# Invariant Anomaly Detection under Distribution Shifts: A Causal Perspective (Appendix)

**João B. S. Carvalho, Mengtao Zhang, Robin Geyer, Carlos Cotrini, Joachim M. Buhmann**
Institute for Machine Learning
Department of Computer Science
ETH Zürich
`{joao.carvalho, mengtao.zhang, robin.geyer, ccarlos, jbuhmann}@inf.ethz.ch`

## 1 Theoretical Details

### 1.1 Measurability

Let $\mathcal{B}(\mathbb{R}^n)$ be the Borel $\sigma$-algebra over $\mathbb{R}^n$, for some $n \in \mathbb{N}$.

**Definition 1.** The $\sigma$-algebra generated by a random variable $Y$ is $\sigma(Y) = \left\{Y^{-1}(B) : B \in \mathcal{B}(\mathbb{R}^n)\right\}$.

**Definition 2.** A random variable $X$ is $Y$-measurable if $\sigma(X) \subseteq \sigma(Y)$.

Intuitively, the $\sigma$-algebra generated by $Y$ describes all events that $Y$ "can express" and can be measured in probability. If $X$ is $Y$-measurable, that means that $Y$ can express all what $X$ can express.

### 1.2 D-separation and Confounding

We provide here a brief overview on d-separation and confounding and refer the reader to Bishop and Nasrabadi [2006] for details.

**Definition 3.** Two random variables in a Bayesian network are confounded if they share a latent parent.

**Definition 4.** A *path* is a sequence of random variables $(V_1, \ldots, V_n)$ in a Bayesian network, where $V_i$ is a parent or child of $V_{i+1}$, for $i < n$. For $1 < i < n$, the variable $V_i$ is a *collision* if $V_i$ is a child of both $V_{i-1}$ and $V_{i+1}$.

**Definition 5.** A path is *blocked* if $V_1$ and $V_n$ are not observed and at least one of following holds:

- For some collision $V$ in the path, neither $V$ nor any of its descendants is observed.
- For some variable $V$ in the path that is not a collision, $V$ is observed.

We now recall the d-separation principle.

**Lemma 1.** Let $W$ be a set of observed variables and $V_1$ and $V_2$ two variables not observed. If any path from $V_1$ to $V_2$ is blocked, then $V_1 \perp V_2 \mid W$.

### 1.3 Detailed Proof of Theorem 1

We repeat in Figure 1 the causal graph for anomaly detection here for convenience. To prove Theorem 1, we use the following lemma.

**Lemma 2.** If $W$ and $E$ are confounded, then $X_a \perp E \mid W$. Otherwise, $X_a \perp E$.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

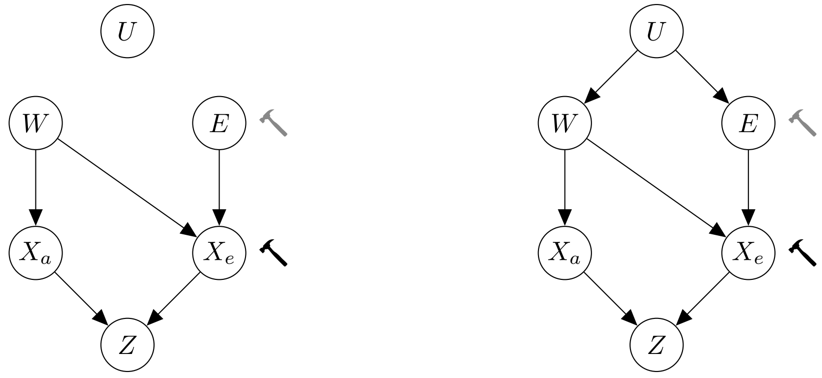


Figure 1: Causal graphs for anomaly detection. The left figure shows the case of no confounding. The right figure shows the case of confounding. An intervention at the $E$ variable induces a domain shift (gray hammer), whereas an intervention at the $X_e$ variable induces a covariate shift (black hammer).

*Proof.* We prove this via d-separation. We start with the case of $W$ and $E$ not being confounded. Note that there are two paths from $X_a$ to $E$. One via $W$ and the other via $Z$. The path via $W$ is blocked, because $X_e$ is a collision and neither $X_e$ nor its descendant $Z$ are observed. The path via $Z$ is blocked, because $Z$ is a collision with no descendants and is not observed. Hence, $X_a \perp E$, when $W$ and $E$ are not confounded. In the case of $W$ and $E$ being confounded, we can show analogously that $X_a \perp E \mid W$. Note that there are three paths from $X_a$ to $E$. The first one is via $U$, but this is blocked, because $W$ is in that path, it is not a collision, and it is observed. The second one is via $W$ and $X_e$, but $W$ is in that path and it is not a collision there, so that path is also blocked. The third one is via $Z$, but $Z$ is a collision in that path with no descendants and it is not observed, so it is also blocked. Since all paths are blocked, when $W$ is observed, we conclude that $X_a \perp E \mid W$, when $W$ and $E$ are confounded. ☐

**Theorem 1.** *Suppose that $f$ learns invariant representations. If $W$ and $E$ are confounded, then $Z \perp E \mid W$. Otherwise, $Z \perp E$.*

*Proof.* Recall that if $f$ learns invariant representations, then we assume it to be $X_a$-measurable. This assumption is justified in Veitch et al. [2021]. As a result, since $Z = f(X_a, X_b)$, we have that $Z$ is $X_a$-measurable.

We assume that $W$ and $E$ are not confounded and show that $Z \perp E$ by proving that $p(Z \in A, E \in B) = p(Z \in A)p(E \in B)$, for any $A, B \in \mathcal{B}(\mathbb{R}^n)$. Note that $p(Z \in A, E \in B) = p\left(Z^{-1}(A) \cap E^{-1}(B)\right) = p\left(X_a^{-1}(C_A) \cap E^{-1}(B)\right)$, for some Borel set $C_A$. The last equality follows from $Z$ being $X_a$-measurable, which implies that $Z^{-1}(A) = X_a^{-1}(C_A)$, for some Borel set $C_A$, by Definition 2. By Lemma 2, we have that $X_a \perp E$. This implies that $p\left(X_a^{-1}(C_A) \cap E^{-1}(B)\right) = p\left(X_a^{-1}(C_A)\right) p\left(E^{-1}(B)\right) = p\left(Z^{-1}(A)\right) p\left(E^{-1}(B)\right) = p(Z \in A)p(E \in B)$, which is what we wanted to show.

We now assume that $W$ and $E$ are confounded and show that $Z \perp E \mid W$ by proving that

$$p(Z \in A, E \in B \mid W \in C) = p(Z \in A \mid W \in C)p(E \in B \mid W \in C),$$

for any $A, B, C \in \mathcal{B}(\mathbb{R}^n)$. By an analogous argument, we can show that $p(Z \in A, E \in B \mid W \in C) = p\left(X_a^{-1}(C_A) \cap E^{-1}(B) \mid W^{-1}(C)\right)$. By Lemma 2, we have that $p\left(X_a^{-1}(C_A) \mid W^{-1}(C)\right) p\left(E^{-1}(B) \mid W^{-1}(C)\right)$. With arguments similar to those above, we can show that the last expression is equal to $p(Z \in A \mid W \in C)p(E \in B \mid W \in C)$, which is what we wanted to show. ☐

## 2 Dataset Details

### 2.1 Camelyon17

Our realistic anomaly detection dataset was derived from the Camelyon17 dataset (Koh et al. [2021], Bandi et al. [2018]), and contains $3 \times 96 \times 96$ patches of whole-slide images of lymph node sections sourced from patients who may have metastatic breast cancer. This dataset encompasses tissue patches obtained from five different hospitals. The objective here is to accurately predict the presence of tumor tissue within the patches drawn from hospitals that were not part of the training data. Prior work has shown that differences in staining between hospitals are the primary source of variation in this dataset, however, other divergent factors in the sampling distribution include different acquisition protocols and patient populations (Tellez et al. [2019]).

