A. Related Works

The first step in explaining distribution shift is detecting such a shift. Many previous works have worked on this problem via methods such as statistical hypothesis testing of the input features [13, 16, 17], training a domain classifier to test between source and non-source domain samples [11], etc. However, these works' primary purpose is to provide the binary information of whether a shift has occurred or not and leave any post-detection methods up to the user (i.e., debugging and/or likely refitting a model).

In [3, 10], the authors attempt to provide more information via localizing a shift to a subset of features or causal mechanisms. [10] does this by introducing the notion of Feature Shift, which first detects if a shift has occurred and if so, localizes that shift to a specific subset of features which have shifted from source to target. This is defined using a hypothesis test which checks for any discrepancy between the conditional distributions of one feature given the rest for both the source and target distributions. The authors use $\phi_{Fisher}(P_{src}(x_j|\boldsymbol{x}_{-j}), P_{tgt}(x_j|\boldsymbol{x}_{-j}))$, as a measure of conditional divergence and report any features which have a statistically significant conditional shift from source to target. In [3], the authors take a causal approach via individually factoring the source and target distributions into a product of their causal mechanisms (i.e., a variable conditioned on its parents) using a shared causal graph, which is assumed to be known/discoverable. Then, the authors "replace" a subset of causal mechanisms from P_{src} with P_{tgt}, and measure divergence from P_{src} (i.e. measuring how much the subset change affects the source distribution). How much each mechanism contributes to all possible swaps is measured (or approximated), and is deemed to be the amount that node can be "assigned blame" for the causing the change in the distribution. While both of these methods more information about distribution shift, they are mainly detection-based methods (via identifying shifted causal mechanisms or feature-level shifts), unlike an explanatory mapping which helps explain *how* the data has shifted.

The characterization of the problem of distribution shift has been extensively studied [12, 16, 19] via breaking down a joint distribution $P(\boldsymbol{x}, y)$ of features $\boldsymbol{x}$ and outputs y, into conditional factorizations such as $P(y|\boldsymbol{x})P(\boldsymbol{x})$ or $P(\boldsymbol{x}|y)P(y)$. For covariate shift [20] the $P(\boldsymbol{x})$ marginal differs from source to target, but the output conditional remains the same, while label shift (also known as prior probability shift) [11, 23] is when the $P(y)$ marginals differ from source to target, but the full-feature conditional remains the same. In this work, we refer to general problem distribution shift, i.e., a shift in the joint distribution (with no distinction between y and $\boldsymbol{x}$), and leave applications of explaining specific sub-genres of distribution shift to future work.

In contrast with current domain generalization benchmarks (e.g., WILDS [9] and DomainBed [7] benchmarks) which are focused on compiling ML train/test distribution shifts, our goal is understanding the shifts (e.g., for knowledge discovery or appropriate mitigation) rather than performing well under shifts. Thus, we even consider distribution shifts that are artificial yet interesting (like splitting the data on an attribute like gender)—or shifts based on thresholding a simulation parameter. Further, our goal likely requires shifts for which some form of ground truth explanation is known (which allows for validation of generated explanations).

B. Interpretable Transport Sets

A de facto standard practice for explaining distribution shift is comparing the means of the source and the target distributions. The mean shift explanation can be generalized as $\Omega_{\text{vector}} = \{T : T(\boldsymbol{x}) = \boldsymbol{x} + \delta\}$ where δ is a constant vector and mean shift being the specific case where δ is the difference of the source and target means. By letting δ be a function of $\boldsymbol{x}$, which further generalizes the notion of mean shift by allowing each point to move a variable amount per dimension, we arrive at a transport set which includes any possible mapping $T : \mathbb{R}^d \to \mathbb{R}^d$. However, even a simple transport set like Ω_{vector} can yeild uninterpretable mappings in high dimensional regimes (e.g., a shift vector of over 100 dimensions). To combat this, we can regulate the complexity of a mapping by forcing it only move points along a specified number of dimensions. We define this as k-*Sparse Transport*:

k-Sparse Transport: For a given class of transport maps, Ω and a given $k \in \{1, ..., d\}$, we can find a subset $\Omega_{sparse}^{(k)}$ which is the set of transport maps from Ω which only transport points along k dimensions or less. Formally, we define an active set $\mathcal{A}$ to be the set of dimensions along which a given T moves points: $\mathcal{A}(T) \triangleq \{j \in \{1, \dots, d\} : \exists \boldsymbol{x}, T(\boldsymbol{x})_j - x_j \neq 0\}$. Then, we define $\Omega_{sparse}^{(k)} = \{T \in \Omega : |\mathcal{A}(T)| \leq k\}$.

