

Explaining Classifiers with Causal Concept Effect (CaCE)

Yash Goyal*
Georgia Tech
ygoyal@gatech.edu

Uri Shalit
Technion
urishalit@technion.ac.il

Been Kim
Google Brain
beenkim@google.com

Abstract

How can we understand classification decisions made by deep neural nets? We propose answering this question by using ideas from causal inference. We define the “Causal Concept Effect” (CaCE) as the causal effect that the presence or absence of a concept has on the prediction of a given deep neural net. We then use this measure as a mean to understand what drives the network’s prediction and what does not. Yet many existing interpretability methods rely solely on correlations, resulting in potentially misleading explanations. We show how CaCE can avoid such mistakes. In high-risk domains such as medicine, knowing the root cause of the prediction is crucial. If we knew that the network’s prediction was caused by arbitrary concepts such as the lighting conditions in an X-ray room instead of medically meaningful concept, this would prevent us from disastrous deployment of such model. Estimating CaCE is difficult in situations where we cannot easily simulate the *do*-operator. As a simple solution, we propose learning a generative model, specifically a Variational AutoEncoder (VAE) on image pixels or image embeddings extracted from the classifier to measure VAE-CaCE. We show that VAE-CaCE is able to correctly estimate the true causal effect as compared to other baselines in controlled settings with synthetic and semi-natural high dimensional images.

1 Introduction

The rise of machine learning use in many applications brought us a new challenge: how to interpret and understand the reason behind a model’s prediction. Particularly in high-risk domains such as medicine it has been widely recognized that understanding the model’s reasoning for a prediction would be one of the crucial components for wide and safe adoption of the technology.

The machine learning community has been responding to this demand. Many approaches have been proposed to tackle this challenge: for example by developing a model with interpretable components built-in [8, 16] or by building post-training interpretability methods [13, 14, 4, 2, 1, 9]. While these methods may be useful in showing features or concepts that are correlated with a model’s prediction, their explanations might be confounded by correlations present in the data which are not actually relevant to the model, as we describe now.

Say we wish to explain what drives classification decisions for an entire class by a deep neural network, e.g. “what drives the decision to classify an image as BICYCLE”. Now consider the following case: within the training dataset, there is 0.8 correlation between the presence of cars and the presence of bicycles. However, the dataset is diverse enough and the classifier is powerful enough such that it does *not* rely on the presence of cars in order to classify bicycles: If we were to take images of bicycles and edit out the cars, we will find that the classifiers output for the label BICYCLE

*This work was done while Yash Goyal was interning at Google Brain.

is virtually unchanged. Even so, as we show below, the strong correlation between cars and bicycles can lead many interpretability methods to wrongfully give the concept CAR as an explanation for classifying bicycles. In this work, we attempt to tease out the causal aspect: does the presence of a concept like CAR actually change the classifier’s output. The case of “editing out the cars” is an example of what is known as the *do*-operator [12]: it formalizes the act of intervening in the world, an act which lies in the heart of defining and understanding causal effects.

The importance of causal explanations cannot be overemphasized. In medicine, many features might be correlated with the diagnosis (prediction) such as economic status or age, while the true cause might be something treatable - drinking from different water source. The correlated explanations would be an unfortunate distraction from the real cause.

In this paper we propose explaining classifiers with the Causal Concept Effect (CaCE) for high-level concepts whose presence or absence (everything else being equal) affect the model’s prediction, as opposed to merely being correlated with the model’s prediction. CaCE is particularly useful for global explanation methods, where the goal is to explain a model’s prediction for an entire class, rather than individual data points (i.e., local explanation methods). As global methods aim to summarize all data points, they are much more vulnerable to confounding of concepts. By concept, we mean a higher level unit than individual input features with coherent semantic meanings (e.g., cat pixels are cat concepts, where individual pixels are not). The example of the cars and bicycles above illustrates the issue. More generally, concepts are often highly correlated with each other in datasets, and we want our explanations to zero-in on the concepts whose presence or absence in isolation causally affects the model’s output.

One of the challenges in estimating CaCE for high dimensional data such as images is that there is no way to directly control for all possible confounding factors. We propose a method to partially address this challenge using conditional VAEs [15] trained on the training data of the classifier of interest. We also leverage the fact that we do not always have to generate the pixels of the images, but instead we can generate lower-dimensional image embeddings and still calculate the causal effect. We show our approach can approximate the true CaCE for a simple synthetic dataset and a semi-natural images dataset, where in both the cases we know the ground truth CaCE.