The in-distribution data was comprised of $151,280$ images evenly distributed across three hospitals, or $100,810$ images evenly distributed across two hospitals depending on the training setting. The other out-of-distribution data covered two additional datasets, the first with $34,904$ patches, and the second with $85,054$ patches. Note that to adapt this dataset to the anomaly detection setting, only normal images were included in the in-distribution training data.

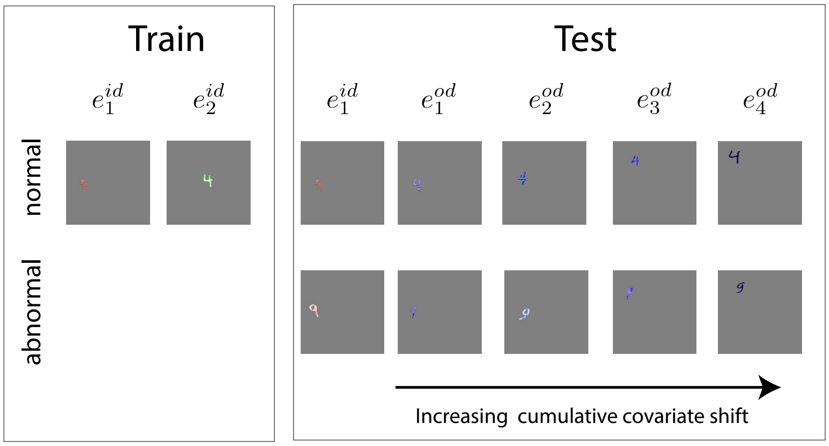


Figure 2: Illustration of our experimental setup for the synthetic covariate shift experiment. The image demonstrates representative examples of training data from two distinct environments, alongside instances of normal and abnormal test data subject to progressively accumulated covariate shifts. This configuration embodies the nuanced challenge of identifying subtle, yet potentially consequential, changes in the data distribution.

## 2.2 Synthetic datasets

The synthetic datasets employed in this study were derived from the DiagViB-6 benchmark (Eulig et al. [2021]). This benchmark uniquely allows for the manipulation of five independent generative factors from colored images: overlaid texture, object size, object position, lightness, and saturation, in addition to the semantic features that correspond to the label. Our synthetic experiments utilized two datasets: MNIST (Deng [2012]) and Fashion-MNIST (Xiao et al. [2017]). All images in both datasets were upsampled to dimensions of $3 \times 256 \times 256$. Initially, we generate two unique and distinct environments specifically designed for the training data. Our primary goal during this stage was to guarantee that all these factors exhibited noticeable differences when compared across the two generated environments. Following the generation of these training environments, we proceeded to develop another pair of environments. These new environments were crafted for the validation data. To ensure consistency, these validation environments were fashioned in such a way that they closely mirrored or replicated the factor configuration that was present in the initial training environments, thus retaining an in-distribution setting.

In the final step, six additional environments, denoted as $e_0, e_1, ..., e_5$, were generated. Each environment $e_i$ consists of images in which $i$ factors have been altered with respect to $e_0$. For a depiction of the samples for these different environments, please refer to Fig. 2.

In devising our evaluation setup, we opted for inducing covariate shifts that are minor deviations from the original in-distribution environments. This decision was motivated by our goal to simulate subtle yet potentially detrimental covariate shifts, particularly in comparison to the challenge of differentiating normal from abnormal.

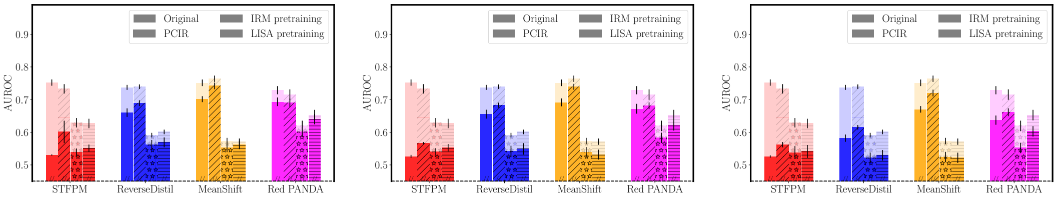
A description of the accumulated covariate shifts in the test environments, $e_0, e_1, e_2, e_3, e_4$, is provided in Table 1.

| $i$ | Chosen factor in $e_i$ |
|---|---|
| 0 | None |
| 1 | Hue |
| 2 | Texture |
| 3 | Lightness |
| 4 | Position |

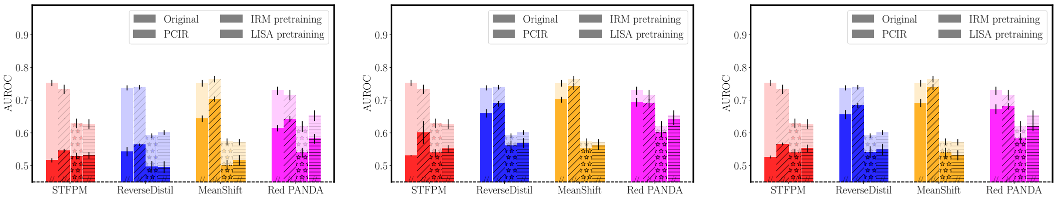
Table 1: Environments $e_0, \ldots, e_4$ used in our synthetic benchmark. For $0 < i \leq 4$, the environment $e_i$ modifies the new factor indicated in the table in addition to the factors modified by $e_0, \ldots, e_{i-1}$.

## 3 Invariantly pretrained encoders

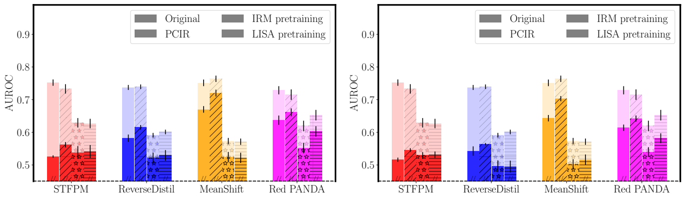
We also extend our experimental evaluation by adding a comparison to Smeu et al. [2022], an environment-aware framework for AD that pretrains the encoder of the AD model using an invariance-inducing method (LISA or IRM). We evaluate this method in MNIST and F-MNIST subject to targeted covariate shifts (the exact same setup as seen in our original experiments). The method was incorporated into all baselines and compared to the same baselines while regularized through partial conditional invariance. We show the results of these experiments in Fig. 3. Across all methods tested, and in both datasets, we observe a sharp decrease in performance when compared to a baseline non-invariant method and that it provides less robustness to covariate shifts than our proposed methodology.



(a) $e_1$:MNIST s.t. one covariate shift (b) $e_2$:MNIST s.t. two covariate shifts (c) $e_3$:MNIST s.t three covariate shifts



(d) $e_4$:MNIST s.t. four covariate shifts (e) $e_2$:F-MNIST s.t. one covariate shift (f) $e_2$:F-MNIST s.t. two covariate shift

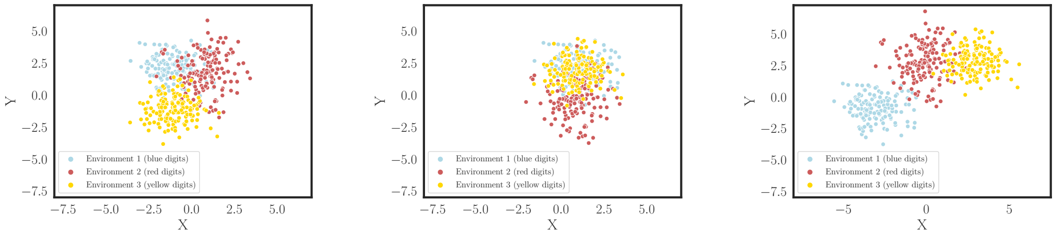


(g) $e_3$:F-MNIST s.t. three covariate shifts (h) $e_4$:F-MNIST s.t. four covariate shifts

Figure 3: Experimental results MNIST and Fashion-MNIST with additional invariant pretraining following. (**background transparent bar-plots:** in-distribution evaluation; **foreground opaque bar-plots:** out-of-distribution evaluation). **(a-d)** Results in MNIST. **(e-h)** Results in Fashion-MNIST.