k-sparse transport is most useful in situations where a distribution shift has happened along a subset of dimensions, such as explaining a shift where some sensors in a network are picking up a change in an environment. However, in situations where points shift in different directions based on their original value, e.g., when investigating how a heterogeneous population responded to an advertising campaign, k-sparse transport is not ideal. Thus, we provide a shift explanation which breaks the source and target distributions into k sub-populations and provides a vector-based shift explanation per sub-population. We define this as $k-$*cluster transport*:

k-Cluster Transport Given a $k \in \{1, \dots, d\}$ we define k-cluster transport to be a mapping which moves each point $\boldsymbol{x}$ by constant vector which is specific to $\boldsymbol{x}$'s cluster. More formally, we define a labeling function $\sigma(\boldsymbol{x}; M) \triangleq \arg\min_j \|\boldsymbol{m}_j - \boldsymbol{x}\|_2$, which returns the index of the column in M (i.e., the label of the cluster) which $\boldsymbol{x}$ is closest to. With this, we define $\Omega^{(k)}_{\text{cluster}} = \{T : T(\boldsymbol{x}) = \boldsymbol{x} + \delta_{\sigma(\boldsymbol{x};M)}, M \in \mathbb{R}^{d \times k}, \Delta \in \mathbb{R}^{d \times k}\}$, where δ_j is the j^{th} column of Δ.

Since measuring the exact interpretability of a mapping is heavily context dependent, we can instead use k in the above transport maps to define a partial ordering of interpretability of mappings *within* a class of transport maps. Let k_1 and k_2 be the size of the active sets for k-sparse maps (or the number of clusters for k-cluster maps) of T_1 and T_2 respectively. If $k_1 \leq k_2$, then $\text{Inter}(T_1) \geq \text{Inter}(T_2)$, where $\text{Inter}(T)$ is the interpretability of shift explanation T. For example, we claim the interpretability of a $T_1 \in \Omega^{(k=10)}_{sparse}$ is greater than (or possibly equal to) the interpretability of a $T_2 \in \Omega^{(k=100)}_{sparse}$ since a shift explanation in Ω which moves points along only 10 dimensions is more interpretable than a similar mapping which moves points along 100 dimensions. A similar result can be shown for k-cluster transport since an explanation of how 5 clusters moved under a shift is less complicated than an explanation of how 10 clusters moved. The above method allows us to have a partial ordering on interpretability without having to determine the absolute value of interpretability of a individual explanation T, as this requires expensive context-specific human evaluations, which is out of scope for this paper.

C. Practical Methods for Finding and Validating Shift Explanations

In this section, we discuss practical methods for shift explanations. We first discuss using our k-sparse and k-cluster maps to allow a user to automatically change the level of interpretability of a shift explanation as desired. Coupled with a PercentExplained metric, this gives an operator various levels/complexities of explanation and a way to validate them. Next, we propose a practical approximation to Equation 1, the Interpretable Transport equation, and Sections C.3 and C.4 cover how to find the optimal explanation from $\Omega^{(k)}_{sparse}$ and $\Omega^{(k)}_{cluster}$ for this equation.

C.1. Interpretability as a Hyperparameter

By optimizing Equation 1 we can find the best shift explanation for a given set of interpretable transport maps Ω. However, directly defining a Ω which contains candidate mappings which are guaranteed to be both interpretable and expressive enough to explain a shift can be a difficult task. Thus, we can instead set Ω to be a super-class, such as Ω_{vector} given in Appendix B, and then adjust k until a $\Omega^{(k)}$ is found which matches the needs of the situation. This allows a human operator to request a mapping with better alignment by increasing k, which correspondingly will decrease the mapping's interpretability, or request a more interpretable mapping by decreasing the complexity (i.e., decreasing k) which will decrease the alignment.