Our main contributions are the following:

- We propose a general framework for the Causal Concept Effect (CaCE) that provides causal explanations for model’s prediction.
- We show conditional generative models can be learned on image pixels as well as image embeddings to estimate the true CaCE.
- We demonstrate the effectiveness of our approach in measuring CaCE for a high dimensional image dataset.

2 Related Work

Using causal language for explanation has a long history, see [17] for a discussion from a philosophy of science point of view. [6] give a formal causal theory of what constitutes an explanation, in terms of what is known as “actual causality”. In this paper we use a much narrower notion of explanation than the one they give – we use the causal effect of a concept on the output of a given model as a form of explanation in and of itself.

Recently developed interpretability methods typically fall into two categories: *global* and *local* [3]. Global explanations explain how a model classifies an entire class of objects. Local explanations explain how a model classifies a single image, and answer questions such as “which part of the image is most responsible for the classification output?”. While local explanations are important for investigating individual data points, upon making decisions to deploy an ML model or not, global explanations provide more succinct information.

Our work identifies and then shows a path towards correcting a major flaw in most global interpretation methods: the problem of confounded concepts. By concept, we mean a higher level unit than individual input features that has coherent semantic meanings (e.g., cat pixels are cat concepts, where individual pixels are not). As concepts are often highly correlated with each other in the data (e.g., cars and roads often co-occur), and we want our explanations to zero-in on the concepts whose presence or absence causally affects the model’s output. We compare our work with recently

developed method [9] that offer concept-based global explanations, but suffers from confounding of concepts, providing potentially misleading, where concepts that are merely correlated with the causal concepts can come up as equally valid explanations.

We note that this problem does not exist as such for most local interpretation methods: because for a given image, the pixels deterministically cause the output of a model, there is no notion of probability or confounding. However, confounding might affect local models where pixels are perturbed based on data-dependent models (e.g. [2, 4, 1]). We leave these cases for future work.

Many interpretability methods developed to have causal-flavor are for *local* explanations, such as removing and adding pixels to generate counterfactual explanations for images [5, 1] or for texts [7]. In particular, [1] used the language of counterfactuals to generate *local* explanations. In addition to local and global differences, our work and these prior works face different sets of challenges: performing the *do*-operation with pixels merely involves changing specific pixels. However, the space of possible operations (combinations of pixels) is huge, as there are millions of pixels, each attaining one of hundreds of values. On other hand, realizing *do*-operation on concepts is not trivial as it requires some form of data generation process; it is no longer just about changing specific pixels. However, the space of possible operations is much smaller than that of on pixels. Our goal is to generate global concept-based explanations that can succinctly explain the presence or absence of concepts caused the model’s prediction.

3 CaCE: Causal Concept Effect

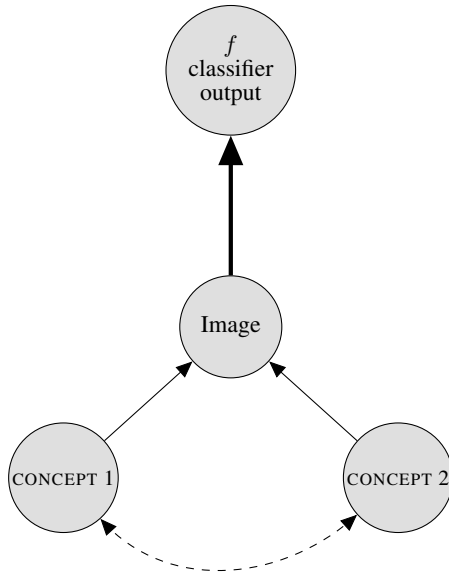


Figure 1: Causal graph relating concepts, images and classifier f output. The dashed edge indicates possible confounding of the two concepts, by other concepts (not shown in the graph). The thick arrow from Image to the output of f indicates that this relation is mechanistic and we have direct access to it through our knowledge of f . This is different from the edges connecting the concepts to the images, which related to the true natural process that gives rise to images. In Section 4 we propose using a conditional-VAE learned on many concepts to approximate this relation.

Denote by I an image, and let $f : I \rightarrow \mathcal{Y}$ be a fixed classifier whose output we wish to explain ². Let C_1, \dots, C_m be concepts which are potential causes of an image: these may be objects, but also concepts such as “cars”, “night time” or “brightness above some threshold L ”.