## 4 Visualization of Invariance *vs* Informativeness

In Fig. 4 we plot the two-dimensional representation of the final layer of a model trained through MeanShift(Reiss and Hoshen [2021]) at different levels of PCIR regularization. The embeddings are obtained through t-SNE. From the progressive increase in the weight of the PCIR term, we see the increased superimposition of the different environments leading to more invariance at the loss of informativeness in the representation.



(a) Invariant and informative (b) Invariant but non-informative (c) Informative but non-invariant

Figure 4: TSNE embeddings of MNIST with three background colors for the digits 4 and 9. The model used was MeanShift subject to different degrees of partial conditional invariant regularization. **(a)** PCIR term set to 5 **(b)** PCIR term set to 150. **(c)** PCIR term set to 0.

## 5 Tables of Results

| | **In dist.** | | **Out of dist.** | | |
|---|---|---|---|---|---|
| | $e_1$ (↑) AUROC | $e_2$ (↑) AUROC | $e_3$ (↑) AUROC | $e_4$ (↑) AUROC | $e_5$ (↑) AUROC |
| STFPM | $0.850 \pm 0.005$ | $0.883 \pm 0.014$ | $0.815 \pm 0.024$ | $0.753 \pm 0.007$ | $0.745 \pm 0.005$ |
| STFPM (PCIR) | $0.868 \pm 0.007$ | $0.873 \pm 0.008$ | $0.814 \pm 0.014$ | $0.798 \pm 0.011$ | $0.774 \pm 0.010$ |
| ReverseDistil | $0.853 \pm 0.009$ | $0.884 \pm 0.014$ | $0.797 \pm 0.015$ | $0.716 \pm 0.009$ | $0.723 \pm 0.004$ |
| ReverseDistil (PCIR) | $0.866 \pm 0.007$ | $0.903 \pm 0.007$ | $0.857 \pm 0.010$ | $0.744 \pm 0.021$ | $0.718 \pm 0.010$ |
| CFA | $0.625 \pm 0.002$ | $0.689 \pm 0.008$ | $0.566 \pm 0.014$ | $0.454 \pm 0.005$ | $0.590 \pm 0.017$ |
| CFA (PCIR) | $0.625 \pm 0.003$ | $0.700 \pm 0.007$ | $0.597 \pm 0.011$ | $0.473 \pm 0.009$ | $0.595 \pm 0.027$ |
| MeanShift | $0.751 \pm 0.012$ | $0.761 \pm 0.014$ | $0.731 \pm 0.015$ | $0.703 \pm 0.009$ | $0.701 \pm 0.010$ |
| MeanShift (PCIR) | $0.781 \pm 0.014$ | $0.791 \pm 0.014$ | $0.778 \pm 0.012$ | $0.772 \pm 0.013$ | $0.770 \pm 0.011$ |
| CSI | $0.601 \pm 0.021$ | $0.600 \pm 0.017$ | $0.582 \pm 0.014$ | $0.491 \pm 0.015$ | $0.531 \pm 0.014$ |
| CSI (PCIR) | $0.648 \pm 0.022$ | $0.652 \pm 0.021$ | $0.621 \pm 0.020$ | $0.581 \pm 0.018$ | $0.600 \pm 0.023$ |
| Red PANDA | $0.761 \pm 0.017$ | $0.764 \pm 0.015$ | $0.671 \pm 0.015$ | $0.641 \pm 0.015$ | $0.651 \pm 0.012$ |
| Red PANDA (PCIR) | $0.761 \pm 0.017$ | $0.769 \pm 0.015$ | $0.742 \pm 0.019$ | $0.731 \pm 0.015$ | $0.740 \pm 0.015$ |

Table 2: Experimental results on realist domain shift in the **Camelyon17** dataset, for both regularized and unregularized models trained on two environments. Results are presented over in-distribution evaluation (environments $e_1$ and $e_2$) and over out-of-distribution (environments $e_3$, $e_4$, and $e_5$).

| | **In dist.** | | | **Out of dist.** | |
|---|---|---|---|---|---|
| | $e_1$ (↑) AUROC | $e_2$ (↑) AUROC | $e_3$ (↑) AUROC | $e_4$ (↑) AUROC | $e_5$ (↑) AUROC |
| STFPM | 0.854 ± 0.011 | 0.873 ± 0.004 | 0.782 ± 0.013 | 0.771 ± 0.013 | 0.735 ± 0.008 |
| STFPM (PCIR) | 0.865 ± 0.006 | 0.873 ± 0.005 | 0.796 ± 0.014 | 0.782 ± 0.012 | 0.759 ± 0.009 |
| ReverseDistil | 0.830 ± 0.028 | 0.866 ± 0.007 | 0.781 ± 0.004 | 0.707 ± 0.023 | 0.710 ± 0.033 |
| ReverseDistil (PCIR) | 0.843 ± 0.009 | 0.880 ± 0.015 | 0.799 ± 0.004 | 0.714 ± 0.010 | 0.715 ± 0.009 |
| CFA | 0.667 ± 0.004 | 0.708 ± 0.011 | 0.628 ± 0.019 | 0.559 ± 0.005 | 0.643 ± 0.006 |
| CFA (PCIR) | 0.667 ± 0.014 | 0.705 ± 0.019 | 0.623 ± 0.004 | 0.561 ± 0.003 | 0.644 ± 0.003 |
| MeanShift | 0.742 ± 0.014 | 0.741 ± 0.013 | 0.681 ± 0.014 | 0.739 ± 0.012 | 0.690 ± 0.013 |
| MeanShift (PCIR) | 0.772 ± 0.013 | 0.784 ± 0.012 | 0.739 ± 0.018 | 0.747 ± 0.013 | 0.731 ± 0.013 |
| CSI | 0.620 ± 0.015 | 0.636 ± 0.014 | 0.548 ± 0.013 | 0.571 ± 0.012 | 0.541 ± 0.015 |
| CSI (PCIR) | 0.650 ± 0.019 | 0.650 ± 0.010 | 0.613 ± 0.015 | 0.624 ± 0.014 | 0.611 ± 0.010 |
| Red PANDA | 0.751 ± 0.012 | 0.742 ± 0.013 | 0.651 ± 0.010 | 0.751 ± 0.010 | 0.641 ± 0.011 |
| Red PANDA (PCIR) | 0.767 ± 0.016 | 0.781 ± 0.014 | 0.761 ± 0.013 | 0.770 ± 0.012 | 0.751 ± 0.009 |

Table 3: Experimental results on realist domain shift in the **Camelyon17** dataset, for both regularized and unregularized models trained on three environments. Results are presented over in-distribution evaluation (environments $e_1$, $e_2$, $e_3$) and over out-of-distribution (environments $e_4$, and $e_5$).

