DECEPTIVE RISK MINIMIZATION: OUT-OF-DISTRIBUTION GENERALIZATION BY DECEIVING DISTRIBUTION SHIFT DETECTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper proposes *deception* as a mechanism for out-of-distribution (OOD) generalization: by learning data representations that make training data *appear* independent and identically distributed (iid) to an observer, we can identify stable features that eliminate spurious correlations and generalize to unseen domains. We refer to this principle as *deceptive risk minimization* (DRM) and instantiate it with a practical differentiable objective that simultaneously learns features that eliminate distribution shifts from the perspective of a detector based on conformal martingales while minimizing a task-specific loss. In contrast to domain adaptation or prior invariant representation learning methods, DRM does not require access to test data or a partitioning of training data into a finite number of data-generating domains. We demonstrate the efficacy of DRM on numerical experiments with concept shift and a simulated imitation learning setting with covariate shift in environments that a robot is deployed in.

1 INTRODUCTION

Is there an unbridgeable gap between in-distribution (ID) and out-of-distribution (OOD) generalization in machine learning? Or can the distinction be erased by a change in perspective? Traditional wisdom holds that there is a vast chasm between the two settings. Applications where training environments are representative of test environments (e.g., via careful curation of large-scale datasets) have seen remarkable empirical progress and real-world impact. This success is backed by a deep theoretical understanding of ID generalization from decades of progress in statistical learning theory (Shalev-Shwartz & Ben-David, 2014). However, in settings where it is challenging to cover all relevant dimensions of variation exhaustively in the training data — a common occurrence in real-world applications such as robotics, healthcare, and cybersecurity — high-capacity models can absorb spurious correlations and fail catastrophically in test settings where these correlations are altered or even reversed (Liu et al., 2021; Sinha et al., 2022; Li et al., 2025; Arjovsky et al., 2019).

In this paper, we take an *observer-centric* viewpoint on the gap between ID and OOD generalization. Our starting point is the following basic observation: from the perspective of an external observer who cannot discern distribution shifts in a sequence of data, *OOD generalization is equivalent to ID generalization*. As an example, consider an observer responsible for overseeing the performance of a robot operating in a warehouse. The robot is presented with a sequence of objects to place into a receptacle, while the observer records the corresponding sequence of bits denoting success (1) or failure (0) of the robot. During this process, the robot encounters changes to its lighting conditions, appearances of objects, and its visual backdrop. However, the robot is able to maintain reliable performance throughout these changes, with only a small-but-consistent failure probability. As a result, the observer is completely oblivious to the distribution shifts faced by the robot: the data recorded by the observer has shed its spurious, domain-specific cues and appears independent and identically distributed (iid). In a sense, the robot has *hidden* the distribution shifts from the observer.

The core idea of this work is to translate this observer-centric perspective on generalization into a prescriptive mechanism for OOD generalization. Suppose that a learner is presented with a sequence of training data that exhibits (potentially mild) distribution shifts. Then, by learning data representations that eliminate these distribution shifts from the perspective of an observer, we can identify stable features that do not rely on spurious correlations and generalize to unseen domains.

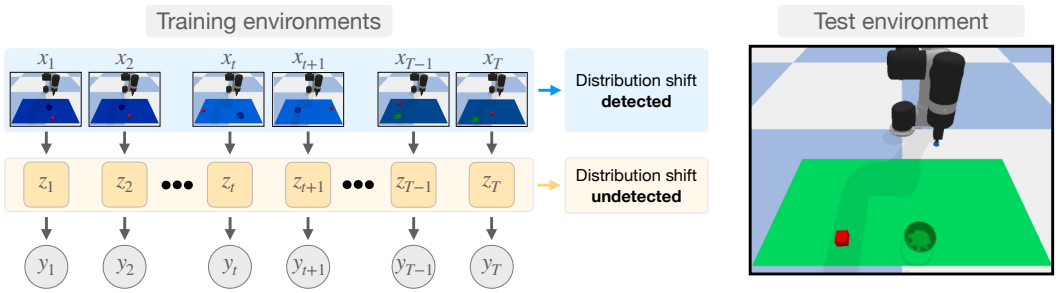

Figure 1: Deceptive risk minimization (DRM): by learning data representations that make training data *appear* iid to an observer (left), we can identify stable features that eliminate spurious correlations and generalize to unseen domains (right). In the figure, the robot's training environments undergo small-but-structured changes in the colors of the table and objects; DRM learns a representation that is insensitive to these changes, which results in generalization to environments with significantly different appearances.

For example, if the robot encounters periodic changes in lighting conditions or visual backdrops (Fig. 1), an encoding of observations that hides these changes from a distribution shift detector will eliminate sensitivity to the changes and result in robust performance when spurious correlations in training data are significantly exaggerated or reversed.

Concretely, we formulate this learning mechanism as an adversarial game (Fig. 1) which we refer to as *deceptive risk minimization* (DRM). An encoder network learns to generate representations that support minimization of a task-specific loss while simultaneously eliminating distribution shifts from the perspective of a detector presented with a sequence of learned representations. We assume that this sequence is presented in the order in which training data were curated, and thus preserves the structure of natural distribution shifts in the original data. For example, in robotics, training data are often collected in environments that vary over time either discretely (e.g., a change in the color of the table) or continuously (e.g., a continuous change in ambient outdoor lighting over a day). We are interested in settings where the training data sequence exhibits some distribution shifts, which are exaggerated or reversed at deployment time. Importantly, unlike prior work in domain adaptation (Ben-David et al., 2010; Ganin et al., 2016; Zhang et al., 2015; Long et al., 2018) or invariant representation learning (Arjovsky et al., 2019; Peters et al., 2016; Ahuja et al., 2020; Krueger et al., 2021), we do not assume access to any data from test environments or that training data are partitioned into different domains corresponding to data-generating distributions; we simply assume that the order of data is preserved. Associating data points with a finite set of domains — either manually or via unsupervised clustering (Le et al., 2025; Murata et al., 2025) — is often impractical or unachievable in settings where there is a continuous change in conditions.

We present a practical instantiation of DRM that utilizes *conformal martingales* (CMs) (Vovk et al., 2022; 2003; Vovk, 2021) for distribution shift detection. CMs offer a general and flexible approach to distribution shift detection, which is often highly effective in practical scenarios (Vovk et al., 2022, Ch. 8). Concretely, the CM approach computes a quantity that remains small when data are iid (or exchangeable), but that can grow quickly in the presence of distribution shifts. We derive an end-to-end differentiable loss that penalizes the conformal martingale computed on encoded inputs; this loss serves to train the encoder to learn representations that eliminate distribution shifts from the perspective of the CM-based detector.

Summary of contributions. We introduce deceptive risk minimization (DRM): a novel learning principle that estimates representations that eliminate spurious correlations by deceiving distribution shift detectors. We develop a practical instantiation of DRM via a differentiable loss that penalizes conformal martingales, and demonstrate the efficacy of this representation learning objective in different empirical settings involving covariate and concept shift. Conceptually, DRM creates a bridge between distribution shift detection and OOD generalization, which we hope future work can build on to unlock practical methods for OOD generalization in real-world applications.