Consider the process that gives rise to the pixels of a typical natural image: there are many objects in the world, which we consider as concepts. There are also different backgrounds, lighting and angle decisions, as well as properties of the camera and signal processing, all leading to an ordered set of pixels which is an image. These concepts are all considered as *causes* for the image, as shown in Figure 1. We say they cause the image since there is a mechanism at work that, given all these concepts, creates a distribution of images with the relevant concepts. Importantly, concepts are far from independent, as some objects typically come together within a given set of images, as in the car and bicycle example we gave above. Consider the causal graph in Figure 1: the presence or absence of the two concepts $\text{CONCEPT1} = \text{‘BICYCLE’}$ and $\text{CONCEPT2} = \text{‘CAR’}$ together, along with the unobserved factors, give rise

to an image. As is common in causal graphs, the dashed arrows in the graph indicate possible

²For a binary classifier, we typically have $\mathcal{Y} = [0, 1]$.

common confounding by unobserved factors. Suppose we have a classifier f that takes in an image and gives an output in terms of the probability of a set of labels. These labels are often a subset of the set of possible concepts.

Note that for a fixed classifier, the only dependence of its output on the concepts is through the image itself. Moreover, the mechanism leading from image to classifier output is in principle known to us, because we have access to the model f . On the other hand, the mechanism leading from concepts to the image is possibly very complex, and unknown to us in general. In Figure 1 this is denoted by the thick arrow from Image to Classifier Output, as opposed to the thin arrows from the Concepts to the Image.

Definition 1 (Causal Concept Effect, CaCE).

The causal effect of a concept C on the output of classifier f is

$$\text{CaCE}(C, f) = \mathbb{E} [f(I)|do(C = 1)] - \mathbb{E} [f(I)|do(C = 0)] .$$

Note that in causal inference literature, this is simply known as the average treatment effect (ATE) or average causal effect (ACE) of the concept C on the output $f(I)$. The expectation is taken with respect to any chosen distribution of images, not necessarily the one which was used to train f to begin with. The choice of distribution should be made with respect to where do we want to explain the output of f . For example, if the goal is to explain only incorrectly classified examples, the expectation can be taken with respect to those data points.

The major challenge with calculating CaCE is instantiating the *do*-operator. For example, consider the quantity $\mathbb{E} [f(I)|do(\text{BICYCLE} = 1)]$: how can we intervene on an image so that it has a bicycle? One possibility is using a model that can add bicycles to an image. This can range from simply pasting a cropped bicycle image, to using sophisticated generative models such as CausalGAN [10] to add such objects. Note that simply selecting all the images with bicycles is incorrect: this is conditioning on the concept bicycle, and not intervening on it.

In principle one could calculate CaCE by directly manipulating the images. For example, if we wish to understand the causal effect of the overall brightness level, we can simply change the brightness of all images. However, this method does not extend to more complicated concepts. We therefore propose a method for estimating CaCE by counterfactually sampling from a conditional VAE, as we describe below.

Example: Why do we need CaCE? Before presenting our method, let us first address why we need CaCE in the first place using a simple example. We consider an image classification dataset where an image contains exactly one bar, as shown in Figure 2. The orientation of the bar can either be horizontal or vertical, which defines the class label of the image – 0 for horizontal and 1 for vertical. The bar can also take two different colors – red or green. In this case, we consider the color as a binary concept (0 for red and 1 for green) for which we measure our metric “CaCE”. We train a network to classify the images according to the class labels, in two different scenarios. In both scenarios, we can calculate the CaCE exactly, by intervening directly on the images and changing the color of the bars. We can therefore go over all images in the test set, and compare the learned network’s output for the original image and for the image with the flipped color. The sum of these differences (each with appropriate + or – sign) is the CaCE.

We first consider an *unconfounded* scenario when each concept (red or green) is equally balanced with the labels (horizontal and vertical). In other words, each class contains equal number of red and green-colored bars. What we find is that in this case, true CaCE for the color concept is equal to zero, as expected. The network learned to ignore the irrelevant concept (color) when classifying. TCAV score returns the same answer - that the color concept’s does not explain the classification decisions for each class.

We then move to a more interesting case, where 90% of the horizontal bars are red in color (and 10% are green) and only 10% of the vertical bars are red in color (and 90% are green). Now color is a strong confounder for the class. However, we find that with enough training data, the network still learns to ignore the color, and the CaCE of color is 0. However, we find that TCAV score for the color concept in this binary classification is 1.0. TCAV is fooled by the strong correlation between the label and the concept, even though the network itself learned to ignore the color.