| | **In dist.** (↑) AUROC | **Out of dist.** (↑) AUROC |
|---|---|---|
| STFPM | 0.699 ± 0.025 | 0.630 ± 0.025 |
| STFPM (PCIR) | 0.724 ± 0.020 | 0.698 ± 0.023 |
| ReverseDistil | 0.673 ± 0.057 | 0.617 ± 0.032 |
| ReverseDistil (PCIR) | 0.734 ± 0.013 | 0.723 ± 0.013 |
| CFA | 0.752 ± 0.003 | 0.705 ± 0.018 |
| CFA (PCIR) | 0.785 ± 0.003 | 0.759 ± 0.005 |
| MeanShift | 0.683 ± 0.041 | 0.629 ± 0.043 |
| MeanShift (PCIR) | 0.731 ± 0.022 | 0.726 ± 0.052 |
| CSI | 0.671 ± 0.026 | 0.626 ± 0.024 |
| CSI (PCIR) | 0.692 ± 0.017 | 0.674 ± 0.019 |
| Red PANDA | 0.732 ± 0.026 | 0.691 ± 0.031 |
| Red PANDA (PCIR) | 0.742 ± 0.029 | 0.721 ± 0.012 |

Table 4: Experimental results on realist shortcut learning in the **Waterbirds** dataset, for both regularized and unregularized. Results are presented over in-distribution evaluation and over out-of-distribution.

| | **In dist.** (↑) AUROC | **1 cov. shift** (↑) AUROC | **2 cov. shifts** (↑) AUROC | **3 cov. shifts** (↑) AUROC | **4 cov. shifts** (↑) AUROC |
|---|---|---|---|---|---|
| STFPM | 0.850 ± 0.005 | 0.883 ± 0.014 | 0.815 ± 0.024 | 0.753 ± 0.007 | 0.745 ± 0.005 |
| STFPM (PCIR) | 0.868 ± 0.007 | 0.873 ± 0.008 | 0.814 ± 0.014 | 0.798 ± 0.011 | 0.774 ± 0.010 |
| ReverseDistil | 0.853 ± 0.009 | 0.884 ± 0.014 | 0.797 ± 0.015 | 0.716 ± 0.009 | 0.723 ± 0.004 |
| ReverseDistil (PCIR) | 0.866 ± 0.007 | 0.903 ± 0.007 | 0.857 ± 0.010 | 0.744 ± 0.021 | 0.718 ± 0.010 |
| CFA | 0.625 ± 0.002 | 0.689 ± 0.008 | 0.566 ± 0.014 | 0.454 ± 0.005 | 0.590 ± 0.017 |
| CFA (PCIR) | 0.625 ± 0.003 | 0.700 ± 0.007 | 0.597 ± 0.011 | 0.473 ± 0.009 | 0.595 ± 0.027 |
| MeanShift | 0.751 ± 0.012 | 0.761 ± 0.014 | 0.731 ± 0.015 | 0.703 ± 0.009 | 0.701 ± 0.010 |
| MeanShift (PCIR) | 0.781 ± 0.014 | 0.791 ± 0.014 | 0.778 ± 0.012 | 0.772 ± 0.013 | 0.770 ± 0.011 |
| CSI | 0.601 ± 0.021 | 0.600 ± 0.017 | 0.582 ± 0.014 | 0.491 ± 0.015 | 0.531 ± 0.014 |
| CSI (PCIR) | 0.648 ± 0.022 | 0.652 ± 0.021 | 0.621 ± 0.020 | 0.581 ± 0.018 | 0.600 ± 0.023 |
| Red PANDA | 0.761 ± 0.017 | 0.764 ± 0.015 | 0.671 ± 0.015 | 0.641 ± 0.015 | 0.651 ± 0.012 |
| Red PANDA (PCIR) | 0.761 ± 0.017 | 0.769 ± 0.015 | 0.742 ± 0.019 | 0.731 ± 0.015 | 0.740 ± 0.015 |

Table 5: Experimental results on synthetic covariate shift over the **MNIST** dataset, for both regularized and unregularized models. Results are presented over in-distribution evaluation, and test sets subject to one to four different covariate shifts, as portrayed in Fig. 2

.

|  | **In dist.**<br>(↑) AUROC | **1 cov. shift**<br>(↑) AUROC | **2 cov. shifts**<br>(↑) AUROC | **3 cov. shifts**<br>(↑) AUROC | **4 cov. shifts**<br>(↑) AUROC |
|---|---|---|---|---|---|
| STFPM | 0.752 ± 0.010 | 0.530 ± 0.002 | 0.526 ± 0.004 | 0.526 ± 0.004 | 0.516 ± 0.007 |
| STFPM (PCIR) | 0.733 ± 0.014 | 0.601 ± 0.034 | 0.566 ± 0.004 | 0.562 ± 0.008 | 0.546 ± 0.006 |
| ReverseDistil | 0.737 ± 0.008 | 0.660 ± 0.013 | 0.655 ± 0.014 | 0.582 ± 0.011 | 0.543 ± 0.014 |
| ReverseDistil (PCIR) | 0.740 ± 0.007 | 0.689 ± 0.009 | 0.683 ± 0.008 | 0.615 ± 0.007 | 0.564 ± 0.004 |
| MeanShift | 0.751 ± 0.010 | 0.701 ± 0.009 | 0.692 ± 0.012 | 0.670 ± 0.010 | 0.643 ± 0.010 |
| MeanShift (PCIR) | 0.764 ± 0.010 | 0.742 ± 0.010 | 0.739 ± 0.009 | 0.720 ± 0.010 | 0.703 ± 0.008 |
| CSI | 0.691 ± 0.016 | 0.629 ± 0.011 | 0.571 ± 0.014 | 0.550 ± 0.012 | 0.542 ± 0.014 |
| CSI (PCIR) | 0.681 ± 0.014 | 0.661 ± 0.016 | 0.639 ± 0.014 | 0.611 ± 0.015 | 0.599 ± 0.011 |
| Red PANDA | 0.729 ± 0.013 | 0.693 ± 0.013 | 0.672 ± 0.015 | 0.637 ± 0.014 | 0.614 ± 0.010 |
| Red PANDA (PCIR) | 0.715 ± 0.016 | 0.690 ± 0.013 | 0.681 ± 0.010 | 0.661 ± 0.012 | 0.642 ± 0.009 |

Table 6: Experimental results on synthetic covariate shift over the **Fashion-MNIST** dataset, for both regularized and unregularized models. Results are presented over in-distribution evaluation, and test sets subject to one to four different covariate shifts, as portrayed in Fig. 2

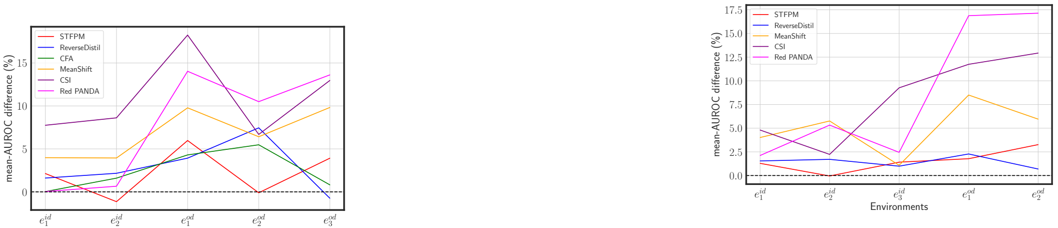
## 6 Performance Gained Compared to Baseline

Our investigation of the influence of partially conditional regularization on model performance is further expanded in Fig.5. This figure presents the percentile difference in mean-AUROC (Area Under the Receiver Operating Characteristic) between partially conditionally regularized and unregularized models.

By comparing each regularized model to its unregularized equivalent across all the tested environments, we have been able to observe a consistent improvement in performance when regularization is applied. This observation holds true across all models and out-of-distribution environments studied.