2 Prelude: Randomness and Structure in the Eye of the Beholder

We provide an illustrative example below in order to explain the key intuitions behind our approach. Consider an imitation learning setting where a human has provided a sequence of examples to a robot demonstrating how to perform a given task (Fig. 1). These demonstrations are provided in

environments that exhibit a small amount of distribution shift. Specifically, the color of the table is varied *slightly* in a structured way: a third of the demonstrations are provided with one table-bowl color combination, the next third with a slightly different color combination, and the final third with another. The standard approach to learning a policy in such a setting is empirical risk minimization (ERM): *assume* that data are iid and learn a mapping from the robot's observations to actions by minimizing a behavior cloning loss on training data. Such a policy performs well when deployed with table colors similar to ones seen during training, but fails with colors that are significantly different (see Sec. 5.3 for numerical results).

The starting point for our approach is the observation that the non-iid nature of data in this setting can be *inferred from the sequence of training environments*. Fig. 2 shows the output from a distribution shift detector based on conformal martingales (described formally in Sec. 4) computed on observations from the training environments. This detector spikes strongly once the table color is changed. Crucially, the detector also spikes when provided with the sequence of latent features from the policy computed via ERM, indicating that the policy's latent representation encodes color information.

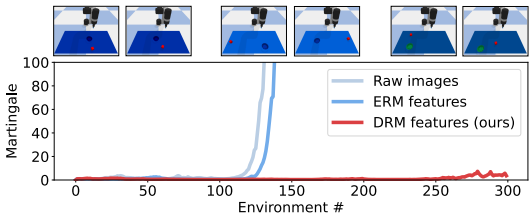

Figure 2: Conformal martingales rapidly detect the distribution shift with raw images or features computed via ERM; in contrast, features from DRM appear iid.

Now consider a policy that eliminates the distribution shift from the perspective of an observer who is only presented with latent policy representations. Intuitively, the data can be made to "appear iid" by eliminating sensitivity to the table color, which leads to OOD generalization to different colors.

3 Problem Formulation

Training data. A learner is presented with a *sequence* of training data $((x_t, y_t))_{t=1}^{T} \in (\mathcal{X} \times \mathcal{Y})^T$ consisting of input-label pairs sampled from a sequence of random variables $((X_t, Y_t))_{t=1}^{T}$, which may be dependent and non-identically distributed. This sequential collection of data is a core assumption of our work and departs from the standard practice of shuffling data. If some of the data collection is parallelized (e.g., by multiple human operators collecting data on different robots on the same day), we assume that this data is serialized by committing to a particular ordering.

Hypothesis, loss, and OOD generalization. Given the training data, the learner produces a hypothesis $h : \mathcal{X} \to \mathcal{Y}$, which maps inputs to predicted labels. The hypothesis is evaluated according to a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, which compares predicted labels with ground-truth labels. The learned hypothesis h is deployed on a sequence of test data drawn from random variables $((X_\tau, Y_\tau))_{\tau=T+1}^{T+T'}$. The overall quality of the hypothesis is measured by its expected loss on test data $\frac{1}{T'} \sum_{\tau=T+1}^{T+T'} \mathbb{E}\left[l(h(X_\tau), Y_\tau)\right]$. When the random variables $(X_t, Y_t)_{t=1}^{T+T'}$ are iid, this reduces to the standard statistical learning setting. Of interest to us is the case where training data reflect some (possibly mild) distribution shifts, which are significantly exaggerated or reversed for test data.

Covariate and concept shifts. We consider two types of distribution shifts: (i) covariate shift, where the distribution of inputs changes over time (e.g., changes in the color of the robot's table in the example from Sec. 2), and (ii) (anti-causal) concept shift, where the conditional distribution of $X|Y$ changes over time (e.g., an image classification setting where the appearances of images for a given label change over time). We discuss potential extensions (e.g., causal concept shift) in Sec. 7.

4 Algorithmic Implementation: Deceptive Risk Minimization

Our goal is to find features that eliminate the distribution shift between training and test settings. Since the learner is only provided with the training sequence, we utilize distribution shifts observed in this data as a proxy. Specifically, we learn features that are stable along the training data sequence — in the sense that they appear iid to an observer — while also minimizing a task-specific loss. We formalize this objective by defining an observer in the form of a distribution shift detector.

Defining the observer. Let $(X_1, X_2, \dots)$ be a sequence of input random variables, and let ϕ be a mapping from an input x to an encoding $\phi(x) \in \mathbb{R}^{n_d}$. We define an observer Δ who takes as

input a realization $(\phi(x_1), \phi(x_2), \dots)$ from the random variables $(\phi(X_1), \phi(X_2), \dots)$ and outputs a boolean $\delta \in \{\text{True}, \text{False}\}$ indicating if a deviation from the iid hypothesis has been detected.

Definition 1 (Practically iid)**.** *A sequence $(\phi(x_1), \phi(x_2), \dots)$ is* Δ-practically iid *if a distribution shift detector Δ does not trigger*[1] *when provided with this sequence as input, i.e., $\Delta((\phi(x_1), \phi(x_2), \dots)) =$ False.*

False alarm rate (FAR). For a detector Δ to be useful, it should not trigger too often when presented with data drawn from a sequence of iid random variables. This is captured by the false alarm rate (FAR), which is the worst-case probability of detection when data are drawn from an iid sequence of random variables (Vovk et al., 2022, Ch. 8).

Formalizing the DRM objective. Next, we formulate the objective of deceptive risk minimization (DRM) as a constrained optimization problem. We consider hypotheses $h : x \mapsto \phi(x) \mapsto f(\phi(x)) \in \mathcal{Y}$, which encode inputs using ϕ and map these to labels via f. We use $(\phi(x_t)|y)_{t=1}^T$ to denote the subsequence of $(\phi(x_t))_{t=1}^T$ with labels equal to y. The following optimization problem minimizes the task-specific loss l while searching for a representation that makes the training data Δ-practically iid. We consider two types of constraints aimed at tackling covariate shift and concept shift respectively.