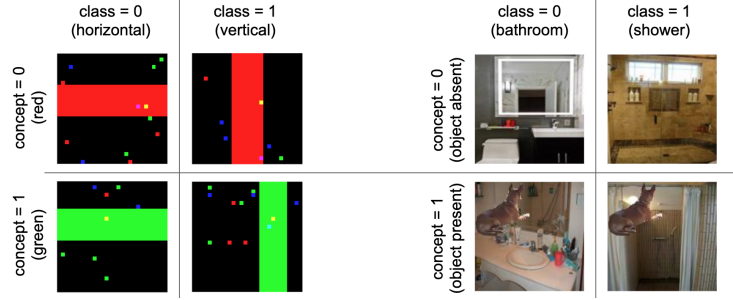


Figure 2: Examples from the two datasets we use in our work— synthetic bars dataset (left) and object+scene images dataset (right) from [18]. We vary the correlation between the class and the concept to construct classifiers with different True-CaCE scores. More details in Sec. 5.

4 Measuring CaCE

In classic tasks of causal inference from observational data, the usual methods for estimating causal effects rely either on adjusting for all confounders, known as backdoor adjustment [12], or on identifying natural experiments and exploiting them, as is the case for instrumental variable analysis.

Let C_0 be the concept we wish to calculate CaCE for, and let \mathcal{C} denote the entire set of concepts. The backdoor adjustment formula gives a way to calculate the interventional distribution $E[f(I)|do(C_0 = c_0)]$ as follows:

$$E[f(I)|do(C_0 = c_0)] = \sum_{\vec{c} \in \mathcal{C} \setminus \{C_0\}} \mathbb{E}[f(I)|C_0 = c_0, C = \vec{c}] p(\vec{c}). \quad (1)$$

The backdoor formula is correct only if we control for all concepts which generate the image and affect concept C_0 . There is no way to guarantee that this is indeed the case. However, we believe that in most cases even controlling for some confounding is better than not controlling at all.

In the following subsections, we first describe how to compute the true measure of CaCE in a controlled setting where we are able to control the data generation process. Then, we propose our approach to estimate CaCE for a more generic setting when we don't have access to the data generation process. Finally, we propose diagnostics for sanity checking whether our approach fails in estimating the CaCE.

4.1 True-CaCE, when you can control the data generation process

Before we present our approach to compute CaCE, let us first talk about the ideal output we want our approach to achieve. CaCE can be computed perfectly if we can intervene in the generation process of the data. We call this measure the *True-CaCE*. Under this ideal scenario we can generate the 'true' counterfactual image for any given image from a dataset by changing the concept of interest, while keeping all the other things in the image fixed. Then, True-CaCE can be computed as the mean difference between the predictions of f on these pairs of an image and its counterfactual.

Since this is a limited scenario and most generation processes in nature don't allow interventions, we propose an approach below to learn a generative model which approximates the true generation process, allowing us to compute an approximation of the True-CaCE. We compare our approach to True-CaCE in controlled experiments.

4.2 VAE-CaCE, when you cannot control the data generation process

In our approach, we use a conditional-VAE to approximate the generation process. We choose a conditional generative model because it allows us to generate examples for a given concept value $C = c$. In addition to conditioning on the concept label, we condition on the class label to be able to generate class-conditional CaCE and to learn a better generative model (in comparison to without conditioning on class label). Hence, our conditional VAE approximates the distribution $p(I|C = c, L = l)$.

In some cases it might be difficult to learn a conditional generative model for image pixels due to the high-dimensionality of the problem. To overcome this issue, we propose learning a conditional VAE on image embeddings (instead of pixels), extracted from an intermediate layer of the classifier. In the rest of the paper, we consider the image pixels as the default input to the conditional-VAE unless we explicitly specify the image embedding.

We use a standard conditional-VAE [15] in our approach. The encoder takes in an image, the class label and the concept label as inputs, create dense embeddings for each of them using convolutional or fully-connected (fc) layers and combines them via concatenation. The combined embedding is further passed through a set of fc layers to output parameters μ and σ of the latent distribution (assumed to be a multivariate Gaussian distribution with diagonal covariance matrix). The decoder takes in a sample z from the latent distribution, the class label and the concept label as inputs, create dense embeddings for each of them using fc layers and combines them via concatenation. The merged embedding is then passed through a set of fc layers followed by a set of deconvolutional layers to output the image reconstruction.

We choose to use a VAE because it allows us to develop the following two methods for approximating Eq. (1) and computing CaCE. These approaches are called Dec-CaCE and EncDec-CaCE. These will approximate CaCE in as much as the conditional-VAE captures the true generative process of the image conditioned on the labels and the concept, and while there are no significant hidden confounders between the set of concepts and labels we use.

