Supplementary Materials:
Causal Transportability for Neural Representations

1 Theoretical Results

We will start by introducing the notation and definitions used throughout the paper. In particular, we use capital letters for random variables (Z), and small letters for their values (z). Bolded letters represent sets of random variables and their samples ($Z = \{Z_1, ..., Z_n\}$, $z = \{z_1 \sim Z_1, ..., z_n \sim Z_n\}$). For simplicity, we use the shorthand $P(z_i)$ to represent probabilities $P(Z_i = z_i)$.

The basic semantic framework of our analysis rests on *structural causal models* (SCMs) [4, Ch. 7]. An SCM M is a tuple $\langle V, U, F, P(u) \rangle$ where V is a set of endogenous variables and U is a set of exogenous variables. F is a set of structural functions where $f_V \in F$ decides values of an endogenous variable $V \in V$ taking as argument a combination of other variables. That is, $V \leftarrow f_V(_V, U_V),_V \subseteq V, U_V \subseteq U$. Values of U are drawn from an exogenous distribution $P(u)$. Naturally, SCM M induces a distribution $P(v)$ over endogenous variables V.

An intervention on a subset $X \subseteq V$, denoted by (x), is an operation where values of X are set to constants x, replacing the functions $\{f_X : \forall X \in X\}$ that would normally determine their values. For a detailed survey on SCMs, we refer readers to [Ch. 7]pearl:2k. Each SCM M is associated with a causal diagram G, which is a directed acyclic graph (DAG) where (e.g., see Fig 2) solid nodes represent observed variables O, dashed nodes represent latent variables L, and arrows represent the arguments $_V$ of each functional relationship f_V. Exogenous variables U are not explicitly shown; a bi-directed arrow between nodes V_i and V_j indicates the presence of an unobserved confounder (UC) affecting both V_i and V_j, i.e., $U_{V_i} \cap U_{V_j} \neq \emptyset$. We will use standard family conventions to represent graphical relationships such as parents, children, descendants, and ancestors.

2 The Vision Generalization Problem through Causal Lens

We are interested in learning image recognition models where the input image is denoted as $X = x$ and the outcome category as $Y = y$. In particular, we will analyze the classification task through the causal lens on two different types of generalization tasks—in-distribution generalization and out-of-distribution generalization.

2.1 In-distribution Generalization

To set the stage for our approach, we first consider the standard discriminative classification task. We are given data samples $D(X, Y) = \{(X_i, Y_i)\}_{i=1}^n$ from the probability distribution $P(X, Y)$ from a domain π. We use $D(X, Y)$ to train a classifier that can be written as $P(Y = y|X = x)$. For evaluation, we obtain new samples $D'(X, Y)$ from the same distribution $P(X, Y)$, such that we see x_i and predict the value of the true label y_i through arg $\max_{y'} P(Y = y'|X = x_i)$, as shown in Fig. 1.

Any observational probability distribution comes from an underlying collection of causal processes, which describes the laws of nature and are often unknown. Causal inference formally models this process as a structured causal model (SCM). We write the SCM as $M = \{V, U, F, P(U)\}$ where $V = \{X, Y\}$ are endogenous variables (observed), $U = \{U_x, U_y, U_{xy}\}$ are exogenous variables (unobserved), F are the functional dependencies between the variables, and $P(U)$ is the unknown probability distribution of the exogenous variables. The SCM M induces the probability distribution $P(X, Y)$ described above.

In our setting, the exogenous set U represents all sources of variations that are not captured in both the image and the label Y. In particular, U_{xy} represents the universe of features and objects in the world, and $P(U_{xy})$ represents its

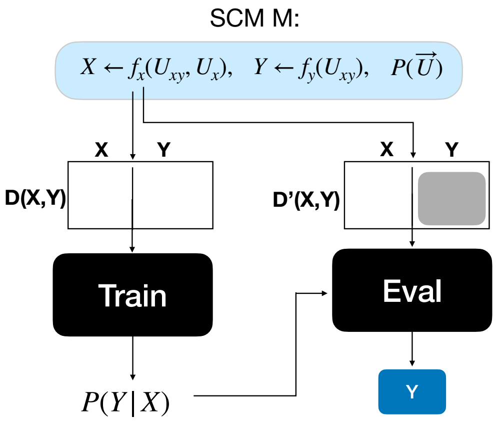
SCM M:
$X \leftarrow f_x(U_{xy}, U_x), \quad Y \leftarrow f_y(U_{xy}), \quad P(\overrightarrow{U})$
X
Y
D(X,Y)
X
Y
D'(X,Y)
Train
Eval
$P(Y|X)$
Y

Figure 1: Unobserved data generating process and sampling/training process. The grey box indicates Y is unknown at inference time.

unknown probability distribution. We will call U_{xy} the concept vector (CV) since it abstracts away all concepts that are out there in the external world affecting the model.