To assist an operator in determining if the interpretability hyperparameter should be adjusted, we introduce a *PercentExplained* metric, which we define to be:

$$\text{PercentExplained}(P_{src}, P_{tgt}, T) := \frac{W_2^2(P_{src}, P_{tgt}) - W_2^2(T_\sharp P_{src}, P_{tgt})}{W_2^2(P_{src}, P_{tgt})} \tag{3}$$

where $W_2^2(\cdot, \cdot)$ is the squared Wasserstein-2 distance between two distributions. By rearranging terms (and ignoring the percentage scaling factor) we get $1 - \frac{W_2^2(T_\sharp P_{src}, P_{tgt})}{W_2^2(P_{src}, P_{tgt})}$, which shows this metric's correspondence to the statistics coefficient of determination R^2, where $W_2^2(T_\sharp P_{src}, P_{tgt})$ is analogous to the residual sum of squares and $W_2^2(P_{src}, P_{tgt})$ is similar to the total sum of squares. This gives an approximation of how much a current shift explanation T accurately maps onto a target distribution. This can be seen as a normalization of a mapping's fidelity with the extremes being $T_\sharp P_{src} = P_{tgt}$, which fully captures a shift, and $T = \text{Id}$, which does not move the points at all. When provided this metric along with a shift explanation, an operator can decide whether to accept the explanation (e.g., the PercentExplained is sufficient and T is still interpretable) or reject the explanation and adjust k.

C.2. Empirical Interpretable Transport

Since the divergence term in Equation 1 can be computationally-expensive to optimize over in practice, we suggest an empirical approximation to the interpretable transport solution:

$$\arg\min_{T \in \Omega} \frac{1}{N} \sum_{i=1}^{N} c(\boldsymbol{x}^{(i)}, T(\boldsymbol{x}^{(i)}) + \lambda d(T(\boldsymbol{x}^{(i)}), T_{OT}(\boldsymbol{x}^{(i)})) \tag{4}$$

where d is a distance function such as the l_2 distance or squared euclidean distance. Most notably, the divergence value in Equation 1 is replaced with the sum over distances between $T(\boldsymbol{x})$ and the optimal transport mapping for $\boldsymbol{x}$. This is computationally attractive as the optimal transport solution only needs to be calculated once, rather than calculating the Wasserstein distance once per iteration like in the Interpretable Transport solution (which even if the W-distance is approximated, can be expensive over many iterations). For optimization purposes, this is also reasonable since $\frac{1}{N}\sum_{i=1}^{N} d(T(\boldsymbol{x}^{(i)}), T_{OT}(\boldsymbol{x}^{(i)}))$ upper-bounds $\phi(P_{T(\boldsymbol{x})}, P_{\boldsymbol{y}})$, when $d = \ell_2^2$, $\phi = W_2^2$ and N approaches the population size of P_{src} (proof shown in appendix).

C.3. Finding k-Sparse Maps

Let k be a desired level of interpretability, which for k-sparse maps is equivalent to saying $k = |\mathcal{A}(T)|$, where $\mathcal{A}$ is our active feature set (i.e., the dimensions along which our mapping can shift points). Our goal is to find the optimal k features to include in $\mathcal{A}$ and then find the best transport along those features for a given transport class Ω. A simple (and often ideal) approach to feature selection problem is to select the k features which have the largest shift in their mean from the source distribution to the target distribution; this approach is used throughout this paper. Although the chosen T will depend the optimization over Ω, we provide two closed form solutions which give optimal alignment for a given k under cases where $\Omega = \Omega_{vector}$ and when Ω is all possible mappings. The mapping which gives the best alignment in $\Omega_{vector}^{(k)}$ is k-sparse mean shift, i.e., $T(\boldsymbol{x}) = \boldsymbol{x} + \tilde{\mu}$ where $\tilde{\mu}$ is a vector where the j^{th} coordinate is $[\mu_{tgt} - \mu_{src}]_j$, if $j \in \mathcal{A}$, else, it is 0. When $\Omega^{(k)}$ is all k-sparse functions, the shift explanation which minimizes the distance term in Equation 4 is the k-sparse optimal transport solution which sets each feature in $\mathcal{A}$ to match that of the OT push forward for that feature, i.e., $[T(\boldsymbol{x})]_j = [T_{OT}(\boldsymbol{x})]_j$ if $j \in \mathcal{A}$, else $[\boldsymbol{x}]_j$. The proofs for the two previous claims can be seen in the Appendix.

