with $R = 100$, $w = 20$, $\alpha = 1\%$. 
Figure 15: Contrast with $R = 100, w = 50, \alpha = 1\%$. 
Figure 16: Defocus blur with $R = 100, w = 50, \alpha = 1\%$. 
Figure 17: Elastic transform with $R = 100, w = 50, \alpha = 1\%$. 
Figure 18: Gaussian blur with $R = 100$, $w = 50$, $\alpha = 1\%$. 
Figure 19: Gaussian noise with $R = 100$, $w = 50$, $\alpha = 1\%$. 
D.1.2 GI-SHIFT

This section includes the plots for GI-shift scenario with different perturbation, window sizes ($w$), and fixed severity.

Figure 20: GI-shift with $R = 100$, $w = 10$, $\alpha = 1\%$. 
Figure 21: GI-shift with $R = 100$, $w = 20$, $\alpha = 1\%$. 
Figure 22: GI-shift with $R = 100$, $w = 50$, $\alpha = 1\%$. 
D.1.3 GID-SHIFT

Similar to the previous two sections, this section includes figures for the GID-shift scenario with different perturbation with severity 2, and different window sizes ($w$).

Figure 23: GID-shift with $R = 100$, $w = 10$, $\alpha = 1\%$. 
Figure 24: GID-shift with $R = 100$, $w = 20$, $\alpha = 1\%$. 
Figure 25: GID-shift with $R = 100$, $w = 50$, $\alpha = 1\%$. 
D.2 Number of Samples for Detection

This section presents the required number of samples for detecting covariate shift over repetitions (Rate $\geq 80\%$) with different perturbations, severities, and window sizes ($w$).

D.2.1 M-shift

Table 5: Number of samples for detection with $R = 100$, $w = 10$

<table>
<thead>
<tr>
<th>Severity</th>
<th>Algorithms</th>
<th>Contrast</th>
<th>Defocus Blur</th>
<th>Elastic Transform</th>
<th>Gaussian Blur</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ours</td>
<td>280</td>
<td>240</td>
<td>510</td>
<td>400</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>610</td>
<td>550</td>
<td>1310</td>
<td>950</td>
<td>450</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>610</td>
<td>460</td>
<td>1610</td>
<td>960</td>
<td>410</td>
</tr>
<tr>
<td>2</td>
<td>Ours</td>
<td>230</td>
<td>200</td>
<td>220</td>
<td>210</td>
<td>180</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>470</td>
<td>450</td>
<td>450</td>
<td>490</td>
<td>350</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>410</td>
<td>410</td>
<td>410</td>
<td>460</td>
<td>310</td>
</tr>
<tr>
<td>3</td>
<td>Ours</td>
<td>190</td>
<td>170</td>
<td>190</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>350</td>
<td>370</td>
<td>410</td>
<td>270</td>
<td>270</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>310</td>
<td>360</td>
<td>360</td>
<td>260</td>
<td>260</td>
</tr>
<tr>
<td>4</td>
<td>Ours</td>
<td>150</td>
<td>140</td>
<td>160</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>250</td>
<td>290</td>
<td>290</td>
<td>210</td>
<td>210</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>210</td>
<td>260</td>
<td>260</td>
<td>210</td>
<td>160</td>
</tr>
<tr>
<td>5</td>
<td>Ours</td>
<td>130</td>
<td>120</td>
<td>140</td>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>210</td>
<td>230</td>
<td>210</td>
<td>170</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>160</td>
<td>210</td>
<td>210</td>
<td>160</td>
<td>160</td>
</tr>
</tbody>
</table>

Table 6: Number of samples for detection with $R = 100$, $w = 20$

<table>
<thead>
<tr>
<th>Severity</th>
<th>Algorithms</th>
<th>Contrast</th>
<th>Defocus Blur</th>
<th>Elastic Transform</th>
<th>Gaussian Blur</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ours</td>
<td>300</td>
<td>250</td>
<td>430</td>
<td>360</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>530</td>
<td>430</td>
<td>970</td>
<td>770</td>
<td>410</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>460</td>
<td>410</td>
<td>1010</td>
<td>710</td>
<td>360</td>
</tr>
<tr>
<td>2</td>
<td>Ours</td>
<td>250</td>
<td>220</td>
<td>240</td>
<td>230</td>
<td>210</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>430</td>
<td>390</td>
<td>390</td>
<td>410</td>
<td>350</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>360</td>
<td>360</td>
<td>360</td>
<td>360</td>
<td>310</td>
</tr>
<tr>
<td>3</td>
<td>Ours</td>
<td>220</td>
<td>200</td>
<td>230</td>
<td>170</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>330</td>
<td>330</td>
<td>370</td>
<td>290</td>
<td>270</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>310</td>
<td>310</td>
<td>360</td>
<td>260</td>
<td>260</td>
</tr>
<tr>
<td>4</td>
<td>Ours</td>
<td>170</td>
<td>170</td>
<td>200</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>270</td>
<td>290</td>
<td>310</td>
<td>230</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>260</td>
<td>260</td>
<td>260</td>
<td>210</td>
<td>210</td>
</tr>
<tr>
<td>5</td>
<td>Ours</td>
<td>160</td>
<td>150</td>
<td>170</td>
<td>110</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>250</td>
<td>250</td>
<td>230</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>210</td>
<td>210</td>
<td>210</td>
<td>160</td>
<td>160</td>
</tr>
</tbody>
</table>
Table 7: Number of samples for detection with $R = 100$, $w = 50$