The causal mechanism of the images X is $f_x \in F$ such that an image x is a specific realization of the CV U_{xy}. Furthermore, the label y is also a realization of the CV but in a lower-dimensional space describing the label associated with the image. The instantiation u_{xy} of the CV can encode, for example, the concepts of “yellow” and ”fish”, and through the process f_x, the pixels corresponding to the yellow fish are imprinted in the image x. We note that if f_x selects the color “yellow” and the animal “fish” more likely together, there would be a strong association between these two concepts.

We also model how the labels are formed. Y takes as argument the image x and u_{xy}, and assigns a categorical label to the image, such as “yellow fish.” The unobserved variables U_x and U_y are other sources of independent variations that affect the image and the label. The world repeats the full generative process multiple times, producing a dataset $D(X, Y)$.

The problem of understanding how well one can generalize from specific samples $D(X, Y)$ to a general classifier $P(Y|X)$ while minimizing some measure of error has been well studied in machine learning [8]. As long as the number of samples within $D(X, Y)$ is large enough, the empirical $P(Y|X)$ gets closer to the hypothetical truth.

2.2 Out-of-distribution Generalization

Building off this framework, we are able to analyze the setting where testing images are generated in an out-of-distribution domain π^*, with its own corresponding SCM M^*. In particular, domains π and π^* have different processes by which objects and concepts are selected from the CV (i.e., the mechanism f_x^* is different than f_x).

We make a few assumptions about what is shared between the domains. Firstly, we assume that the labeling is consistent across the two domains and concepts do not change names, which structurally implies that $f_y = f_y^*$. Secondly, we assume the features of objects are consistent across domains, which we write as $P(U_{xy}) = P^*(U_{xy})$.

In realistic out-of-domain generalizations, we do not know the specific form of the structural functions (f_x, f_x^*, f_y) as well as the exogenous distribution $P(U_{xy})$. We leverage a graphical representation to represent the structural invariances across domains, shown in Fig. 2 (right). To model both domains in a single SCM, Bareinboim et al. introduce a *switch node* in the representation, which models the differences between the feature selection functions f_x and f_x^*. This switch node is commonly called an S-node. [1]

If data were available from the target domain π^*, we could have performed a similar process as the one described above and learned a classifier $P^*(Y|X)$. Unfortunately, all that we observe is data from M, i.e. $D(X, Y)$, and we can only train $P(Y|X)$. An illustration of this setting is shown in Fig. 2 (left). One possible route one could take in this setting is to try to use the classifier trained in the original domain π, which is $P(Y|X)$, as it is the same as the hypothetical one $P^*(Y|X)$ trained with the dataset from the target domain π^*. The next result shows that this may be not a good idea.

[1] This S node has been introduced in the context of causal generalizability since at least [1].

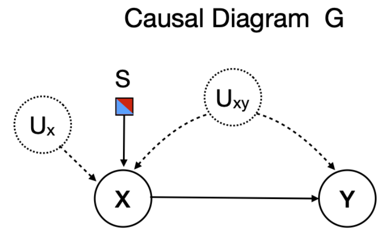

Figure 2: Causal diagram for feature selection.

Proposition 1. *Let M be the SCM relative to domain π that underlines the classifier $P(Y \mid X)$. If we consider another domain π^* with SCM M^* and such that M, M^* are compatible with the assumptions encoded in Fig. 2(b), then $P^*(Y|X) \neq P(Y|X)$. In words, if the selection functions of the domains π, π^* are different ($f_x \neq f_x^*$), and everything else remains invariant, then the corresponding correlation-based prediction models are not transportable across settings.*

Proof. We prove that they are not equal via a counter-example. Assume the first SCM M has $X = U_{xy} \vee S$ and $S = 0$. The second SCM M^* has the switch variable $S = 1$. The remaining features need to be shared, so let $P^{(j)}(U_{xy} = 1) = 1/2$ and $Y = U_{xy} \vee X$. On the training distribution $P(Y = 1|X = 0) = 0$, but on the testing distribution $P^*(Y = 1|X = 0) = 1$. □

In words, this result states that it is impossible to ascertain any performance guarantee about the performance of the classifier $P(Y \mid X)$ when evaluated in the domain π^*. Further assumptions about M^* are needed, such as over the distributions of the exogenous variables or the causal mechanisms.[2]

Against this background, we note something very simple yet powerful. The probability distribution $P(Y \mid X)$ can be seen as describing how Y changes in response to variations in X. In fact, given that the assignment mechanism of X in the domain π is possibly completely different than the one in π^*, nothing can be said about these variations. Still, these variations are realized through two very different mechanisms. First, X variations may affect Y through the direct link $X \rightarrow Y$, which is known as the causal effect of X on Y, $P(Y \mid do(X = x_1)) - P(Y \mid do(X = x_0))$. Second, X may bring about changes in Y through the path going through the unobserved confounders U_{xy} (i.e., $X \leftarrow U_{xy} \rightarrow Y$). These variations are sometimes called spurious effects [9, 5][3]. More specifically, we note that conditioning on X opens the backdoor paths between S and Y in a d-separation sense [4, Ch. 1]. So the sources of spurious variations are not invariant across domains. The next result shows causal effects are invariant across domains.