C.4. Finding k-Cluster Maps

Instead of shifting respective to features, we can define k vector shifts for k groups in our source domain, with the goal of explaining how each group changed from source to target. To do this, we perform *paired* clustering in the source and target domains, so that we can relate a given cluster in P_{src} to its most similar counterpart in P_{tgt} (as opposed to pushing the k clusters in P_{src} onto the entire target domain). With this, we construct M_{src} and M_{tgt} where the k columns of M represent the k cluster means for the source and target distributions, respectively. Then, we define $\Delta = M_{tgt} - M_{src}$ so that each vector shift δ_j is the difference in means between the j^{th} source and the target clusters. In practice, the set of paired clusters can be found by performing clustering in a joint Z space of P_{src} and $P_{T_{OT}(\boldsymbol{x})}$ where the resultant k cluster centroids in this space are of the form $[M_{src}, M_{tgt}]$.

Formally, this is done using the following algorithm:

Algorithm 1 Finding k Paired Clusters

```
**Input:** $X$, $Y$, $k$
$d \leftarrow X.ndim$
$T_{OT} \leftarrow$ OptimalTransportAlg$(X, Y)$ //e.g., Sinkhorn
$Z \leftarrow [X, T_{OT}(X)]$
$Z_{cluster-centroids} \leftarrow$ ClusteringAlg$(Z, k)$ //e.g., k-means
$M_{src} \leftarrow [Z_{cluster-centroids}]_{1:d}$ //slicing column-wise
$M_{tgt} \leftarrow [Z_{cluster-centroids}]_{d:2d}$
**Output:** $M_{src}$, $M_{tgt}$
```

C.5. Proof that the distance in empirical interpretable transport upper-bounds the Wasserstein distance

First, let's remember our empirical method for finding T:

$$\underset{T \in \Omega}{\arg\min}\ \frac{1}{N}\sum_{i}^{N} c(\boldsymbol{x}^{(i)}, T(\boldsymbol{x}^{(i)})) + \lambda d(T(\boldsymbol{x}^{(i)}), T_{OT}(\boldsymbol{x}^{(i)})) \tag{5}$$

where T_{OT} is the optimal transport solution between our source and target domains with the given c cost function. The distance term d on the right-hand side of this equation is assumed to be the ℓ_2 cost or squared euclidean cost, and is an empirical approximation of the divergence term $\phi(P_{T(\boldsymbol{x})}, P_Y)$ in Equation 1, where ϕ is assumed to be the Wasserstein distance, W. We claim this is a reasonable approximation since as N approaches the size of the dataset (or for densities, $\lim_{N\to\infty}$), the distance term becomes the expectation: $\mathbb{E}_{x \sim P_{src}} d(T(\boldsymbol{x}^{(i)}), T_{OT}(\boldsymbol{x}^{(i)}))$ which is an upper-bound on the $W(P_{T(\boldsymbol{x})}, P_Y)$ distance. To show this, we start with the expanded W distance:

$$\begin{aligned}
W(P_{T(\boldsymbol{x})}, P_Y) &= \min_{R \in \Psi} \mathbb{E}_{\boldsymbol{x} \sim P_{src}} d\left(T(\boldsymbol{x}), R(T(\boldsymbol{x}))\right), \quad \Psi := \{R : R_\sharp T(\boldsymbol{x}) = P_Y\} \\
&\leq \mathbb{E}_{\boldsymbol{x} \sim P_{src}} d\left(T(\boldsymbol{x}), R(T(\boldsymbol{x}))\right), \qquad \forall R \in \Psi \\
&\text{If we let } Q = T_{OT} \cdot T^{-1}\text{, and since } Q \in \Psi \text{ we can say} \\
&\leq \mathbb{E}_{\boldsymbol{x} \sim P_{src}} d\left(T(\boldsymbol{x}), Q(T(\boldsymbol{x}))\right) = \mathbb{E}_{\boldsymbol{x} \sim P_{src}} d\left(T(\boldsymbol{x}), T_{OT}(\boldsymbol{x})\right) \\
\implies W(P_{T(\boldsymbol{x})}, P_Y) &\leq \mathbb{E}_{\boldsymbol{x} \sim P_{src}} d\left(T(\boldsymbol{x}), T_{OT}(\boldsymbol{x})\right)
\end{aligned}$$