Deceptive risk minimization (DRM)

$$\inf_{f,\phi} \ \frac{1}{T}\sum_{t=1}^{T} l(x_t, f(\phi(x_t))$$

$$\text{s.t. } (\phi(x_t))_{t=1}^T \text{ is } \Delta\text{-practically iid} \quad \text{[covariate shift]} \tag{1}$$

$$(\phi(x_t)|y)_{t=1}^T \text{ is } \Delta\text{-practically iid}, \forall y \in \mathcal{Y} \quad \text{[concept shift]}. \tag{2}$$

Instantiation with conformal martingales. We now describe a particular distribution shift detector Δ based on conformal martingales (CMs). This will allow us to flexibly handle both covariate and concept shifts. In addition, this detector will allow us to formulate a *differentiable* surrogate for the constraints equation 1 and equation 2 in the DRM optimization problem. Intuitively, the CM approach constructs a quantity that grows quickly when random variables are not iid, and remains small otherwise. In order to detect covariate shift, we first assess how well every encoded data point $\phi(x_i)$ (with $i \leq t$) *conforms* to the sequence of data points $(\phi(x_j))_{j=1}^t$ observed up to time t using a conformity score:

$$\alpha_i^{\text{covariate}} := \min_{j \in \{1,\dots,t\}: j \neq i} d(\phi(x_i), \phi(x_j)), \tag{3}$$

where $d : \mathbb{R}^{n_d} \times \mathbb{R}^{n_d} \to \mathbb{R}^+$ captures how different two encoded data points $\phi(x_i)$ and $\phi(x_j)$ are. In our numerical experiments, we will use the cosine distance or its sharpened form (Ahmad & Mazzara, 2024). In cases where we are interested in detecting concept shift rather than covariate shift, we will alternately use a label-conditioned conformity score:

$$\alpha_i^{\text{concept}} := \min_{j \in \{1,\dots,t\}: j \neq i, y_j = y_i} d(\phi(x_i), \phi(x_j)). \tag{4}$$

The conformity scores are used to compute conformal p-values for each $t \leq T$:

$$p_t^{\text{covariate}} := \frac{|\{i|1 \leq i \leq t, \alpha_i < \alpha_t\}| + \xi_t|\{i|1 \leq i \leq t, \alpha_i = \alpha_t\}|}{t}, \tag{5}$$

$$p_t^{\text{concept}} := \frac{|\{i|1 \leq i \leq t, \alpha_i < \alpha_t, y_i = y_t\}| + \xi_t|\{i|1 \leq i \leq t, \alpha_i = \alpha_t, y_i = y_t\}|}{|\{i|1 \leq i \leq t, y_i = y_t\}|}, \tag{6}$$

where $\xi_t \in [0, 1]$ is sampled independently from the uniform distribution on $[0, 1]$.

As shown by Vovk et al. (2022, Ch. 2), the p-values $p_t^{\text{covariate}}$ and p_t^{concept} are independent and uniformly distributed in $[0, 1]$ in the *absence* of covariate and concept shift respectively. Thus, the CM approach constructs a quantity that measures how far away from being uniformly and independently distributed the p-values are. This is achieved using a betting martingale (Vovk et al., 2022, Ch. 8), whose computation is shown in Algorithm 1.

[1]To keep this definition simple, we restrict attention to deterministic detectors (e.g., constructed by taking a detector that has randomness and fixing the seed).

Intuitively, the betting martingale represents the capital of a bettor who gambles *against* the hypothesis that random variables are iid. Large values of the betting martingale S_t thus serve as an indicator of distribution shift. Conversely, in the *absence* of distribution shift, S_t is guaranteed to remain small with high probability. In particular, a detector Δ^{CM} which is triggered if the betting martingale ever exceeds a threshold $1/\alpha$ has a false alarm rate bounded by α Vovk et al. (2022).

Algorithm 1: Betting martingale

Inputs: $p_1, \ldots, p_T$; **Outputs:** $S_1, \ldots, S_T$
Define $E = \{-1, -0.5, 0, 0.5, 1\}$, $\mu = 0.005$
Initialize $C \leftarrow 1$, $C_e \leftarrow 1/|E|$
for $t = 1$ to T **do**
  for each $e \in E$ **do**: $C_e \leftarrow (1-\mu)C_e + (\mu/|E|)C$
  end for
  for each $e \in E$ **do**: $C_e \leftarrow C_e\,[1 + e(p_t - 0.5)]$
  end for
  $C \leftarrow \sum_{e \in E} C_e$; $\quad S_t \leftarrow C$
end for

Conformal martingales suggest a practical method to instantiate the DRM optimization problem: replace the hard constraints in Eq. 1 and Eq. 2 with soft constraints that penalize large values of the betting martingale. The only remaining hurdle is to make the martingale computation differentiable. The steps in Algorithm 1 are differentiable, and hence the only sources of non-differentiability are in the computation of the conformity scores α_i (Eq. 3 or Eq. 4) and the p-values p_t (Eq. 5 or Eq. 6). We follow a procedure similar to Stutz et al. (2021), which differentiates through the calibration procedure of conformal prediction. We replace the minimization operation in Eq. 3 or Eq. 4 by the standard soft-min operation. The computation of $\{i | 1 \leq i \leq t, \alpha_i < \alpha_t\}$ (or $\{i | 1 \leq i \leq t, \alpha_i < \alpha_t, y_i = y_t\}$) is equivalent to the computation of a quantile. This can be approximated by smoothed sorting methods (Blondel et al., 2020; Cuturi et al., 2019), which have a "dispersion" hyperparameter σ such that smooth sorting approaches hard sorting as $\sigma \to 0$.

We thus formulate the practical instantiation of DRM as follows by optimizing a weighted combination of the task-specific supervised learning loss (e.g., cross-entropy) and the soft martingale values. We discuss additional algorithmic implementation details in Appendix A.

Deceptive risk minimization: differentiable objective

$$\inf_{f,\phi} \frac{1}{T}\sum_{t=1}^{T} l(x_t, f(\phi(x_t)) + \lambda \frac{1}{T}\sum_{t=1}^{T} \tilde{S}_t(\phi(x_1), \ldots, \phi(x_t)), \tag{7}$$

where $\tilde{S}_t(\phi(x_1), \ldots, \phi(x_t))$ is the soft martingale (see Algorithm 2 for details). (8)

5 Experiments

We evaluate DRM in three sets of experiments, which seek to investigate the following questions: (1) How effective is DRM in enabling OOD generalization with concept shift or covariate shift and spurious correlations in the training data? (2) Can DRM match the performance of invariant risk minimization (IRM) (Arjovsky et al., 2019), which assumes an oracle partitioning of training data into a finite number of "environments" corresponding to different data distributions? (3) How effective is the conformal martingale approach for distribution shift detection, which forms the bedrock of DRM's algorithmic implementation? Hyperparameters for experiments are listed in Appendix B.

5.1 Concept shift: toy 2D example

We begin with a binary classification task with 2D inputs, where one input dimension correlates strongly but spuriously with the label. Empirical risk minimization (ERM) latches on to this correlation and relies heavily on the spuriously correlated input dimension. However, when the correlation is reversed at test time, the performance of the ERM classifier collapses. This is a 2D version of Colored-MNIST (Arjovsky et al., 2019), which allows for easy visualization.