Proposition 2. *Let M be the SCM relative to domain π that underlines the classifier $P(Y \mid X)$. If we consider another domain π^* with SCM M^* and such that M, M^* are compatible with the assumptions encoded in Fig. 2(b), then*

$$P^*(Y \mid do(X)) = P(Y \mid do(X)) \tag{1}$$

In words, if everything in the domains π, π^* are invariant, but for the selection functions (i.e., $f_x \neq f_x^*$), then the causal effect of X on Y is transportable across settings.

Proof. By the def. of S-node, $P^*(Y|X)$ can be written as $P(Y|X, S = 1)$ []. If we consider the combined causal graphs, it's the case that $(X \perp\!\!\!\perp Y)$ in $G_{\overline{X}}$, which implies the result. □

In words, the causal effects from X to Y are invariant across domains. Intuitively, given that we are talking about the interventional world, where the arrows towards X can be thought as removed, the effect of the selection mechanism is severed as well in both domains. This seems promising since data is available only in domain π, and we are able to connect parts of Y's variations in π^* (l.h.s. in Eq. 1) with the counterpart variations in π (r.h.s.). We then need to compute $P(Y|do(X))$ but only observational data is available in π, $P(X, Y)$.

Proposition 3. *Let M be the SCM relative to domain π and described through causal diagram G in Fig. 2(b) (without the S-node). The interventional distribution $P(Y|do(X))$ is not identifiable from the diagram G and the observational distribution $P(X, Y)$.*

[2] We take a non-parametric approach in the sense of [4], making no assumptions about the SCM's form.

[3] They are written in counterfactual notation as $P(Y \mid X = x_1) - P(Y_{X=x_1}|X = x_0)$, where the latter term reads as the value of Y had X been x_1 when in fact $X = x_0$.

Proof. We prove via a counter-example. Assume the first SCM M has $X = U_{xy}$, $Y = (X \oplus U_{xy}) \wedge U_y$. The second SCM $M*$ has $X = U_{xy}$, $Y = U_y$. For both SCMs, $P(U_i = 1) = 1/2$ and $i = \{x, y, xy\}$. While the observation distribution match for the two SCMs, for the first SCM, $P(Y = 1|do(X = 1)) = 3/4$, for the second SCM, $P*(Y = 1|do(X = 1)) = 1/2$. □

Summary. Proposition 1 shows that no guarantees can be provided when training a classifier in the domain π and using it in another domain π^*. After noting the spurious and causal variations encoded in the classifier, Proposition 2 shows that the causal part of the model is invariant, it is the spurious correlation part that is different across the domains. Despite the causal variations being invariant across π and π^*, Proposition 3 demonstrates that these effects cannot be separated using only the observational data available in π. We discuss in the next section possible methods to circumvent such negative results and the impossibility of generalizing across settings.

Proposition 4. *If the trained proxy variable Z encodes all the variations of X necessary to predict Y (i.e., $Y \perp\!\!\!\perp X|U_{xy}, Z$) and $P(z, x) > 0$, then the effect of X on Y in the proxy model and the original model are the same (i.e., $P(Y|do(X)) = P'(Y|do(X))$).*

Proof. By conditioning on the confounding factor, on the original model G, we have:

$$P(Y|do(X = x)) = \sum_{U_{xy}} P(Y|X = x, U_{xy})P(U_{xy})$$

Given $Y \perp\!\!\!\perp X|U_{xy}, Z$, we have $P(Y|Z = z, U_{xy}) = P(Y|Z = z, U_{xy}, X = x)$.
On the proxy model, we have:

$$\begin{aligned} P'(Y|do(X = x)) &= \sum_z P(Z = z|X = x) \sum_{U_{xy}} P(Y|Z = z, U_{xy})P(U_{xy}) \\ &= \sum_z P(Z = z|X = x) \sum_{U_{xy}} P(Y|Z = z, U_{xy}, X = x)P(U_{xy}) \\ &= \sum_z \sum_{U_{xy}} P(Y, Z = z|U_{xy}, X = x)P(U_{xy}) \\ &= \sum_{U_{xy}} P(Y|X = x, U_{xy})P(U_{xy}) \end{aligned}$$