C.6. Proving the k-sparse optimal transport is the k-sparse transport that minimizes our distance from OT loss

When performing unrestricted k-sparse transport, i.e., where $\Omega^{(k)}_{sparse}$ is any transport which only moves points along k dimensions, the k-sparse optimal transport solution is the exact mapping that minimizes the distance function in the right-hand side of Equation 5 if d is the ℓ_2 distance or squared Euclidean distance. As a reminder, k-sparse optimal transport is: $[T(\boldsymbol{x})]_j = [T_{OT}(\boldsymbol{x})]_j$ if $j \in \mathcal{A}$, else $[\boldsymbol{x}]_j$, where $\mathcal{A}$ is the active set of k dimensions which our k-sparse transport T can move points. Let $\bar{\mathcal{A}}$ be $\mathcal{A}$'s compliment (i.e. the dimensions which are unchanged under T). Let $\boldsymbol{z} = T(\boldsymbol{x})$, $\boldsymbol{z}_{OT} = T_{OT}(\boldsymbol{x})$, and $\boldsymbol{x} \in \mathbb{R}^{n \times d}$. If d is the squared Euclidean distance:

$$\begin{aligned}
d(\boldsymbol{z}, \boldsymbol{z}_{OT}) &= \sum_{j \in [d]} \sum_{i \in [n]} \left(\boldsymbol{z}_{i,j} - \boldsymbol{z}_{OT_{i,j}}\right)^2 \\
&= \sum_{j \in \mathcal{A}} \sum_{i \in [n]} \left(\boldsymbol{z}_{i,j} - \boldsymbol{z}_{OT_{i,j}}\right)^2 + \underbrace{\sum_{j \in \bar{\mathcal{A}}} \sum_{i \in [n]} \left(\boldsymbol{x}_{i,j} - \boldsymbol{z}_{OT_{i,j}}\right)^2}_{=\alpha \text{ , since constant w.r.t T}} \\
&= \sum_{j \in \mathcal{A}} \sum_{i \in [n]} \left(\boldsymbol{z}_{i,j} - \boldsymbol{z}_{OT_{i,j}}\right)^2 + \alpha \\
&\text{now if T is the truncated optimal transport solution, } [\boldsymbol{z}]_j = [\boldsymbol{z}_{OT}]_j \quad \forall j \in \mathcal{A} \\
&= 0 + \alpha
\end{aligned}$$

Since α is the minimum of $d(\boldsymbol{z} - \boldsymbol{z}_{OT})$ for a given $\mathcal{A}$, the truncated optimal transport problem minimizes the $d(T(\boldsymbol{x}^{(i)}), T_{OT}(\boldsymbol{x}^{(i)}))$ distance. This can easily be extended to show that the optimal active set for this case is the one that minimizes α, thus the active set should be the k dimensions which have the largest squared difference between $\boldsymbol{x}$ and $\boldsymbol{z}_{OT}$.

C.7. Proof that k-mean shift is the k-vector shift that yields the best alignment

When performing k-sparse vector transport, i.e., where $\Omega^{(k)}_{vector} = \{T : T(\boldsymbol{x}) = \boldsymbol{x} + \tilde{\delta}\}$ where $\tilde{\delta} = [\delta]_j$ if $j \in \mathcal{A}$ else $[\delta]_j = 0$ and $\delta \in \mathbb{R}^d$, $|\mathcal{A}| \leq k$, the k-sparse mean shift solution is the exact mapping that minimizes the distance function in the right-hand side of Equation 5 when d is the ℓ_1 distance.