Training and test distributions. The learner is presented with a sequence of training data $(x_t, y_t)_{t=1}^{T}$, where $x_t := [x_t^{[1]}, x_t^{[2]}]$ is two-dimensional and $y_t \in \{0, 1\}$. The first input dimension $x_t^{[1]}$ is drawn from a normal distribution $\mathcal{N}(0, 2^2)$, and a preliminary label $\tilde{y}_t$ is assigned purely as a function of $x_t^{[1]}$: $\tilde{y}_t = 1$ if and only if $x_t^{[1]} \geq 0$. The final label y_t is assigned by flipping $\tilde{y}_t$ with probability 0.25. The second dimension $x_t^{[2]}$ of the input is constructed so that it strongly correlates with the label. Specifically, we first construct $\tilde{x}_t^{[2]} = y_t^{\text{sign}} + u \cdot y_t^{\text{sign}}$, where $y_t^{\text{sign}} = 2y_t - 1$ and u is sampled

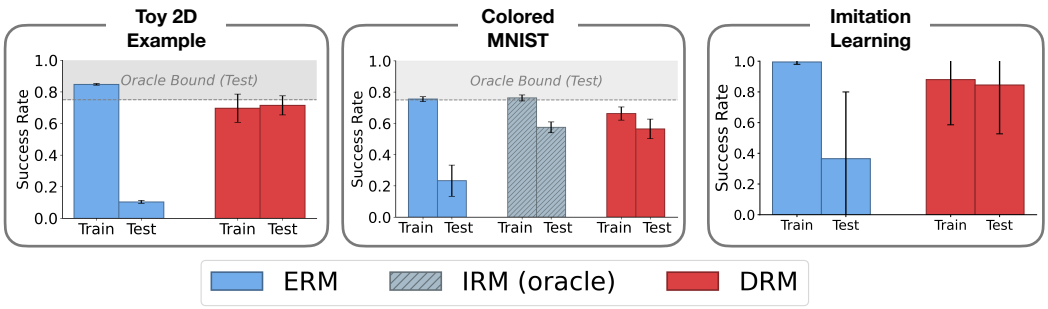

Figure 3: Training and test success rates for the three examples. ERM latches on to spurious correlations or distractors in each case, resulting in severe performance degradation at test time. In contrast, DRM learns stable features that lead to strong generalization. For Colored-MNIST, DRM also matches the performance of IRM, which assumes oracle knowledge of the time at which a distribution shift occurs. For the 2D example and Colored-MNIST, the maximal achievable test success for any robust classifier is 0.75 ("Oracle Bound (Test)").

from the uniform distribution on $[0, 1]$. The learner observes $x_t^{[2]}$, which flips the sign of $\tilde{x}_t^{[2]}$ with a probability that varies smoothly from p_1 to p_T over time: $p_t = p_1 + (p_T - p_1)(t-1)/(T-1)$, for $t \in \{1, \ldots, T\}$, with $p_1 = 0$ and $p_T = 0.3$. This time-varying probability is the source of concept shift in the training data, where the distribution of the input conditioned on the label varies slightly over time. At test time, the correlation between $x_t^{[2]}$ and the label y_t is reversed by choosing a flipping probability $p_{\text{test}} = 0.9$. We highlight that IRM (Arjovsky et al., 2019) and its variants (Wang et al., 2022; Krueger et al., 2021; Ahuja et al., 2020; Lu et al., 2021) — which assume that data are separated according to finitely many data-generating distributions — are not directly applicable here since the distribution changes *continuously* for the training data.

Results. We train a multi-layer perceptron using both ERM and DRM, and utilize the last hidden representation (unit-normalized) as our feature $\phi(x_t)$ for computing the conformal test martingale in DRM. Fig. 3 (left) compares the performance of ERM with DRM on training and test data (across 10 seeds). The reversal of the spurious correlation results in a dramatic drop in performance on test data for ERM. In contrast, the performance of the classifier learned by DRM is almost entirely unimpacted. This performance also nearly matches an oracle that relies exclusively on $x_t^{[1]}$ for classification, which has a 0.75 classification accuracy on test data.

Visualizing classifiers. Fig. 4 visualizes the classifiers learned by ERM and DRM. ERM learns a classifier that heavily exploits the spuriously correlated input dimension $x_t^{[2]}$ in order to maximize training performance, which leads to a collapse in performance on the test distribution. In contrast, DRM learns a classifier that relies almost exclusively on the robust input dimension $x_t^{[1]}$.

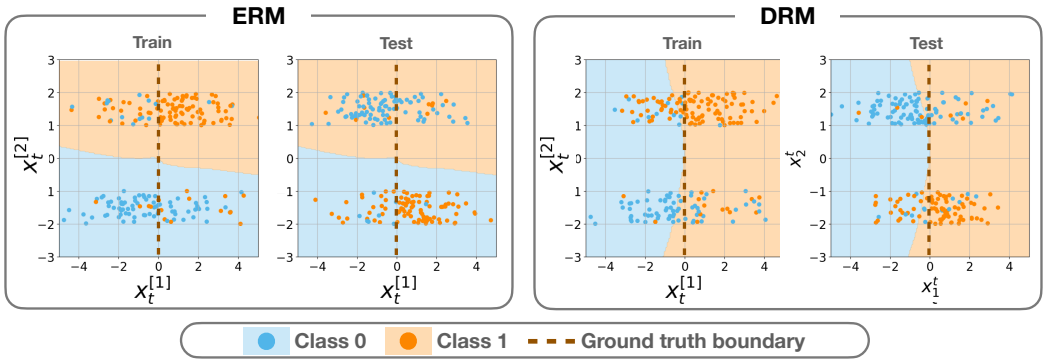

Figure 4: Classifiers learned by ERM and DRM (2D problem). ERM separates data according to the spurious input dimension $x_t^{[2]}$, which leads to poor performance at test time. DRM disregards $x_t^{[2]}$ almost entirely and learns a classifier that is close to the ground truth ($x_t^{[1]} \geq 0$), leading to strong generalization.

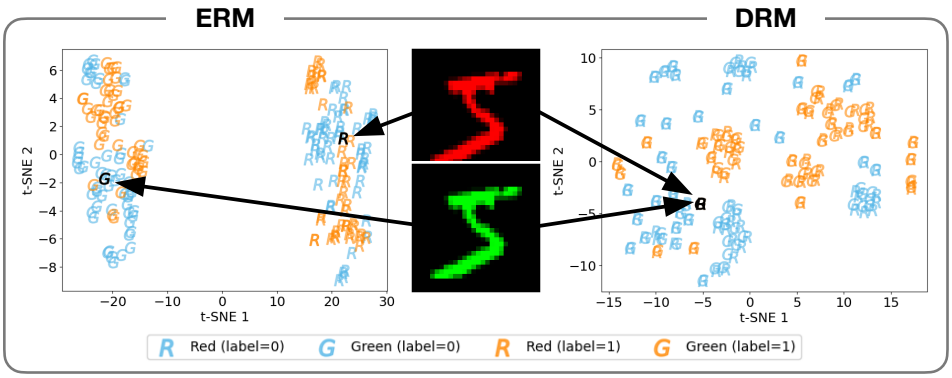

Figure 5: The t-SNE embeddings for features learned by ERM and DRM for Colored-MNIST. ERM embeddings are clustered distinctly by color (R/G). In contrast, DRM embeddings are clustered based on the label, suggesting that DRM has learned to ignore the spurious color information.

5.2 Concept Shift: Colored-MNIST

Next, we consider the Colored-MNIST task introduced in (Arjovsky et al., 2019). The goal is to classify MNIST (LeCun et al., 1995) images, where the digits have been colored either red or green. Similar to the toy 2D example, the color is assigned in a way that has a strong (but spurious) correlation with the label. As a result, ERM-based methods that only rely on minimizing training loss exploit the color information to make predictions; when the correlation between color and the label is reversed at test time, performance collapses.