Thus they are equal. □

Proposition 5. *If the trained proxy variable Z encodes all the variations of X necessary to predict Y (i.e., $Y \perp\!\!\!\perp X|U_{xy}, Z$) and $P(z, x) > 0$, then the effect of X on Y in the proxy model can be identified from observational data [5].*

Proof. $P(Y|do(X = x)) = \sum_z P(z|x) \sum_{u_{xy}} P(y|z, u_{xy})P(u_{xy})$. To remove the U_{xy}, we use two properties that we have when constructing the front-door causal graph.

$$\sum_{u_{xy}} P(y|z, u_{xy})P(u_{xy}) = \sum_x \sum_{u_{xy}} P(y|z, u_{xy})P(u_{xy}|x)P(x)$$

Since $Y \perp\!\!\!\perp X|Z, U_{xy}$, we know $P(y|z, u_{xy}) = P(y|z, u_{xy}, x)$; since $U \perp\!\!\!\perp Z|X$, we have $P(u_{xy}|x, z)$.

$$\sum_{u_{xy}} P(y|z, u_{xy})P(u_{xy}) = \sum_x \sum_{u_{xy}} P(y|z, u_{xy}, x)P(u_{xy}|x, z)P(x) = \sum_x P(y|x, z)P(x)$$

Thus we have

$$P(Y|do(X = x)) = \sum_z P(z|x) \sum_{x'} P(Y|x', z)P(x') \tag{2}$$

□

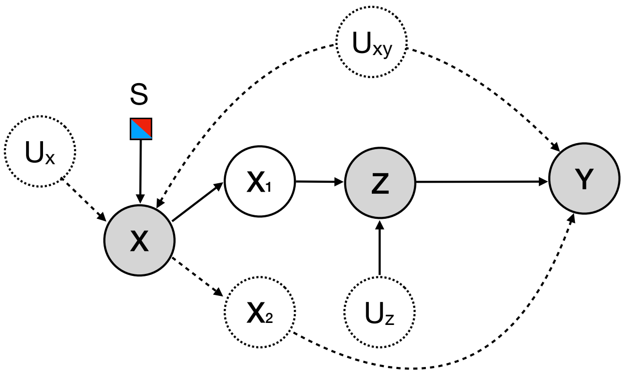

Figure 3: Causal diagram under imperfect front-door variable.

Proposition 6. *In Figure 3, We assume X can be separated into two sub-variables, X_1 and X_2. If the trained proxy variable Z encodes all the variations of $X1$ necessary to predict Y (i.e., $Y \perp\!\!\!\perp X_1|U_{xy}, Z$), $P(z, x_1) > 0$, and $Y \not\perp\!\!\!\perp X_2|U_{xy}, Z$), then our front-door criteria can estimate the causal effect $P(Y|do(X_1))$.*

Proof. $P(Y|do(X_1 = x)) = \sum_z P(z|x) \sum_{u_{xy}} P(y|z, u_{xy})P(u_{xy})$. To remove the U_{xy}, we use two properties that we have when constructing the front-door causal graph.

$$\sum_{u_{xy}} P(y|z, u_{xy})P(u_{xy}) = \sum_x \sum_{u_{xy}} P(y|z, u_{xy})P(u_{xy}|x)P(x)$$

Since $Y \perp\!\!\!\perp X_1|Z, U_{xy}$, we have $Y \perp\!\!\!\perp X|Z, U_{xy}$, thus we know $P(y|z, u_{xy}) = P(y|z, u_{xy}, x)$; since $U \perp\!\!\!\perp Z|X_1$, we have $U \perp\!\!\!\perp Z|X$, thus we know $P(u_{xy}|x, z)$.

$$\sum_{u_{xy}} P(y|z, u_{xy})P(u_{xy}) = \sum_x \sum_{u_{xy}} P(y|z, u_{xy}, x)P(u_{xy}|x, z)P(x) = \sum_x P(y|x, z)P(x)$$

Thus we have

$$P(Y|do(X_1 = x)) = \sum_z P(z|x) \sum_{x'} P(Y|x', z)P(x') \tag{3}$$

Thus we prove that our approach can estimate the causal effect of $P(Y|do(X_1 = x))$. □

3 Implementation Details

We provide the details for our models and training algorithm. We also include our code in the supplementary.

3.1 Colored MNIST

The Colored MNIST is created by adding a background color to the original MNIST dataset [3]. We show example in Figure 4.