$$
\begin{aligned}
d(\boldsymbol{z}, \boldsymbol{z}_{OT}) &= \sum_{j\in[d]} \sum_{i\in[n]} \left(\boldsymbol{z}_{i,j} - \boldsymbol{z}_{OT_{i,j}}\right)^2 \\
&= \sum_{j\in\mathcal{A}} \sum_{i\in[n]} \left(\boldsymbol{z}_{i,j} - \boldsymbol{z}_{OT_{i,j}}\right)^2 + \underbrace{\sum_{j\in\bar{\mathcal{A}}} \sum_{i\in[n]} \left(\boldsymbol{x}_{i,j} - \boldsymbol{z}_{OT_{i,j}}\right)^2}_{=\alpha\text{ , since constant w.r.t T}} \\
&= \sum_{j\in\mathcal{A}} \sum_{i\in[n]} \left(\boldsymbol{z}_{i,j} - \boldsymbol{z}_{OT_{i,j}}\right)^2 + \alpha \\
&= \sum_{j\in\mathcal{A}} \sum_{i\in[n]} \left(\boldsymbol{x}_{i,j} + \delta_j - \boldsymbol{z}_{OT_{i,j}}\right)^2 + \alpha \\
&= \sum_{j\in\mathcal{A}} \sum_{i\in[n]} \left(\boldsymbol{x}_{i,j}^2 + \delta_j^2 + \boldsymbol{z}_{OT_{i,j}}^2 + 2\delta_j(\boldsymbol{x}_{i,j} - \boldsymbol{z}_{OT_{i,j}}) - 2\boldsymbol{z}_{OT_{i,j}}\delta_j - 2\boldsymbol{x}_{i,j}\boldsymbol{z}_{OT_{i,j}}\right) + \alpha
\end{aligned}
$$

Similar to the k-sparse optimal transport solution, we can see that $\mathcal{A}$ should be selected as the k dimensions which have the largest shift, thus minimizing α. The coordinate-wise gradient of the above equation is:

$$
\nabla_{\delta_j}\, d(\boldsymbol{z}, \boldsymbol{z}_{OT}) = \begin{cases} \sum_{i\in[n]} \left(2\delta_j + 2\boldsymbol{x}_{i,j} - 2\boldsymbol{z}_{OT_{i,j}}\right) & j \in \mathcal{A} \\ 0 & j \in \bar{\mathcal{A}} \end{cases}
$$

Now with this we can say:

$$
\begin{aligned}
\nabla_{\delta_{j\in\mathcal{A}}}\, d(\boldsymbol{z}, \boldsymbol{z}_{OT}) &= \sum_{i\in[n]} \left(2\delta_j + 2\boldsymbol{x}_{i,j} - 2\boldsymbol{z}_{OT_{i,j}}\right) \\
&= 2n\delta_j + \sum_{i\in[n]} \left(2\boldsymbol{x}_{i,j} - 2\boldsymbol{z}_{OT_{i,j}}\right) \\
&\text{now let } \delta_j = \delta_j^* \\
0 &= 2n\delta_j^* + \sum_{i\in[n]} \left(2\boldsymbol{x}_{i,j} - 2\boldsymbol{z}_{OT_{i,j}}\right) \\
n\delta_j^* &= \sum_{i\in[n]} \left(\boldsymbol{z}_{OT_{i,j}} - \boldsymbol{x}_{i,j}\right) \\
\delta_j^* &= \frac{1}{n} \sum_{i\in[n]} \left(\boldsymbol{z}_{OT_{i,j}} - \boldsymbol{x}_{i,j}\right) \\
\delta_j^* &= \mu_{\boldsymbol{z}_{OT_j}} - \mu_{\boldsymbol{x}_j}
\end{aligned}
$$

Thus showing the optimal delta vector to minimize k-vector transport is exactly the k-sparse mean shift solution.

D. Additional Experiment Details and Results

Here we provide more raw samples from the ColorMNIST experiment as well as an additional counterfactual example experiment, but this time on a toy dataset (as opposed to the real world experiment seen in subsection 3.2) to illustrate the power of distributional counterfactual examples.