Training and test distributions. Each image is first assigned a preliminary label $\tilde{y} = 0$ for digits $0 - 4$ and $\tilde{y} = 1$ for $5 - 9$. The final label y flips $\tilde{y}$ with probability 0.25. A color id $c \in \{0, 1\}$ is obtained by flipping y with probability p_t, and the image is colored red if $c = 1$ and green if $c = 0$. The training data sequence consists of examples drawn from two distributions, with the change-point occurring halfway through the data. Specifically, $p_t = 0.1$ for the first half of the data ($t \leq \lceil T/2 \rceil$) and $p_t = 0.4$ for the second half ($t > \lceil T/2 \rceil$). At test time, the probability is chosen to be $p_{\text{test}} = 0.9$.

Results. We train a convolutional network with four layers, and use the (unit-normalized) output of the second layer as our feature $\phi(x_t)$ for computing the DRM martingale penalty. Fig. 3 (middle) compares ERM and DRM on the training and test distributions. We also present the performance of IRM, which assumes *oracle knowledge* of the specific point in the training data at which the distribution shift occurs. The reversal of the correlation between the label and the color leads to a significant degradation of performance for ERM. In contrast, DRM achieves a performance that is very similar to IRM, without requiring the training data to be separated into different domains.

Visualizing features. In order to obtain more insight into the representations learned by ERM and DRM, Fig. 5 visualizes the features $\phi^{\text{ERM}}(x_t)$ and $\phi^{\text{DRM}}(x_t)$ using their t-SNE embeddings (Maaten & Hinton, 2008). The embeddings are labeled according to the ground-truth labels (blue: 0, orange: 1) for the corresponding input images, along with the color (red: R or green: G) that was applied to the image. The ERM embeddings form two distinct clusters corresponding to the *color* of the image, confirming that ERM learns to rely almost exclusively on the color rather than the shape of the digit. In contrast, the DRM embeddings are separated based on the label rather than the color. The figure shows a grayscale image colored red or green; these images are mapped to an almost identical t-SNE embedding by DRM, suggesting that DRM has learned to ignore the spurious color information.

5.3 Covariate Shift: Imitation Learning

Training and test distributions. For our final example, we consider the imitation learning setting from Fig. 1, which involves covariate shift across environments that the robot is trained and deployed in. The task is to pick up and place a red block into a bowl using observations from an RGB camera. The training data consists of 300 expert demonstrations of pick-and-place locations, which are provided in different environments. A third of the demonstrations are provided with one table-and-bowl color combination, the next third with a slightly different combination, and the final third with another combination; these are visualized in Fig. 1. At test time, the bowl and table background color are changed to a novel combination that significantly exaggerates the variation in green and blue channels seen during training (Fig. 1 right); see Appendix D for RGB values.

Policy training. We utilize the transporter network approach (Zeng et al., 2021), which uses two separate neural networks for picking and placing objects. For simplicity, we adopt the same network architecture for both picking and placing the red block (instead of the key-query placing model in (Zeng et al., 2021)). Each model takes RGB image observations as input. We use residual networks (ResNets) (He et al., 2016) with 36 total layers (convolutional and residual) that form an hourglass encoder-decoder structure. The models are trained to output an image that predicts per-pixel values corresponding to a likelihood that the robot should move to that location for picking / placing. The pick and place models are both trained via a supervised objective in the form of the cross-entropy loss between predicted and demonstrated pick / place locations. We find that the picking network is not impacted by the distribution shifts in table and bowl colors (since it is trained to locate the red block, whose color does not change). As a result, we only apply the DRM objective to the placing network.

Efficacy of detector. Fig. 2 visualizes the martingale values on the training data sequence computed using raw image observations, the features learned via ERM, and the features learned via DRM. As the figure illustrates, the conformal martingale is highly sensitive even to the mild distribution shift that occurs between the first 100 and seconds 100 environments. The martingale value spikes rapidly after the distribution shift for both the raw images and the ERM features. In contrast, the DRM features successfully eliminate the distribution shift from the perspective of the CM.

Results. As shown in Fig. 3 (right), DRM learns a policy that is robust to the distribution shift observed between training and testing. In contrast, the near-perfect training performance of a pure behavior cloning objective (ERM) degrades significantly for test environments.

6 Related work

Distribution shift detection. Traditional methods for distribution shift detection use batch-based statistical hypothesis testing in order to conclude if a distribution shift has occurred between training and test data (Gretton et al., 2012; Rabanser et al., 2019; Kulinski et al., 2020; Farid et al., 2024). In contrast, DRM relies on *online* methods for distribution shift detection, which have been developed relatively recently. These methods are provided with a stream of data, with no demarcation of where a distribution shift may have occurred. In addition to conformal martingales, methods include universal inference (Ramdas et al., 2022), e-processes (Shin et al., 2022), and recency prediction (Luo et al., 2024; Saha & Ramdas, 2024). Theoretical work has characterized the efficiency with which various methods detect distribution shifts (Shin et al., 2022; Ramdas et al., 2022). Our work creates a bridge between the problem of *detecting* distribution shifts and that of *generalizing* to distribution shifts.

Domain generalization, invariance, and causality. Our work is closely related to invariant risk minimization (IRM) (Arjovsky et al., 2019), and the significant amount of subsequent work that it inspired (see, e.g., (Wang et al., 2022; Krueger et al., 2021; Ahuja et al., 2020; Lu et al., 2021)). IRM and its variants seek to find representations that underlie *causal* mechanisms (Schölkopf et al., 2021; Peters et al., 2016; 2017) that generate data. This objective is typically approximated via different regularization schemes (Arjovsky et al., 2019), distributionally robust optimization (Krueger et al., 2021), or via game-theoretic training methods (Ahuja et al., 2020). Practically, the key distinction between IRM and DRM is that we do not assume that data points are associated — either manually or via unsupervised clustering (Le et al., 2025; Murata et al., 2025) — with a finite number of data-generating distributions. This assumption is often impractical or not faithful to reality, e.g., in robotics settings where distribution shifts occur *continuously* as data is being collected (Sinha et al., 2022). Our numerical experiments in Section 5.2 show that DRM can achieve similar performance to IRM without oracular knowledge of distribution shift times. Conceptually, DRM provides a different mechanism for OOD generalization built on the idea of deceiving distribution shift detectors.

Domain adaptation and online adaptation. The objective of aligning training and test distributions also underlies domain adaptation methods, e.g., techniques that align features for training and test distributions (Ben-David et al., 2010; Ganin et al., 2016; Ganin & Lempitsky, 2015; Zhang et al., 2015; Long et al., 2018; Gong et al., 2016; Li et al., 2018; Courty et al., 2016), or ones that re-weight training data points to match the test distribution (Shimodaira, 2000; Huang et al., 2006; Lipton et al., 2018). Domain adaptation methods typically assume that labeled "source" data points are separately identified from unlabeled or sparsely labeled "target" data points that come from the test distribution. In contrast, DRM does not assume that data are separated into different sources. Similar to domain generalization methods (e.g., IRM or its variants), we also do not assume access to data from the particular test distribution of interest.