4.2.1 Using only the generative network (Dec-CaCE)

Similar to the case of being able to intervene on the generation process (such as Sec.4.1), the generative network of the VAE allows us to sample sets of counterfactual examples from $p(I|z, C = c, L = l)$. We can do this by only changing the value c of the concept C of interest for CaCE, while keeping the class label l and the sampled latent vector z fixed. For a binary concept, CaCE can be computed by averaging the difference in the prediction scores $f(I)$ of the two counterfactual examples – one with $C = 1$ and another with $C = 0$ for many random samples of z .

4.2.2 Using both the inference and generative networks (EncDec-CaCE)

Recall that CaCE computes the causal effect of a concept on the prediction of a class. But often times, we might be interested in measuring the CaCE for a particular image or a specific set of images, e.g., set of images for which the classifier makes incorrect predictions. This is not doable using the “generative-net-only” approach (Dec-CaCE) but can be achieved by utilizing both the inference and the generative nets.

In this approach, EncDec-CaCE, for each given image I_i with the class label l_i and the concept label c_i in the (test) dataset, we can approximately infer the posterior distribution $p(z_i|I_i, l_i, c_i)$ using the inference network, and then sample a counterfactual image from $p(I|z_i, l_i, \tilde{c}_i)$ using the generative network for an alternative value \tilde{c}_i of the concept. We can then measure the causal effect of the concept for a single image i.e., the difference in the prediction scores of the original image and the sampled counterfactual image. This approach allows us to measure CaCE for any set of images by averaging over their individual CaCE values.

4.3 Sanity checks on CaCE

Our method relies on the assumption that hidden confounding does not significantly impact the concepts and labels we use. As is always the case in causal inference, this assumption is not statistically testable without significant assumptions. We thus describe two simple sanity checks for our approach. Note that passing this check does not mean that the estimated CaCE is correct; failing it, however, suggests that the estimated CaCE is probably substantially wrong.

Diagnostic test I: positive effect We suggest estimating the CaCE of the label on the classification output of that same label. Assuming that the classifier f has reasonably good performance classifying the label $l = C_0$, the presence or absence of l should have a strong causal effect on the output. Failure of this diagnostic might mean that the conditional-VAE is weak and does not capture the relation of labels and the image / image embedding.

Diagnostic test II: null effect We suggest estimating the CaCE of a concept which we know should have essentially no effect on the output of f . For example, we can add a random, independent dummy concept with probability 0.5 to each image in the dataset, and estimate its CaCE.

5 Results

In this section, we present experiments showing we can approximate the true CaCE in controlled but challenging settings. We present our approach on two datasets. To be able to evaluate how well our approach works, we have the full data generation process such that we can intervene on the concept label of each image. This allows us to compute the True-CaCE.

In addition to comparing our approach’s estimates of CaCE with the True-CaCE, we compare with: 1) a non-causal baseline and 2) TCAV [9]. A simple baseline is one that computes conditional expectation CONEXP of the prediction scores conditioned on the concept label i.e., same quantity as CaCE but without the do-operators,

$$\text{CONEXP}(C, f) = \mathbb{E}[f(I)|C = 1] - \mathbb{E}[f(I)|C = 0].$$

TCAV, as proposed in [9], computes a global explanation score for a given (concept, label) pair, without taking confounding into account. Moreover, TCAV requires access to internal representations of the image in the classifier while CaCE and our method VAE-CaCE (pixels) can be applied to any black-box classifier.

5.1 Estimating CaCE on Synthetic Dataset

We first use a dataset of simple bar images as described in Section 3 to investigate each proposed method to measure CaCE. Note that in this dataset, the orientation of the bar indicates the class label, and the color of the bar represents the concept we are investigating (Figure 2). We vary how often each concept (color) appears in each class in order to encourage a spectrum of true CaCEs. For each such dataset, we learn a simple binary classifier consisting of 3 fully-connected layers. The results are shown in Table 1. Note that CaCE computes a difference in probability scores, hence CaCE scores for the two classes in a binary classification would sum to zero. Therefore, we only present CaCE scores for class 0 (horizontal) below.

Table 1: CaCE scores for synthetic bars dataset.