D.1. Additional Counterfactual Example Experiment to Explain a Multi-MNIST shift

As mentioned in subsection 3.2, image-based shifts can be explained by supplying an operator with a set of distributional counterfactual images with the notion that the operator would resolve which semantic features are distribution-specific. Here we provide a toy experiment (as opposed to the real world experiment seen in subsection 3.2) to illustrate the power of distributional counterfactual examples. To do this, we apply the distributional counterfactual example approach to a Multi-MNIST dataset where each sample consists of a row of three randomly selected MNIST digits [6] and is split such that P_{src} consists of all samples where the middle digit is even and zero and P_{tgt} is all samples where the middle digit is odd.

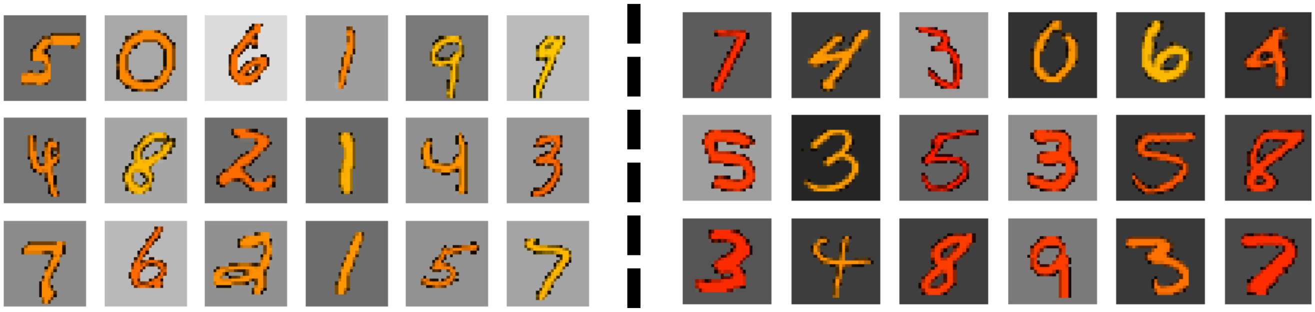

Figure 3. Samples from the source environment (left) with more yellow digits and lighter backgrounds while the target environment (right) has more red digits and/or darker backgrounds.

To generate the counterfactual examples, we use a Domain Invariant Variational Autoencoder (DIVA) [8], which is designed to have three independent latent spaces: one for class information, one for domain-specific information (or in this case, distribution-specific information), and one for any residual information. We trained DIVA on the Shifted Multi-MNIST dataset for 600 epochs with a KL-β value of 10 and latent dimension of 64 for each of the three sub-spaces. Then, for each image counterfactual, we sampled one image from the source and one image from the target and encoded each image into three latent vectors: z_y, z_d, and $z_{residual}$. The latent encoding z_d was then "swapped" between the two encoded images, and the resulting latent vector set was decoded to produce the counterfactual for each image. This process is detailed in Algorithm 2 below. The resulting counterfactuals can be seen in Figure 4 where the middle digit maps from the source (i.e., odd digits) to the target (i.e., even digits) and vice versa while keeping the other content unchanged (i.e., the top and bottom digits).

Algorithm 2 Generating distributional counterfactuals using DIVA

```
Input: $\boldsymbol{x}_1 \sim D_1$, $\boldsymbol{x}_2 \sim D_2$, model
$z_{y_1}, z_{d_1}, z_{residual_1} \leftarrow$ model.encode($\boldsymbol{x}_1$)
$z_{y_2}, z_{d_2}, z_{residual_2} \leftarrow$ model.encode($\boldsymbol{x}_2$)
$\hat{\boldsymbol{x}}_{1\rightarrow 2} \leftarrow$ model.decode($z_{y_1}, z_{d_2}, z_{residual_1}$)
$\hat{\boldsymbol{x}}_{2\rightarrow 1} \leftarrow$ model.decode($z_{y_2}, z_{d_1}, z_{residual_2}$)
Output: $\hat{\boldsymbol{x}}_{1\rightarrow 2}$, $\hat{\boldsymbol{x}}_{2\rightarrow 1}$
```