7 DISCUSSION AND FUTURE WORK

We have introduced *deceptive risk minimization* (DRM): a novel learning objective aimed at identifying stable features that eliminate spurious correlations by hiding distribution shifts from an observer. Our practical instantiation augments a standard ERM loss with a differentiable objective based on conformal martingales. We have provided empirical evidence that DRM can lead to strong generalization to covariate and concept shifts in the presence of spurious correlations in training data. We end with a Q&A discussion on limitations of DRM, potential ways to address them, and other exciting directions for future work. See Appendix E for additional discussion.

Q: When does the DRM objective fail to lead to OOD generalization? Broadly, there are three possible failure modes of DRM. First, it may be possible to find representations that make *training* data appear practically iid, but that do not lead to making the combination of training *and* test data practically iid. This can occur if the axes of variation seen in training data do not span differences between training and test data (e.g., in the imitation learning example from Sec. 5.3, the table and bowl colors were only varied along the blue and green channels, and thus will not lead to generalization when the red channel is altered). Care should be taken to curate training data that span as many relevant axes of variations as possible, even if the *magnitude* of variations is not representative of test data. The second failure mode is when the distribution shift detector we are deceiving is not sufficiently powerful. We expect that continued progress in distribution shift detection will lead to improvements in DRM. Another particularly promising direction is to simultaneously train both the data representation and the detector as an adversarial game. Third, there may be cases where it is not feasible to find representations that eliminate distribution shift in training data, but where one can find invariant predictors as advocated by IRM (Arjovsky et al., 2019, Appendix C). In such cases, DRM is not the right tool. We also note that Rosenfeld et al. (2020) construct data-generating distributions that cause IRM to fail. Since Rosenfeld et al. (2020) consider IRM and related objectives that find invariances across a finite number of data-generating distributions, the results are not directly applicable to DRM. An interesting theoretical direction is to characterize the precise conditions under which a DRM-style objective can lead to OOD generalization. We provide a preliminary sketch of theoretical underpinnings of DRM in Appendix F by connecting deception to generalization.

Q: What are the computational challenges related to implementing DRM? The primary computational bottleneck is in Eq. 3 and Eq. 4, which compute the conformity scores for each example. For each example in the sequence of data points used for distribution shift detection, we compute the minimum distance in embedding space to other examples in the sequence (quadratic complexity). Currently, we address this by sampling subsequences of data from the training sequence, and using these to compute martingale values which are then averaged (Appendix A). Finding strategies to improve this computational bottleneck — perhaps with inspiration from efficient implementations of the quadratic-complexity attention mechanism (Zhuang et al., 2023) — is an important avenue for making DRM scalable.

Q: How sensitive is DRM to different hyperparameters? The primary hyperparameters in DRM are: the dispersion parameter σ for smooth sorting (Sec. 4), the length of the sequences used for distribution shift detection (Sec. A), and the relative weighting λ between the ERM objective and the DRM regularization (Eq. 7). Hyperparameters chosen for the numerical experiments are reported in Appendix B, and we present results from a hyperparameter sweep for the 2D example in Appendix C. We find that DRM is sensitive to the dispersion parameter σ and relatively insensitive to the weighting λ and the length of the detection sequences.

Q: Are there other kinds of distribution shift that could be handled by a DRM-style objective? The two kinds of distribution shift we have considered in this paper are covariate shift and (anti-causal) concept shift. Chapter 8.2 of Vovk et al. (2022) presents a conformal martingale for detecting *label shift*, i.e., a shift in the marginal distribution of class labels. Causal concept shift— a change in the conditional distribution $Y|X$ — is highly relevant in causal inference. In the absence of label shift and covariate shift, anti-causal and causal concept shift are equivalent (via Bayes' rule), as in our examples from Sec. 5.1 and 5.2. Extending DRM to tackle causal concept shift in general settings is an important avenue for future work.

Overall, we are excited by the prospect that the bridge between distribution shift detection and generalization provided by DRM will lead to new techniques that address the problem of OOD generalization, which remains prevalent despite the scale of modern machine learning.

REPRODUCIBILITY STATEMENT

The paper provides all algorithmic details (Sec. 4), implementation details (Appendix A), and hyperparameters (Appendix B) for reproducing results from experiments. In addition, code for all experiments is provided as part of the submission. Results can be reproduced with a single RTX 4090 GPU.

REFERENCES

Muhammad Ahmad and Manuel Mazzara. Scsnet: Sharpened cosine similarity-based neural network for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 21:1–4, 2024.

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010.

Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pp. 950–959. PMLR, 2020.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2016.

Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.

Alec Farid, Sushant Veer, Divyanshu Pachisia, and Anirudha Majumdar. Task-driven detection of distribution shifts with statistical guarantees for robot learning. *IEEE Transactions on Robotics*, 2024.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pp. 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning*, pp. 2839–2848. PMLR, 2016.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems*, 19, 2006.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (REx). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Sean Kulinski, Saurabh Bagchi, and David I Inouye. Feature shift detection: Localizing which features have shifted via conditional distribution tests. *Advances in Neural Information Processing Systems*, 33:19523–19533, 2020.

Phuong Quynh Le, Christin Seifert, and Jörg Schlötterer. Invariant learning with annotation-free environments. *arXiv preprint arXiv:2504.15686*, 2025.

Yann LeCun, Lawrence D Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995.

Kangming Li, Andre Niyongabo Rubungo, Xiangyun Lei, Daniel Persaud, Kamal Choudhary, Brian DeCost, Adji Bousso Dieng, and Jason Hattrick-Simpers. Probing out-of-distribution generalization in machine learning for materials. *Communications Materials*, 6(1):9, 2025.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning*, pp. 3122–3130. PMLR, 2018.

Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems*, 31, 2018.

Chaochao Lu, Yuhuai Wu, Jośe Miguel Hernández-Lobato, and Bernhard Schölkopf. Nonlinear invariant risk minimization: A causal approach. *arXiv preprint arXiv:2102.12353*, 2021.

Rachel Luo, Rohan Sinha, Yixiao Sun, Ali Hindy, Shengjia Zhao, Silvio Savarese, Edward Schmerling, and Marco Pavone. Online distribution shift detection via recency prediction. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16251–16263. IEEE, 2024.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Tomoya Murata, Atsushi Nitanda, and Taiji Suzuki. Clustered invariant risk minimization. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 2016.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT press, 2017.

Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Aaditya Ramdas, Johannes Ruf, Martin Larsson, and Wouter M Koolen. Testing exchangeability: Fork-convexity, supermartingales and e-processes. *International Journal of Approximate Reasoning*, 141:83–109, 2022.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Aytijhya Saha and Aaditya Ramdas. Testing exchangeability by pairwise betting. In *International Conference on Artificial Intelligence and Statistics*, pp. 4915–4923. PMLR, 2024.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

Jaehyeok Shin, Aaditya Ramdas, and Alessandro Rinaldo. E-detectors: A nonparametric framework for sequential change detection. *arXiv preprint arXiv:2203.03532*, 2022.