The input dimension is 32 × 32. For VAE, we use 2-layer MLP as the encoder and 2-layer MLP as the decoder. The latent dimension is 2048, and the dimension for Z layer is 32. The classification model $P(Y|X', Z)$ is another 2-layer MLP, which takes in 80 dimension vector, which contains 32 elements from the VAE, and 48 elements from subsampled input X' with stride 8. The MLP classifier's latent dimension is 256, and output is 10 dimensions, corresponding to 10 categories in MNIST. We stop the gradient back-propagation from the classifier to VAE.

We train the model with Adam [2] with learning rate $3e - 4$. We first pretrain the VAE model for 3 epoch, and then jointly train VAE and the classifier for another 10 epoch. We provide the code to reproduce the result in supplementary.

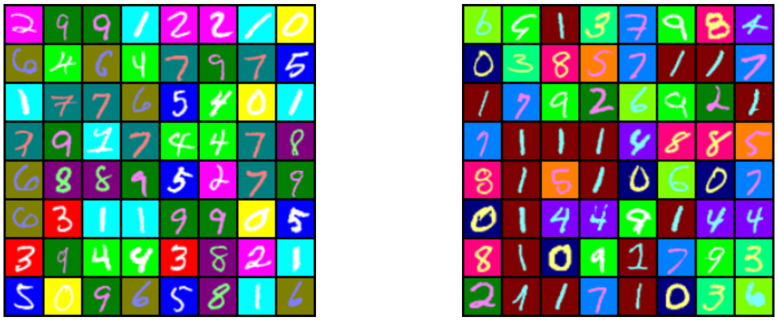

Figure 4: Illustration of color MNIST dataset. For each digit category, we generate two different background colors. The feature background color is spuriously correlated to the category, where the confounder is us, the dataset creator. But the observed data is only color digits and corresponding targets.

Layer	kernel size	stride	channel in	channel out
Conv1	2	4	3	32
Conv2	1	2	3	64
Conv3	1	2	3	128

Table 1: Architecture for convolution branch.

3.2 WaterBird Dataset

The waterbird dataset [6] contains birds in two categories, waterbird and landbird. They are placed on one of the water backgrounds and land backgrounds. The training environment often selects waterbird with water background, and landbird with land background. The average test follows the same distribution as the training set. The worst test environment uses an opposite feature selection function, where the feature of the waterbird is selected with a land background, and the feature of the landbird is selected with a water background.

We use the latent representation from a pretrained ResNet50 to get the mediating variable Z. Our front-door classifier $P(Y|X', Z)$ uses a 3-layer convolution network to process the input X', and then concatenates with the feature with Z. We then use 2-layer fully-connected network to produce a class prediction. The convolutional architecture is shown in Table 1. We use average pooling on top of CNN. We use 1024 for the hidden layer in the 2-layer FC network, and the output dimension is 2. We use a dropout rate of 0.5 to sample from the backbone network Z. We fix the convolution layers during training.

We train the model for 20 epochs. We use Adam optimizer with a learning rate $5e-5$. We train both the ResNet backbone $P(Z|X)$ and our front-door classifier $P(Y|X', Z)$ during our training.

3.3 Experiment on Domain Generalization

We experiment with domain generalization with PACS and VLCS datasets. PACS contains 7 categories from Art painting, Cartoon, Photo, and Sketch domains, and VLCS contains 5 categories from CALTECH, LABELME, PASCAL, and SUN dataset domains. We train on three domains without accessing the domain index, and validate on the fourth domain. We use the Resnet-18 model as our backbone, we use dropout with rate of 0.5 to sample from the penultimate layer of the ResNet model, and get Z. For the front-door classifier $P(Y|X', Z)$, we first encode the X' input with 5 convolution layers, then we flatten the feature and concatenate it with the Z produced by the backbone. We show the architecture in Table 2. We fix this branch without updating so that the features are produced by randomly initialized convolution. We first warm-up for 3 epochs without updating the Resnet-18 backbone and train the model for a total of 100 epochs. We optimize the model with Adam [2] with learning rate $5e-5$. We train the model with batch size 32. For VLCS, we train the model a total of 50 epochs and keep everything else the same. We select the best model with 256 validation examples on the target domain, and report the test accuracy on all the examples on the target domain.

Layer	kernel size	stride	channel in	channel out
Conv1	5	2	3	32
Conv2	3	2	3	64
Conv3	3	2	3	128
Conv4	3	2	3	256
Conv5	3	2	3	512

Table 2: Architecture for convolution branch.

3.4 ImageNet-9

ImageNet-9 studies how much the model relies on the background and foreground to make predictions. It contains 8 variants from the ImageNet dataset with different foreground and background setup. In Table 3 in our paper, 'MixNext' denotes use the background from the next category for the query image. 'MixRand' denotes use the background from a random category for the query image. 'FG' denotes image with only the foreground object. 'Mixsame' denotes use the background from a random image that has the same category as the query image. 'NoFG' means removing the foreground object from the image. 'only BG B' and 'only BG T' denotes two variants of using only background. An ideal model should have low accuracy when there is no background and high accuracy when there is foreground. It addition, it has an adversarial background evaluation setup, where the adversarial images exhaust all possible background and mark it as a right prediction if all the predictions are right. It is a more challenging setup for accessing how much the model relies on the causal foreground signals.