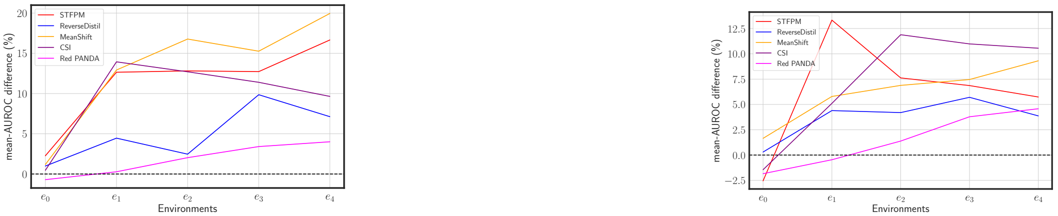
The increase in performance due to regularization varies from $2.5\%$ to as substantial as $20\%$ in some models. This variability implies a model-specific dependency on the degree of invariance that can be induced. Crucially, however, regularization substantially bolsters out-of-distribution performance without any additional training cost, reinforcing its merit as an effective strategy for developing robust anomaly detectors.

To further highlight this point and for the sack of completeness, we also present in Fig. 6, Fig. 7 and Fig. 8each regularized model compared to its baseline.



(a) Camelyon17: two in-distribution

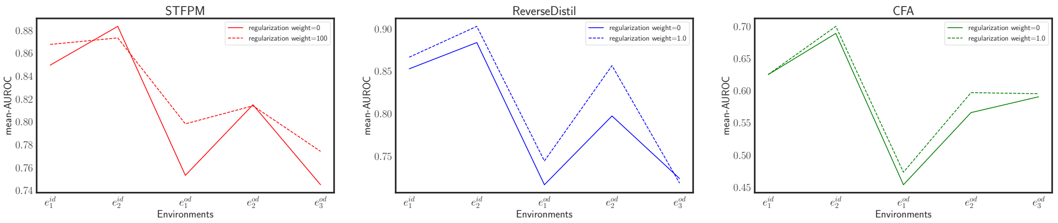
(b) Camelyon17: three in-distribution



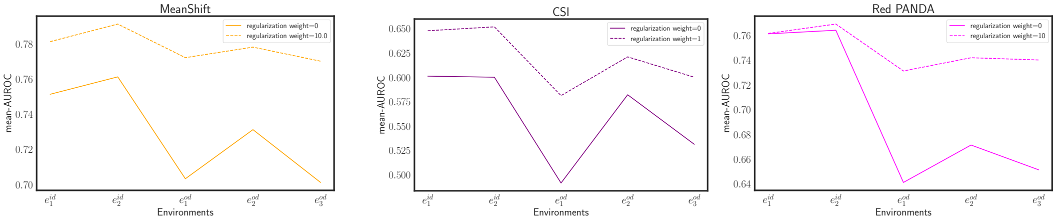
(c) MNIST

(d) Fashion-MNIST

Figure 5: Percentual performance gain over each regularized model when compared to unreglarized baseline.

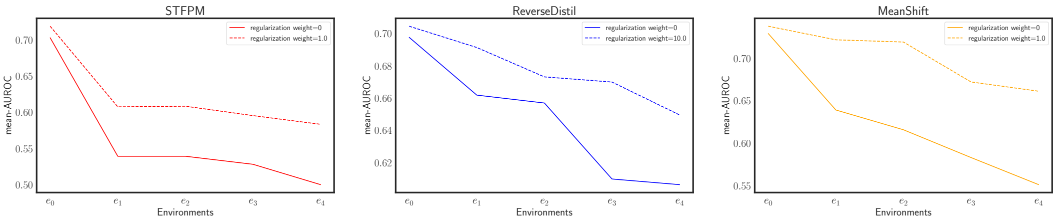


(a) Camelyon17: STFPM (b) Camelyon17: ReverseDistil (c) Camelyon17: CFA

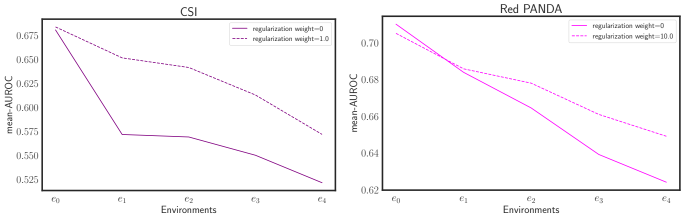


(d) Camelyon17: MeanShift (e) Camelyon17: CSI (f) Camelyon17: Red PANDA

Figure 6: Mean-AUROC curve of each anomaly detector and its regularized version in the Camlyon17 dataset.



(a) MNIST: STFPM (b) MNIST: ReverseDistil (c) MNIST: MeanShift



(d) MNIST: CSI (e) MNIST: Red PANDA

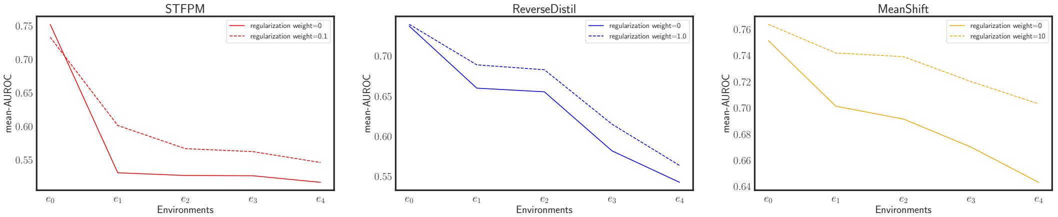
Figure 7: Mean-AUROC curve of each anomaly detector and its regularized version in the MNIST dataset.

## 7 Ablation Studies

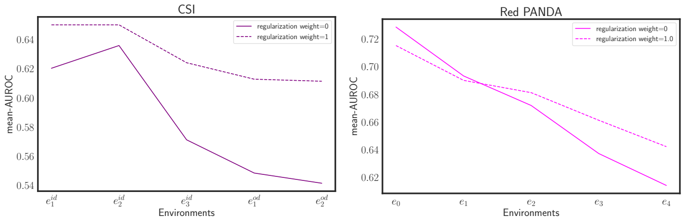
As part of our comprehensive examination of how covariate shifts influence individual regularization weights, we have plotted the performance trajectories of all evaluated models, traversing environments from$e_0$ to $e_4$. These are captured in Fig.9 and Fig.10.

As expected, and already observed in both our main findings and previous work (Ming et al. [2022]), a review of these plots unveils a trend that permeates across all examined models: the performance of models appears to inversely correlate with the number of induced covariate shifts. As the complexity introduced by these shifts mounts, the models' performance experiences a proportionate and systematic decline. This observable trend is essentially monotonic, signifying a erosion in model performance with each incremental rise in the quantity of covariate shifts.

However, it is important to note that this trend is not without exceptions. In particular, when scrutinizing the data pertaining to environment $e_4$, we can observe anomalies to this downward trend.



(a) Fashion-MNIST: STFPM (b) Fashion-MNIST: ReverseDistil (c) Fashion-MNIST: MeanShift



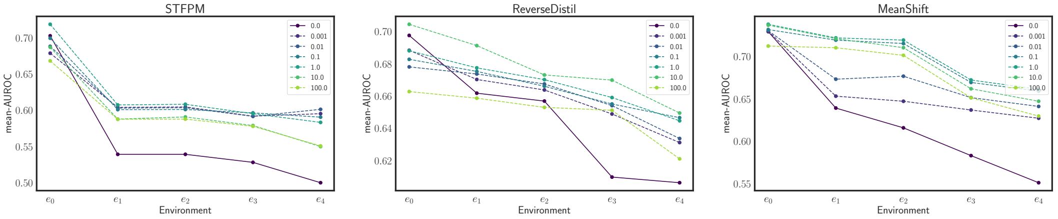
(d) Fashion-MNIST: CSI (e) Fashion-MNIST: Red PANDA

Figure 8: Mean-AUROC curve of each anomaly detector and its regularized version in the Fashion-MNIST dataset.