<table>
<thead>
<tr>
<th>Severity</th>
<th>Algorithms</th>
<th>Contrast</th>
<th>Defocus Blur</th>
<th>Elastic Transform</th>
<th>Gaussian Blur</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ours</td>
<td>360</td>
<td>300</td>
<td>490</td>
<td>430</td>
<td>290</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>570</td>
<td>510</td>
<td>870</td>
<td>710</td>
<td>470</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>560</td>
<td>460</td>
<td>810</td>
<td>660</td>
<td>410</td>
</tr>
<tr>
<td>2</td>
<td>Ours</td>
<td>310</td>
<td>270</td>
<td>280</td>
<td>280</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>510</td>
<td>450</td>
<td>470</td>
<td>470</td>
<td>390</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>460</td>
<td>410</td>
<td>460</td>
<td>460</td>
<td>360</td>
</tr>
<tr>
<td>3</td>
<td>Ours</td>
<td>260</td>
<td>240</td>
<td>270</td>
<td>220</td>
<td>210</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>410</td>
<td>410</td>
<td>450</td>
<td>330</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>360</td>
<td>360</td>
<td>410</td>
<td>310</td>
<td>310</td>
</tr>
<tr>
<td>4</td>
<td>Ours</td>
<td>210</td>
<td>210</td>
<td>230</td>
<td>170</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>310</td>
<td>370</td>
<td>370</td>
<td>270</td>
<td>270</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>310</td>
<td>360</td>
<td>360</td>
<td>260</td>
<td>260</td>
</tr>
<tr>
<td>5</td>
<td>Ours</td>
<td>170</td>
<td>170</td>
<td>190</td>
<td>130</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>H2</td>
<td>290</td>
<td>310</td>
<td>290</td>
<td>210</td>
<td>210</td>
</tr>
<tr>
<td></td>
<td>H5</td>
<td>260</td>
<td>260</td>
<td>260</td>
<td>210</td>
<td>210</td>
</tr>
</tbody>
</table>
### D.2.2 GI-SHIFT

#### Table 8: Number of samples for detection with $R = 100, w = 20$

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Contrast</th>
<th>Defocus Blur</th>
<th>Elastic Transform</th>
<th>Gaussian Blur</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>2100</td>
<td>2040</td>
<td>2100</td>
<td>2040</td>
<td>2050</td>
</tr>
<tr>
<td>H2</td>
<td>2890</td>
<td>2150</td>
<td>2610</td>
<td>2170</td>
<td>2190</td>
</tr>
<tr>
<td>H5</td>
<td>4110</td>
<td>4010</td>
<td>4110</td>
<td>4010</td>
<td>4010</td>
</tr>
</tbody>
</table>

#### Table 9: Number of samples for detection with $R = 100, w = 50$

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Contrast</th>
<th>Defocus Blur</th>
<th>Elastic Transform</th>
<th>Gaussian Blur</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>1580</td>
<td>980</td>
<td>1340</td>
<td>1170</td>
<td>970</td>
</tr>
<tr>
<td>H2</td>
<td>2310</td>
<td>2130</td>
<td>2190</td>
<td>2170</td>
<td>2130</td>
</tr>
<tr>
<td>H5</td>
<td>2460</td>
<td>2310</td>
<td>2510</td>
<td>2310</td>
<td>2310</td>
</tr>
</tbody>
</table>

### D.2.3 GID-SHIFT

#### Table 10: Number of samples for detection with $R = 100, w = 20$

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Contrast</th>
<th>Defocus Blur</th>
<th>Elastic Transform</th>
<th>Gaussian Blur</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>620</td>
<td>430</td>
<td>530</td>
<td>470</td>
<td>450</td>
</tr>
<tr>
<td>H2</td>
<td>1890</td>
<td>990</td>
<td>1670</td>
<td>1170</td>
<td>1030</td>
</tr>
<tr>
<td>H5</td>
<td>2010</td>
<td>2010</td>
<td>2060</td>
<td>2010</td>
<td>2010</td>
</tr>
</tbody>
</table>

#### Table 11: Number of samples for detection with $R = 100, w = 50$

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Contrast</th>
<th>Defocus Blur</th>
<th>Elastic Transform</th>
<th>Gaussian Blur</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>590</td>
<td>480</td>
<td>550</td>
<td>520</td>
<td>490</td>
</tr>
<tr>
<td>H2</td>
<td>1110</td>
<td>870</td>
<td>1010</td>
<td>970</td>
<td>910</td>
</tr>
<tr>
<td>H5</td>
<td>1510</td>
<td>910</td>
<td>1310</td>
<td>1010</td>
<td>960</td>
</tr>
</tbody>
</table>