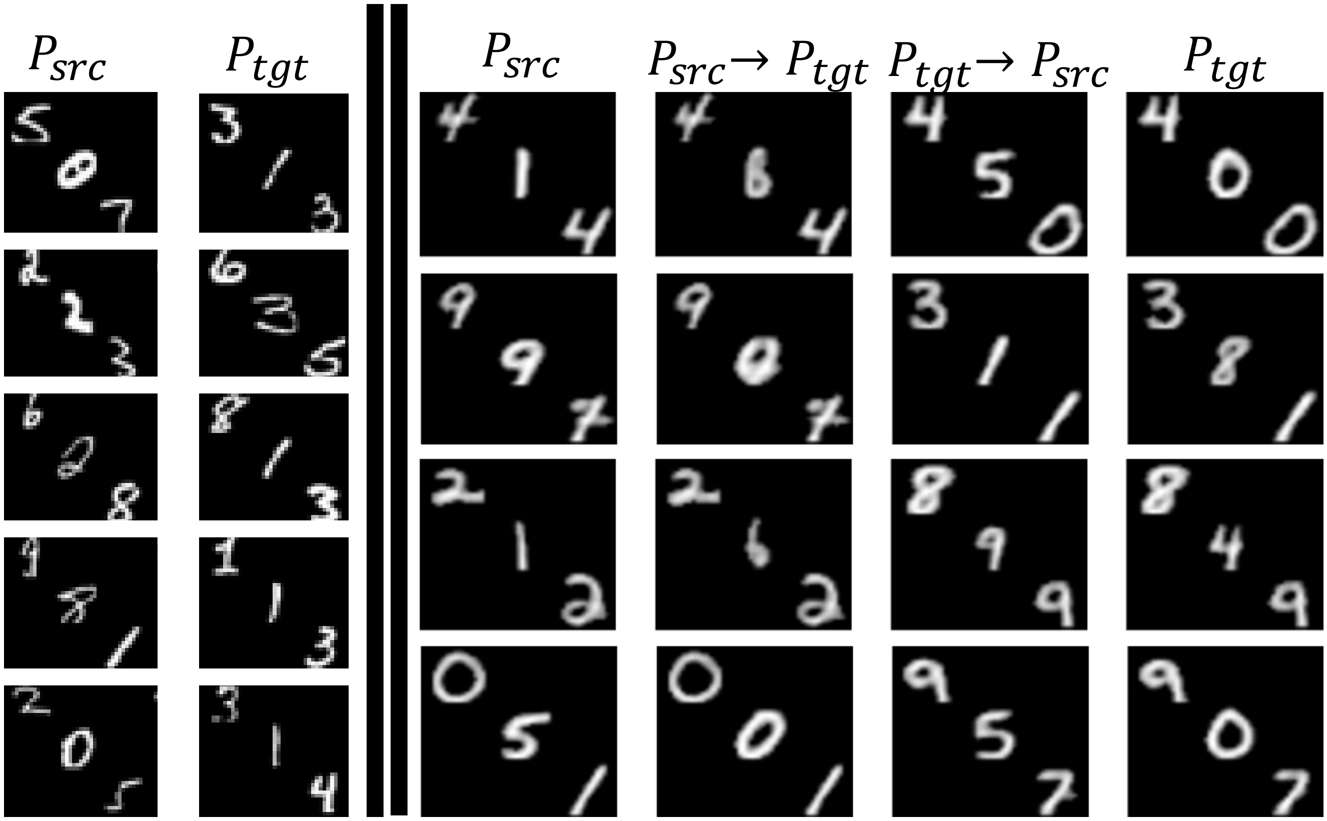

Figure 4. A comparison of the baseline grid of unpaired source and target samples (left) and counterfactual pairs (right) which shows how counterfactual examples can highlight the difference between the two distributions. For each image, the top left digit represents the class label, the middle digit represents the distribution label (where P_{src} only contains even digits and zero and P_{tgt} has odd digits), and the bottom right digit is noise information and is randomly chosen. The second, third columns show the counterfactuals from $P_{src} \rightarrow P_{tgt}$ and $P_{tgt} \rightarrow P_{src}$, respectively. Hence we can see under the push forward of each image the "evenness" of the domain digit changes while the class and noise digits remain unchanged.