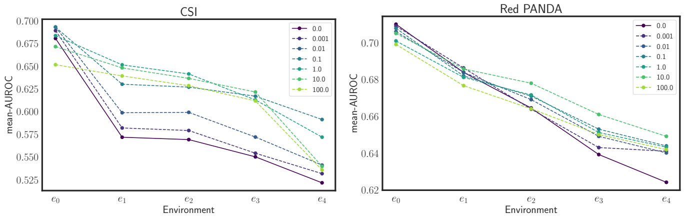
In these exceptional instances, despite the increase in the number of covariate shifts, the performance of certain models appears to resist the general declining pattern.

Thus, our overall conclusion, while acknowledging these exceptions, is that the prevalence of covariate shifts largely contributes to a degradation in model performance.

It is however important to note that the unregularized methods still underperforms when compared to the same method under a even small amount (0.001) of partially conditional regularization added.



(a) MNIST: STFPM (b) MNIST: ReverseDistil (c) MNIST: MeanShift



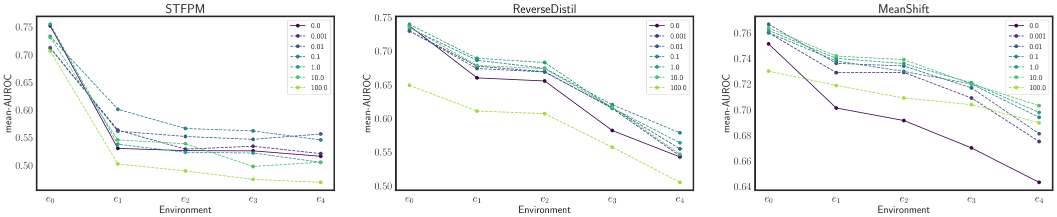
(d) MNIST: CSI (e) MNIST:Red PANDA

Figure 9: Ablation study over the weight of the regularization term for MNIST under distribution shift, with separate plots for each model.

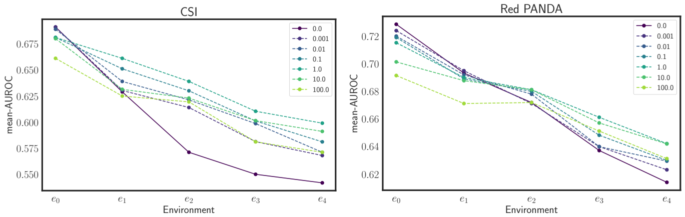
## 8 Additional Discussion

### 8.1 Shortcut Learning in Anomaly Detection

To expand on the experiment tackling real-world shortcut learning, and to better understand how a distribution shift affects different kinds of shortcut features (Geirhos et al. [2020]) captured by the



(a) Fashion-MNIST: STFPM (b) Fashion-MNIST: Reverse Distil (c) Fashion-MNIST: MeanShift



(d) Fashion-MNIST: CSI (e) Fashion-MNIST: Red PANDA

Figure 10: Ablation study over the weight of the regularization term for Fashion-MNIST under distribution shift, with separate plots for each model.

model, we now will look at how inducing distinct changes to the anomaly detection causal graph may lead to malfunctions in the model.

Suppose we recover our formulation of the partition of an object $X$ into the semantic features that distinguish normal and abnormal samples, $X_a$, and into the style features induced by the environment, $X_e$. In that case, it is possible to distinguish between settings that may lead to a model failure when a shortcut feature is captured.

Let us simplify our analysis by considering the setting where the training data is sampled under the intervention $p^{do(E=e')}(X)$, that is the style features are fixed into a specific setting, $X_e^{'}$. This is a prevalent setting in real-world applications as spurious correlations between style and semantic features may occur when sampling the training data. Remember that in the training set, $X_a$ only produces features of normal objects. Under this constraint, it was already previously noted that anomaly detection methods are particularly susceptible to capturing the style features as a prominent factor for the representation of $X$ (Ming et al. [2022]).

Moving to the evaluation stage of the anomaly detector, we can then consider two settings.

The first setting consists of no changes to the intervention $p^{do(E=e')}(X)$, that is, there was no distribution shift in the sampling of the data. In this setting, the main surrogate for a model failure is derived from anomalies that are characterized by small changes in $X_a$ from normal to abnormal samples. That is, the style features would be considered the main source of information to characterize new samples, and under these constraints, it is only natural that all test instances would be highly likely to be set as normal. This setting was introduced by both Ming et al. [2022] and Cohen et al. [2023], and the underlying features can be referred to as *nuisance features*.

The second setting consists of a different intervention, $p^{do(E=e'')}(X)$ such that it differs from the training density $p^{do(E=e')}(X)$. In particular, we can consider an intervention that only changes stylistic features, $X_e$, captured by the model. We are essentially operating under a highly targeted covariate shift that focuses on the shortcut features. Therefore, depending on the extent of the changes in $X_a$ from normal to abnormal samples and how well they are captured by the model, this distribution shift would lead to an anomaly detector that classifies all new samples as anomalies.

Note that in both settings, as shown in our main presentation of the method, by inducing a partially conditional invariance to different environments, our regularization method also inherently introduces an invariance to the style features $X_e$. As supported by our results in the synthetic covariate shift experiments, we believe this also produces models that are not only robust to covariate shifts, as in the second scenario, but also to shortcut learning, as described in the first scenario.

### 8.2 Data Augmentation

Although data augmentation is a well-established technique to tackle model robustness to distribution shifts, there are two main problems with solely relying on data augmentation to solve a problem of distribution shift. First, data augmentation relies on both knowing of the existence of a distribution shift, and a way to simulate it. This could be achievable when the distribution shift is characterized by transformations in easily identifiable attributes (e.g. the color of a digit), and which are also easy to simulate through data transformations. However, in real-world settings, this is rarely the case: distribution shifts are complex transformations that are almost impossible to identify, and in many cases impossible to programmatically simulate. For example, in the case of the Camelyon17 dataset that considers histological cuts from different hospitals, the changes between environments that are derived from the change in histological staining are easy to visually inspect, and could even be simulated. But, by changing the hospital, the patient population also changes, and with it several cofounders, such as group age or patient comorbidities. These are both impossible to know without further annotation and unfeasible to simulate. Furthermore, data augmentation does not induce an invariance to a particular distribution shift, giving no additional "guarantees". It only increases the pool of examples of the data in hopes that the model implicitly captures the additional simulated variance in the images.

Yet, data augmentation could still be beneficial in the context of penalized invariant regularization, by producing meaningful augmentations in the context of a multi-environment setting. It could be used to generate additional data for a single intervention, thus increasing the pool of available samples of a specific environment, or be used to generate new data for a new intervention, leading to an increased number of environments available, and the overall pool of samples in the dataset. This would effectively alleviate the second drawback of solely using data augmentations.

### 8.3 Choice of Distance Metric

We considered other options aside from MMD, like the Wasserstein distance. MMD is known to be computationally easier than Wasserstein's distance. This is because MMD can be easily computed using Gaussian kernels, whereas the Wasserstein distance requires computing a supremum over a set of probability measures, which is computationally intractable in general.

Another option we considered was the KL-divergence. However, previous works have demonstrated that this divergence is very unstable for probability distributions supported on low-dimensional manifolds (Arjovsky et al. [2017]). Note that many of the methods used for anomaly detection and machine learning are trained on samples from such distributions. A similar argument applies for variations of this divergence like Shannon-Jensen divergence and total variation.