Rohan Sinha, Apoorva Sharma, Somrita Banerjee, Thomas Lew, Rachel Luo, Spencer M Richards, Yixiao Sun, Edward Schmerling, and Marco Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022.

David Stutz, Ali Taylan Cemgil, Arnaud Doucet, et al. Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*, 2021.

Vladimir Vovk. Testing randomness online. *Statistical Science*, 36(4):595–611, 2021.

Vladimir Vovk, Ilia Nouretdinov, and Alexander Gammerman. Testing exchangeability on-line. In *Proceedings of the International Conference on Machine Learning*, pp. 768–775, 2003.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2022.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8052–8072, 2022.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pp. 726–747. PMLR, 2021.

Kun Zhang, Mingming Gong, and Bernhard Schölkopf. Multi-source domain adaptation: A causal view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Bohan Zhuang, Jing Liu, Zizheng Pan, Haoyu He, Yuetian Weng, and Chunhua Shen. A survey on efficient training of transformers. *arXiv preprint arXiv:2302.01107*, 2023.

A ALGORITHMIC IMPLEMENTATION DETAILS

Algorithm 2 presents the steps for computing the soft martingale regularization for the DRM problem.

Algorithm 2 Computing the DRM regularizer

Input: sequence of features $(\phi(x_t))_{t=1}^T$; choose shift type $s \in \{\text{covariate}, \text{concept}\}$
Output: sequence of soft martingale values $(\tilde{S}_t)_{t=1}^T$
for $t \leftarrow 1\ T$ **do**
  for $i \leftarrow 1\ t$ **do**
    if $s = \text{covariate}$ **then**
      $\tilde{\alpha}_i \leftarrow \text{soft}\big(\alpha_i^{\text{covariate}}\big)$ ▷ replace min with soft-min in equation 3
    else
      $\tilde{\alpha}_i \leftarrow \text{soft}\big(\alpha_i^{\text{concept}}\big)$ ▷ replace min with soft-min in equation 4
    end if
  end for
  if $s = \text{covariate}$ **then**
    $\tilde{p}_t^{\text{covariate}} \leftarrow \text{soft}\big(p_t^{\text{covariate}}\big)$ ▷ use equation 5 with soft-quantile
  else
    $\tilde{p}_t^{\text{concept}} \leftarrow \text{soft}\big(p_t^{\text{concept}}\big)$ ▷ use equation 6 with soft-quantile
  end if
end for
Compute $(\tilde{S}_t)_{t=1}^T$ using Algorithm 1 with inputs $(\tilde{p}_t^{\text{covariate}})_{t=1}^T$ or $(\tilde{p}_t^{\text{concept}})_{t=1}^T$.

We additionally discuss a few implementation details for DRM.

Multiple detection sequences. In settings where the training data sequence is large, we subsample multiple sequences, compute (soft) martingales for each, and average these to form the regularization term in Eq. 7. This results in improved computational efficiency and robustness compared to computing a single martingale value from the entire training data sequence.

Feature normalization. As described in Sec. 4, we utilize cosine distances to define conformity scores. Since the resulting conformity scores are only sensitive to directional differences between features, we normalize encodings to have unit norm, i.e., $\|\phi(x)\|_2 = 1$.

Warm-starting with ERM. For the Colored-MNIST example (Sec. 5.2), we found that warm-starting DRM with a small number of epochs of ERM helped improve performance. This is consistent with the implementation of invariant risk minimization (IRM) from Arjovsky et al. (2019).

B HYPERPARAMETERS FOR NUMERICAL EXPERIMENTS

Parameter	Toy 2D Example	Colored-MNIST	Imitation Learning
# training examples (T)	2000	2000	300
ERM loss batch size	64	64	64
Size of detection sequence	1000	1000	200
# of detection sequences	1	3	3
Regularization weight (λ)	5e5	5e6	1e4
Dispersion for soft-ranking (σ)	0.001	0.1	0.001
Learning rate	0.005	0.005	0.001
# ERM epochs	0	2	0
# total training epochs	2	3	25

C HYPERPARAMETER SWEEP FOR 2D EXAMPLE

The following table shows success rates for different values of the regularization weight (λ), averaged across 5 training seeds.

	$\lambda =$ 1e4	$\lambda =$ 1e5	$\lambda =$ 5e5	$\lambda =$ 1e6
Success (Train ‖ Test)	0.67 ‖ 0.72	0.74 ‖ 0.57	0.70 ‖ 0.63	0.69 ‖ 0.56

The following table shows success rates for different values of the detection sequence length, averaged across 5 training seeds.

	200	400	600	800	1000
Success (Train ‖ Test)	0.69 ‖ 0.53	0.69 ‖ 0.68	0.71 ‖ 0.63	0.71 ‖ 0.56	0.70 ‖ 0.63

The following table shows success rates for different values of the soft-rank dispersion parameter (σ), averaged across 5 training seeds.

	$\sigma = 0.001$	$\sigma = 0.01$	$\sigma = 0.1$
Success (Train ‖ Test)	0.70 ‖ 0.63	0.65 ‖ 0.56	0.71 ‖ 0.51

D IMITATION LEARNING EXAMPLE DETAILS

The task is to pick up and place a red block into a bowl using observations from an RGB camera. The training data consists of 300 expert demonstrations of pick-and-place locations, which are provided in different environments. A third of the demonstrations are provided with one table-and-bowl color combination (Table RGB: [0, 0.2, 0.7], Bowl RGB: [0, 0, 0.5]), , the next third with a slightly different combination (Table RGB: [0, 0.4, 0.9], Bowl RGB: [0, 0.2, 0.7]), and the final third with another combination (Table RGB: [0, 0.3, 0.6], Bowl RGB: [0, 0.6, 0.3]); these are visualized in Fig. 1. At test time, the bowl and table background color are changed to a novel combination (Table RGB: [0, 0.9, 0.4], Bowl RGB: [0, 0.7, 0.2]) that significantly exaggerates the variation in green and blue channels seen during training (Fig. 1 right).

E ADDITIONAL DISCUSSION

Q: Can other methods be used for distribution shift detection in place of conformal test martingales? In this work, we instantiated DRM using conformal martingales (CMs). This choice was motivated by (i) prior work that demonstrates the ability of CMs to detect distribution shifts rapidly (Vovk et al., 2022), (ii) the ability of CMs to detect different kinds of distribution shifts (e.g., covariate and concept shifts), and (iii) the fact that we can construct a differentiable surrogate for CMs. There is exciting future work in contrasting the theoretical and empirical benefits of utilizing other approaches to distribution shift detection (Sec. 6). An approach based on a different detector may make DRM more computationally efficient.

Q: Could DRM be used for covariate shifts due to compounding errors in imitation learning?
A: One idea is to implement the iterative data collection process in DAGGER (dataset aggregation) (Ross et al., 2011), and use DRM to find features that remain robust to the covariate shift between the states visited in successive iterations. Such a strategy may lead to more robust policies compared to DAGGER, which re-trains the policy by aggregating data across iterations of data collection. Working out the details of such an approach could make for interesting future work.