Our front-door model uses the CNN architecture in Table 1 to calculate $P(Y|X, Z)$. We train 10 epoch with Adam, and an learning rate of 0.001. We add Gaussian noise with variance 0.01 to the latent representation Z. We use $N_i = 10$ and $N_j = 256$.

3.5 ImageNet-Rendition

ImageNet-Rendition contains renditions of 200 ImageNet classes with a total of 30,000 images. It includes art, cartoons, deviantart, graffiti, embroidery, graphics, origami, paintings, patterns, plastic objects, plush objects, sculptures, sketches, tattoos, toys, and video game renditions of ImageNet classes. We train the modle in the same way as the ImageNet-9 dataset on ImageNet, and test on ImageNet-Rendition directly.

3.6 ImageNet-Sketch

ImageNet-Sketch data set consists of 50000 images, 50 images for each of the 1000 ImageNet classes. We train the modle in the same way as the ImageNet-9 dataset on ImageNet, and test on ImageNet-Rendition directly.

3.7 Robustness Under Invisible Perturbations

We create two domains for the CIFAR-10 training set by adding a constant vector to the image. The additive vector is tiny to be invisible to humans. It is fixed once it is generated randomly. For each domain, we add an individual vector to each category's images. Thus the additive vector is a spurious feature that correlates with the category but does not cause the category. We remove the additive spurious feature on the test set.

We use Pre-activate ResNet 18 as our backbone, we make the VAE latent space to be the feature after the first block of the ResNet18. We use 3 convolution layers as the decoder layer.

In Table 3, we show classification accuracy on in domain and out of domain performance. As we can see, the front-door method achieves consistent high accuracy on both in domain confounded test set and out-of-domain causal test set.

4 Visualization

	Confounded Test Accuracy	Causal Test Accuracy
Chance	10%	10%
ERM	99.99%	10.87%
Front-door (Ours)	85.29%	**85.23%**

Table 3: We show 10-way classification accuracy on the spurious CIFAR-10 dataset. Additive vector is a spurious correlation we created for the training domain. The spurious correlation no longer holds during the causal test. While ERM is almost random guessing, our approach makes predictions using causal features and achieves high performance.