Finally, we remark that MMD has a strong theoretical foundation. In spite of its simplicity, it is derived from a supremum of differences of expected values of functions from a reproducible-kernel Hilbert space (Gretton et al. [2012]). It has been shown that when this metric is 0 then the two distributions must match, which is precisely what we aim to achieve when learning representations that are invariant to the environments.

## 9 Implementation Details

### 9.1 Baselines

**STFPM** The STFPM (Wang et al. [2021]) algorithm incorporates a pretrained teacher network and a student network that share the same structure. The student network assimilates the distribution of images devoid of anomalies by aligning the features with corresponding features in the teacher network. To boost robustness, the algorithm uses multi-scale feature matching. This multi-tiered feature alignment lets the student network absorb a blend of multi-level insights from the feature pyramid, thereby permitting the detection of anomalies of varying magnitudes.

During the inference stage, the feature pyramids of both teacher and student networks are put into comparison. A larger discrepancy between these pyramids implies a heightened likelihood of the presence of anomalies.

**Reverse distillation** The reverse distillation method (Deng and Li [2022]) is assembled from three networks: an initial pretrained feature extractor, $f$, a bottleneck embedding, $\phi$, and the student decoder network, $\nu$. The primary layer, or the backbone, of $f$ is derived from a *ResNet* model that was pretrained on the ImageNet dataset.

During the execution of a forward pass, the model extracts features from three separate ResNet blocks. These features are encoded by amalgamating the three feature maps using the multi-scale feature fusion block of $\phi$, and then transferred to the decoder, $\nu$, which is constructed to mirror the feature extractor, albeit with operations reversed.

Throughout the training process, the output of these mirrored blocks is made to match the outputs from the respective layers of the feature extractor. This is ensured by adopting the cosine distance as the loss metric.

**CFA** Feature Adaptation based on CFA (Lee et al. [2022]) identifies anomalies utilizing features that are specifically tailored to the target dataset. The CFA model comprises two main elements: firstly, a learnable patch descriptor that learns and assimilates features oriented towards the target, and secondly, a scalable memory bank that remains unimpacted by the size of the target dataset. In conjunction with a pre-trained encoder, CFA applies a patch descriptor and memory bank. By doing so, it makes use of transfer learning to bolster the density of normal features. Consequently, this facilitates an easier distinction between normal and abnormal features.

**Mean-shifted** The mean-shifted contrastive learning method (Reiss and Hoshen [2021]) introduces a novel loss function that calculates angular distances using the mean of all feature vectors as a reference point. This is done in contrast to using the origin as a reference and the Euclidean distance. It also combines two loss functions, one involving contrastive terms akin to Chen et al. [2020], but these terms are positioned around a hypersphere centred on the mean of all feature vectors. To deter positive samples from repelling themselves, it also incorporates an angular centre loss that encourages samples to gravitate towards the mean of normal samples.

**CSI** CSI (Tack et al. [2020]) is a direct extension of Chen et al. [2020], introducing a unique form of data augmentations known as distribution-shifting augmentations. In this setup, distribution-shifted augmentations are treated as negative samples instead of positive ones and are consequently pushed away from all positive samples. These augmentations include manipulations such as rotations and permutations. The augmentation's potential to shift the distribution is assessed through the AUROC, where samples altered by the said augmentation are considered out-of-distribution samples. The underlying notion here is that distinguishability is directly proportional to the shift in distribution.

**Red PANDA** The Red PANDA method for anomaly detection Cohen et al. [2023] tackles the particular problem of anomaly detection under nuisance or distracting features. Relying on labels from the nuisance factors, it employs a contrastive disentanglement loss following Kahana and Hoshen [2022], in conjunction with a perceptual loss to train a generator function end-to-end with a pretrained encoder.

### 9.2 Anomaly Scoring

**STFPM** During training, the student feature tries to align the distribution of training dataset with the teacher. During prediction, the input $x$ is fed into both student and teacher feature extractors, where the student outputs $f_s(x)$ and teacher outputs $f_t(x)$. For the anomaly scoring, it relies in a traditional density estimation approach. The assumption would be that the normal samples are mapped to the high density area where the student encoder and the teacher encoder are aligned, and the anomalous samples are mapped to the low density area of the student extractor. Therefore, the anomaly score is computed as the distance between $f_s(x)$ and $f_t(x)$, i.e. $AS(x) = d(f_s(x), f_t(x))$. In this case the distance metric $d$ is the $l_2$-norm.

**Reverse distillation** The anomaly scoring function here is derived from the standard anomaly scoring functions used in reconstruction-based anomaly detection algorithm. In particular, the anomaly score is defined as the distance between the encoded features and the reconstructed features from the decoder. $AS(y) = d(f(y), \nu(\phi(f(y)))$, where $\nu(\cdot)$ is the decoder, $\phi(\cdot)$ is the distiller and $f(\cdot)$ is the encoder. The idea behind this anomaly scoring function is that the decoder has learned to reconstruct normal samples due to training dataset, but isn't able to reconstruct anomalous samples. Therefore, anomalous samples would have a higher anomaly score.

**CFA** For CFA we use the standard image-level density estimation approach. In particular, the training samples are mapped to a feature map $f(x)$ and clustered into $k$ clusters using $k$-means. For the prediction, given an input $y$, its final feature $f(y)$ is computed after feeding it into the feature extractor and descriptor. Then the $d$ nearest cluster centers of $f(y)$ would be selected, and the anomaly score for that sample is the mean distance to those $d$ centers. $AS(y) = \Sigma_{f_i \in N_k(x)} d(f_i, f(y))$. In this case, the distance metric $d$ is simply the $l_2$ distance. The idea behind this approach is to assume that the anomalous samples would be mapped to low-density area in feature space, which are far away from all the cluster centers, while the normal samples would be mapped to high density area in feature space, which is close to the normal samples in training dataset.

**Mean-shift and Red PANDA** Both contrastive learning based methods used as baselines, namely, mean-shift (Reiss and Hoshen [2021]) and Red Panda (Cohen et al. [2023]) rely on finetuning an encoder (both pre-trained or not) by grouping the set of feature vectors from images in the training data around a sub-region of the hypersphere centered in the origin. During prediction time, the most common approach to classify a new sample as anomaly or not is through the mean distance of the $k$NN normal images. Following the original work, in mean-shift, $k$ was set to 2, and in Red Panda it was set to 1.

**CSI** CSI (Tack et al. [2020]) relied on only a vanilla contrastive loss between original samples, and highly augmented samples to serve as a proxy for abnormal samples. Similarly to the previous setting of mean-shift and Red Panda, this leads to a feature space that falls around the hypersphere centered in the origin, but not necessarily in the surface. However, operating with the underlying hypothesis that the highly augmented samples match the anomalies, the feature vectors from images in the training data are already being pushed to the diametrically opposite side of the hypersphere when compared to abnormal samples. Additionally, it was empirically verified that the norm of vectors of abnormal samples is much lower than that of in-distribution samples. This leads to a distance criterion that measures the closest training sample through the cosine similarity and the norm of the feature vector of the sample: $\max_m \text{sim}(f(x_m), f(x)) \cdot |f(x)|$, where $f$ is the encoder that maps the input object, $x$, to its feature space, and sim is the cosine similarity.