Q: Could DRM be used for reinforcement learning?
A: One immediate application of DRM in reinforcement learning (RL) is in the setting where one has access to a sequence of Markov decision processes (MDPs) for training (similar to the imitation learning setup considered in Sec. 5.3). In this case, the distribution shift detector can take as input observations from different environments, and the DRM objective would then attempt to learn a policy whose features appear iid across environments.

F DECEIVE TO GENERALIZE: THEORETICAL INTUITIONS

In this section, we draw theoretical connections between the objective of deceiving a distribution shift detector and that of achieving OOD generalization. The connection is made in three parts. First, we demonstrate that if a particular distribution shift detector $\Delta^\star$ can be deceived into concluding that the random variables corresponding to training and test losses are iid, then the expected test loss is very close to the expected training loss. Second, we allow for detectors that take encoded representations $\phi(x)$ as input instead of loss values. Third, we define the Δ-span of a representation learned from training random variables as containing test distributions such that training and test random variables are practically iid. Any test distribution in the span thus has expected test loss close to the expected training loss.

F.1 EFFICIENCY OF DISTRIBUTION SHIFT DETECTION

In Sec. 4, we defined observers Δ in the form of distribution shift detectors that control the false alarm rate (FAR). A detector should also ideally detect distribution shifts as quickly as possible. The notion of efficiency can be formalized by the worse average delay (WAD) of a detector.

Worst average delay (WAD). Suppose that the marginal distributions of the sequence of random variables $(\phi(X_1), \phi(X_2), \dots)$ change at an unknown time ν, referred to as a *changepoint*. The worst average delay (WAD) in detecting the change is (Shin et al., 2022):

$$\sup_{\nu \geq 0} \mathbb{E}[N^\star - \nu | N^\star > \nu], \tag{9}$$

where $N^\star$ is the time at which a distribution shift is declared ($N^\star = \infty$ if a change is never declared).

The following definition formalizes the idea of random variables with a changepoint appearing iid to a given observer. Intuitively, the sequence of random variables is practically iid if the worst average delay in detecting a changepoint is large.

Definition 2 (Practically iid w/ changepoint)**.** *A sequence of random variables $(\phi(X_1), \dots, \phi(X_\nu), \phi(X_{\nu+1}), \dots)$ with changepoint ν is* $(\Delta_\alpha, \epsilon)$-practically iid *if the detector Δ_α with FAR bounded by α has a large WAD in detecting the changepoint: WAD $> (1/\epsilon)\log(1/\alpha)$.*

F.2 CONNECTING DETECTION TO GENERALIZATION

Consider the sequence of input random variables $(X_1, \dots, X_T, X_{T+1} \dots)$ as in Sec. 3, where the changepoint T separates training and test distributions. We will demonstrate that there is an encoding ϕ of inputs and a detector $\Delta^\star_\alpha$ such that if $(\phi(X_1), \dots, \phi(X_T), \phi(X_{T+1}), \dots)$ is $(\Delta^\star_\alpha, \epsilon)$-practically iid, then the expected test loss is close to the expected training loss.

Proposition 1. *Let h be a hypothesis that maps inputs to labels, and consider a binary-valued loss function, i.e., $l(x, h(x)) \in \{0, 1\}, \forall x$. Suppose that the expected loss under the training random variables is bounded as follows:*

$$\mathbb{E}[l(X_t, h(X_t)) | \mathcal{F}_{t-1}] \leq l_{\textit{train}},\ \forall t \leq T, \tag{10}$$

where $\mathcal{F}_{t-1}$ denotes the natural filtration of the data. Consider the sequence of random variables $(X_1, \dots, X_T, \dots)$, where the test random variables $(X_{T+1}, X_{T+2}, \dots)$ are iid. Then the expected test loss is:

$$l_{\textit{test}} := \mathbb{E}[l(X_{T+1}, h(X_{T+1})) | \mathcal{F}_T)]. \tag{11}$$

There exists a detector $\Delta^\star_\alpha$, an encoding function ϕ, and a constant c such that the following result holds in the limit as $\alpha \to 0$. If $(\phi(X_1), \dots, \phi(X_T), \phi(X_{T+1}), \dots)$ are $(\Delta^\star_\alpha, \epsilon)$-practically iid, then:

$$\textit{kl}(l_{\textit{test}} \| l_{\textit{train}}) \leq c\epsilon, \tag{12}$$

where $\textit{kl}(\cdot\|\cdot)$ is the KL-divergence between two Bernoulli random variables with parameters $l_{\textit{test}}$ and $l_{\textit{train}}$.

Proof. Define $\phi : x \mapsto l(x, h(x))$. The random variables $(\phi(X_1), \dots, \phi(X_{T+1}), \dots)$ then correspond to Bernoulli random variables with dependent, time-varying means. In the limit

$\alpha \to 0$, the detector presented by Shin et al. (2022) has FAR bounded by α and achieves a WAD $\leq c\log(1/\alpha)/\text{kl}(l_{\text{test}}\|l_{\text{train}}))$. Now, suppose for contradiction that $\text{kl}(l_{\text{test}}\|l_{\text{train}}) > c\epsilon$. Then, we have WAD $< (1/\epsilon)\log(1/\alpha)$, which contradicts the statement that $(\phi(X_1), \dots, \phi(X_T), \dots)$ are $(\Delta^\star_\alpha, \epsilon)$-practcially iid. □

The practical utility of the detector $\Delta^\star_\alpha$ above is limited since it takes losses as input; because we ultimately rely only on making training data practically iid, $\Delta^\star_\alpha$ can be deceived into not detecting a distribution shift on training data simply by overfitting and driving the loss on all training examples to 0. To address this challenge, we allow for detectors (e.g., based on conformal martingales) that take latent representations $\phi(x) \in \mathbb{R}^d$ as input. The following corollary follows immediately from Proposition 1.

Corollary 1. *Let h_ϕ be a hypothesis with latent encoding ϕ. Consider a detector Δ_α that observes inputs encoded by ϕ, and that is at least as efficient as the detector $\Delta^\star_\alpha$ that relies on loss values, i.e., the WAD of Δ_α for any pre- and post-change distributions is less than or equal to the WAD of the detector $\Delta^\star_\alpha$. Then, there exists a constant c such that the following result holds in the limit as $\alpha \to 0$. If $(\phi(X_1), \dots, \phi(X_T), \phi(X_{T+1}), \dots)$ are $(\Delta_\alpha, \epsilon)$-practically iid, then $\text{kl}(l_{test}\|l_{train}) \leq c\epsilon$.*

The results above rely on having access to test data. Instead, consider a representation ϕ such that the training data sequence $(\phi(x_1), \dots, \phi(x_T))$ is Δ-practically iid, and define the Δ-span of this representation as containing test distributions such that $(\phi(X_1), \dots, \phi(X_T), \phi(X_{T+1}), \dots)$ are $(\Delta_\alpha, \epsilon)$-practically iid. Then, in the limit as $\alpha \to 0$, it follows from the results above that for any test distribution in the Δ-span, $\text{kl}(l_{\text{test}}\|l_{\text{train}}) \leq c\epsilon$.