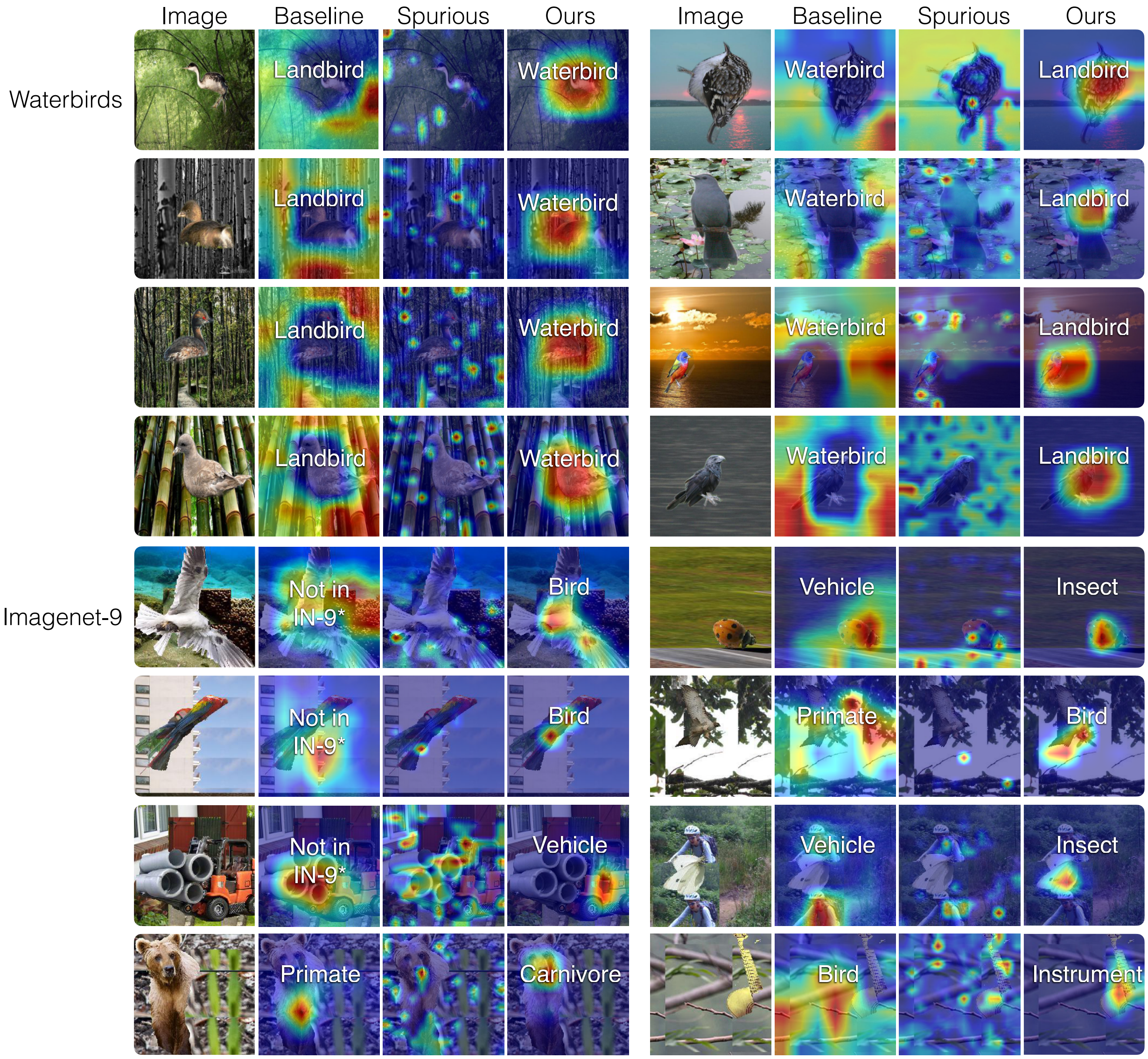

Figure 5: We visualize the input regions that the models use for prediction for the Waterbirds and Imagenet-9 datasets. We use GradCAM [7] and highlight the the discriminative regions that the model relies on with red. The white text shows the model's prediction. The correlation based ERM method often attends to spurious background context. By marginalizing over the spurious features (visualized in the Spurious column), our front-door model captures the correct, causal features, which predicts the correct object for the right reason.