### 9.3 Hyperparameters

As this work considered a novel setting where each anomaly detection method was evaluated for the first time, we modestly optimized hyperparameters. Our approach consisted of two primary steps. The first involved scaling up two key factors: (a) batch size, and (b) learning rate. Subsequently, we methodically scanned through an array of distinct parameters for each baseline model. These included the backbones ResNet18, ResNet34, ResNet50 and WideResNet50, alongside various anomaly scoring methodologies that leverage image-level, density estimation, reconstruction error, and pixel-wise density estimation approaches. An additional aspect of our study was an ablation analysis where the regularization weight was fine-tuned by sweeping through the set of values $0.001, 0.01, 0.1, 1, 10, 100$. We adhered to the hyperparameters as depicted in the original works for all other variables and refrained from performing any further optimizations on them.

### 9.4 Backbone choice

One thing we notice during our experiments is that for models that rely on the pretrained backbones, the choice of backbone matters. For instance, for STFPM, the optimal choice was the simplest feature extractor ResNet18, but for reverse distillation, the optimal choice was WideResNet50. The choices seem to be model dependent more than dataset relevant. For more details on the chosen backbone for each method refer to Tab. 7

### 9.5 Computational resources

The complete project required 3400 hours of GPU usage throughout all experiments, covering development, testing, and comparisons. The resources supplied were part of a local custer, and consited of two GPU models: the NVIDIA TITAN RTX and the NVIDIA Tesla V100.

| | STFPM | ReverseDistil | CFA | MeanShift | CSI | Red PANDA |
|---|---|---|---|---|---|---|
| | Wang et al. [2021] | Deng and Li [2022] | Lee et al. [2022] | Reiss and Hoshen [2021] | Tack et al. [2020] | Cohen et al. [2023] |
| **Camelyon17** (3x96x96) | | | | | | |
| learning rate | $10^{-2}$ | $10^{-2}$ | $10^{-5}$ | $10^{-3}$ | $10^{-3}$ | $10^{-4}$ |
| optimizer | SGD | Adam | AdamW | Adam | Adam | SGD |
| batch size | 32 | 32 | 16 | 32 | 64 | 128 |
| backbone | ResNet18 | WideResNet50 | ResNet18 | ResNet50 | ResNet18 | ResNet50 |
| pretaining | True | True | True | True | False | True |
| best reg. weight | 100 | 10 | 100 | 10 | 1 | 10 |
| anomaly score | density estimation | reconstruction error | density estimation | density estimation | density estimation | density estimation |
| **DiagViB-6 (MNIST)** (3x256x256) | | | | | | |
| learning rate | $10^{-1}$ | $10^{-2}$ | - | $10^{-3}$ | $10^{-3}$ | $3 \cdot 10^{-4}$ |
| optimizer | SGD | Adam | - | Adam | Adam | SGD |
| batch size | 32 | 32 | - | 32 | 32 | 32 |
| backbone | ResNet18 | WideResnet50 | - | ResNet50 | ResNet18 | ResNet50 |
| pretaining | True | True | - | True | False | True |
| best reg. weight | 1 | 10 | - | 1 | 1 | 10 |
| anomaly score | density estimation | reconstruction error | - | density estimation | density estimation | density estimation |
| **DiagViB-6 (Fashion-MNIST)** (3x256x256) | | | | | | |
| learning rate | $10^{-1}$ | $10^{-2}$ | - | $10^{-3}$ | $10^{-3}$ | $3 \cdot 10^{-4}$ |
| optimizer | SGD | Adam | - | Adam | Adam | SGD |
| batch size | 32 | 32 | - | 32 | 32 | 32 |
| backbone | ResNet18 | WideRestNet50 | - | ResNet50 | ResNet18 | ResNet50 |
| pretaining | True | True | - | True | False | True |
| best reg. weight | 0.1 | 1 | - | 10 | 1 | 1 |
| anomaly score | density estimation | reconstruction error | - | density estimation | density estimation | density estimation |

Table 7: A detailed summary of the hyperparameters used for each evaluated model across three datasets: Camelyon17, DiagViB-6 (MNIST), and DiagViB-6 (Fashion-MNIST). Parameters include learning rate, scheduler, optimizer, batch size, backbone, pretraining, regularization weight, and mmd kernel type, along with the type of anomaly score. Notably, the CFA model could not be successfully implemented for DiagViB-6 based experiments despite trying an extensive range of hyperparameter combinations. Models are referenced by their respective citations.

### 9.6 Code and Licensing

The main Python libraries used in our implementation, were Pytorch, which is under a BSD-3 license[1], and Pytorch Lightning, which is under Apache 2.0 license[2].

Methods that were derived from the anomalib library (Akcay et al. [2022]), namely STFPM, reverse distillation, and CFA, were already implemented as a Pytorch Lightning Module, and are all under an Apache 2.0 license[3]. These were incorporated directly in our pipeline.

DCoDR (Kahana and Hoshen [2022]), which Red Panda is based from, was released under a Software Research License[4]. Our experiments for Red Panda were derived directly from the official repository. Mean-shifted contrastive learning was released under a Software Research License[5]. We re-implemented this method as a Pytorch Lightning Module, loosely following its original official implementation. DiagViB-6 (Eulig et al. [2021]) and the Camelyon17 (Koh et al. [2021]) datasets were also publicly released with a GNU Affero General Public License v3.0[6] and a MIT License[7], respectively. Our implementation follows directly from its official repository.

---

[1] https://github.com/pytorch/pytorch/blob/main/LICENSE

[2] https://github.com/Lightning-AI/lightning/blob/master/LICENSE

[3] https://github.com/openvinotoolkit/anomalib/blob/main/LICENSE

[4] https://github.com/jonkahana/DCoDR/blob/main/LICENSE

[5] https://github.com/talreiss/Mean-Shifted-Anomaly-Detection/blob/main/LICENSE

[6] https://github.com/boschresearch/diagvib-6/blob/main/LICENSE

[7] https://github.com/p-lambda/wilds/blob/main/LICENSE

## References

S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc. Anomalib: A deep learning library for anomaly detection, 2022.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/arjovsky17a.html`.

P. Bandi, O. Geessink, Q. Manson, M. Van Dijk, M. Balkenhol, M. Hermsen, B. E. Bejnordi, B. Lee, K. Paeng, A. Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 38 (2):550–560, 2018.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

N. Cohen, J. Kahana, and Y. Hoshen. Red PANDA: Disambiguating image anomaly detection by removing nuisance factors. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=z37tDDHHgi`.

H. Deng and X. Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9737–9746, June 2022.

L. Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

E. Eulig, P. Saranrittichai, C. K. Mummadi, K. Rambach, W. Beluch, X. Shi, and V. Fischer. Diagvib-6: A diagnostic benchmark suite for vision models in the presence of shortcut and generalization opportunities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10655–10664, 2021.

R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

J. Kahana and Y. Hoshen. A contrastive objective for learning disentangled representations. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 579–595. Springer, 2022.

P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

S. Lee, S. Lee, and B. C. Song. CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *arXiv preprint arXiv:2206.04325*, 2022.

Y. Ming, H. Yin, and Y. Li. On the impact of spurious correlation for out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10051–10059, 2022.

T. Reiss and Y. Hoshen. Mean-shifted contrastive loss for anomaly detection. *arXiv preprint arXiv:2106.03844*, 2021.

S. Smeu, E. Burceanu, A. L. Nicolicioiu, and E. Haller. Env-aware anomaly detection: Ignore style changes, stay true to content! *arXiv preprint arXiv:2210.03103*, 2022.

J. Tack, S. Mo, J. Jeong, and J. Shin. CSI: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, 2020.

D. Tellez, G. Litjens, P. Bándi, W. Bulten, J.-M. Bokhorst, F. Ciompi, and J. Van Der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58:101544, 2019.

V. Veitch, A. D'Amour, S. Yadlowsky, and J. Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

G. Wang, S. Han, E. Ding, and D. Huang. Student-teacher feature pyramid matching for anomaly detection. In *The British Machine Vision Conference (BMVC)*, 2021.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.