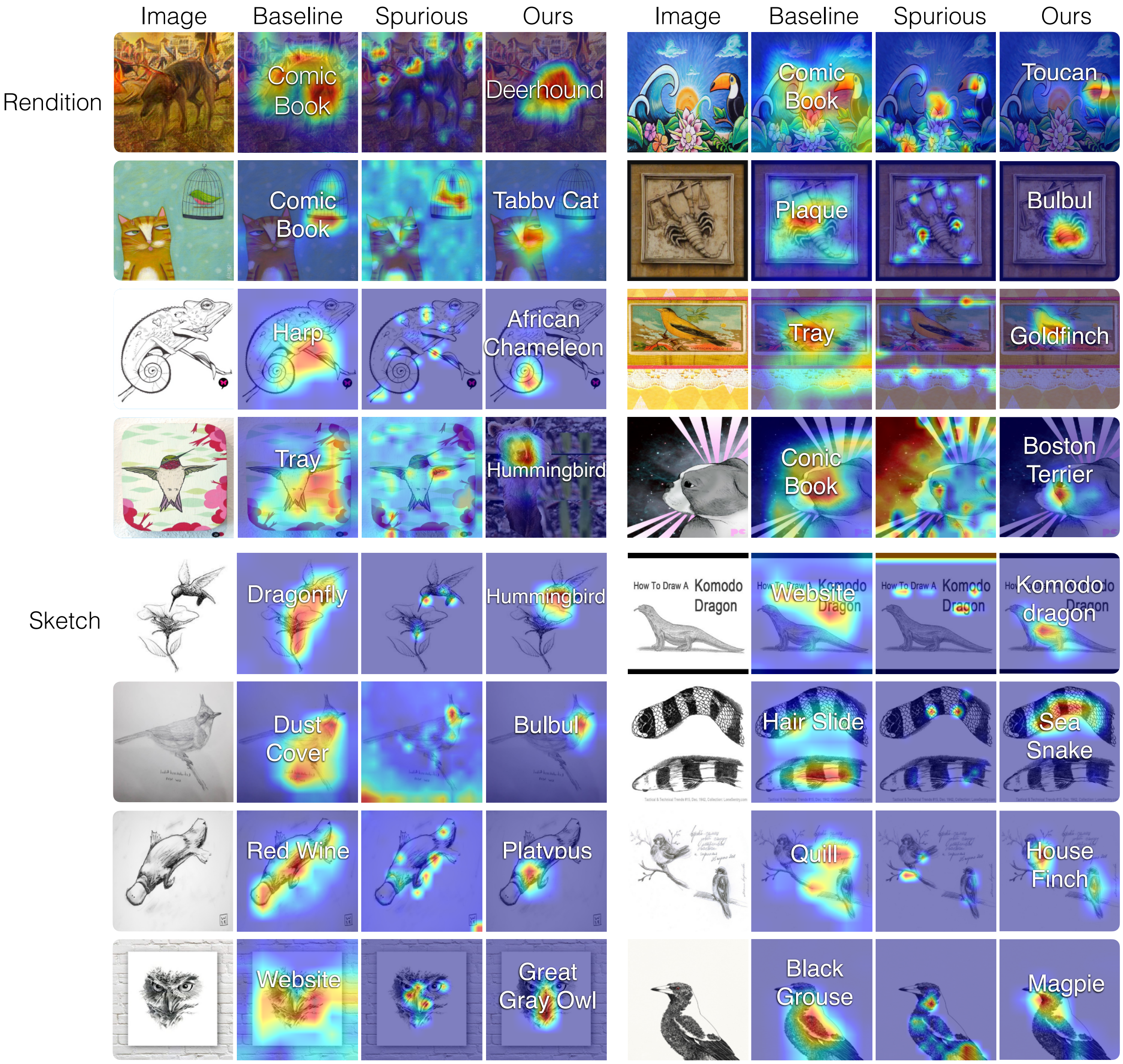

Figure 6: We visualize the input regions that the models use for prediction for the Rendition and Sketch datasets. We use GradCAM [7] and highlight the the discriminative regions that the model relies on with red. The white text shows the model's prediction. The correlation based ERM method often attends to spurious background context. By marginalizing over the spurious features (visualized in the Spurious column), our front-door model captures the correct, causal features, which predicts the correct object for the right reason.

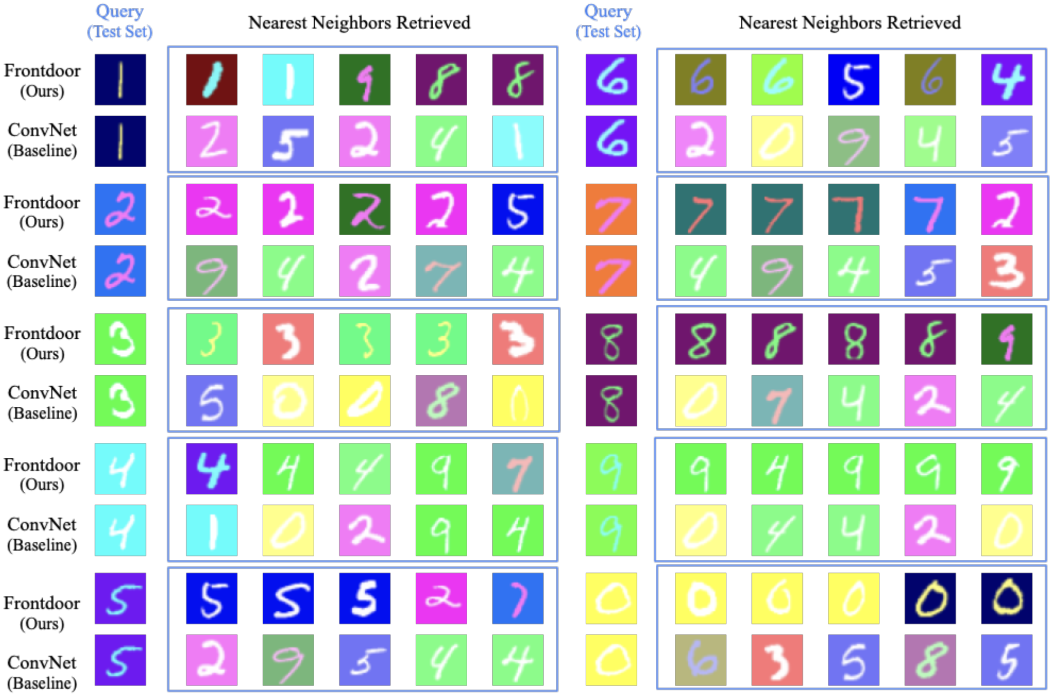

Figure 7: K-Nearest Neighbors Retrieval on Colorful MNIST. The front-door method retrieves neighbors based on the causal information instead of the spurious background color or the style.

References

[1] Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[3] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[4] Judea Pearl. Causality: Models, reasoning, and inference, 2000.

[5] Judea Pearl. Detecting latent heterogeneity. *Sociological Methods and Research*, pages 1–20 (online), DOI: 10.1177/0049124115600597, 2015.

[6] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization, 2020.

[7] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[8] Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

[9] J. Zhang and E. Bareinboim. Fairness in decision-making–the causal explanation formula. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 2037–2045, New Orleans, LA, 2018. AAAI Press.