% of red in class 0 (horz)	% of red in class 1 (vert)	True-CaCE	Dec-CaCE (pixels)	EncDec- CaCE (pixels)	ConExp (baseline)	TCAV
60	40	0.000	0.021	0.000	0.213	0.766
99	01	0.585	0.592	0.304	1.000	1.000
98	02	0.476	0.491	0.262	0.987	0.967
99	50	0.395	0.444	0.041	0.699	0.968

We first consider the case where 60% of the horizontal bars and 40% of the vertical bars are red in color. In this case, the True-CaCE is zero indicating there is enough diversity in the dataset that changing the color of the bar does not change the prediction of the classifier at all – that means the classifier learned to ignore the not very useful concept of color for this classification task. As we see, all our methods for estimating CaCE correctly estimate it to be very small. On the other hand, approaches that do not take confounding into account incorrectly assign a large importance to the concept, where TCAV seems especially vulnerable.

In the second row we have an opposite case, where the concept label is very highly correlated with the class label. Here we see that the True-CaCE is 0.585 indicating that the classifier heavily relies on the color concept while making predictions. However, this value is far from 1.0 indicating that the color concept is important but not conclusive in its effect on the output of the classifier. Dec-CaCE correctly estimates this number, while our other CaCE estimation methods underestimate the causal effect but still have it far from 0. On the other hand, TCAV and CONEXP interpret color as fully explaining the classifier’s decision, which again we know is incorrect due to our perfect control of the data generating process.

Similar trends can be seen for other settings of correlations between the color concept and the class label (orientation) in rows 3 and 4 of Table 1.

5.2 Estimating CaCE on Natural Images

This dataset, introduced in [18], combines images from Miniplaces [19] and COCO [11] datasets. Each image in this dataset is generated by pasting an object segmentation crop from a COCO image on to a scene image. The scene category defines the class label for the modified image, while the presence or absence of the object crop is a binary concept we want to investigate the causal effect of.

The benefit of this dataset is that it allows us to control the value of the binary concept we are investigating during the generation process. Similar to the bars dataset case, we vary how often the object is pasted on to the images for a class while generating the dataset, resulting in a range of True-CaCE values.

To control for unrelated randomness and focus on investigating the causal effect of the presence and absence of the object, we use a single object instead of a diverse set. We also keep the size of the crop and its location when pasted in the scene image to be fixed.

For simplicity and better understanding of the results, we convert the 100-way classification problem into a binary one by choosing 2 scene categories out of the 100 Miniplaces categories. If we choose the 2 categories randomly, they often are vastly distinct such that the classifier learns to ignore the object and the True-CaCE is always zero. To encourage the classifier to use the presence or absence of the object as a signal for its prediction, we choose the most confusing pair of classes from the dataset – ‘bathroom’ and ‘shower’. For each setting of the dataset, we finetune a ResNet-50 model, pretrained for 100-way classification on Miniplaces dataset, for binary classification between these 2 classes.

The results are shown in Table 2. As in Table 1, we present the CaCE scores for class 0 (‘bathroom’).

Table 2: CaCE scores for natural images dataset

% of obj in ‘bathroom’	% of obj in ‘shower’	True- CaCE	Dec-CaCE (pixels)	EncDec- CaCE (pixels)	Dec-CaCE (feats)	EncDec- CaCE (feats)	ConExp (baseline)	TCAV
60	40	0.077	0.069	0.055	0.073	0.056	0.150	0.750
99	01	0.611	0.542	0.566	0.514	0.423	0.741	1.000
98	02	0.584	0.520	0.557	0.498	0.376	0.711	1.000
95	05	0.489	0.484	0.546	0.437	0.343	0.639	1.000
99	50	0.226	0.173	0.199	0.152	0.082	0.365	1.000

For the first case where 60% of the ‘bathroom’ and 40% of the ‘shower’ images contain the object, the True-CaCE and estimates from all our methods are very small, while *ConExp* and TCAV incorrectly assign a large importance to the concept, consistent with results in Sec. 5.1.

In cases of high correlation (rows 2-5), we observe that all our methods estimate CaCE values close to the True-CaCE values most of the times, with the pixel-based methods naturally closer than the feature-based methods. On the other hand, the baseline CONEXP and TCAV tend to overestimate the importance of the concept, as we would expect when the concept is strongly correlated with the label. As evident from these empirical results, we believe our CaCE estimates can help provide a better understanding of the degree to which correlated concepts actually impact the classifier’s output.

6 Conclusions

The goal of interpretability methods is to help humans make decisions about machine learning models, whether the decision is about deployment in high risk domains, or checking if the model is unfair to a subgroup of people. It is critical that the explanations correctly reflect how the model is making predictions, instead of merely reflecting correlations with predictions. We propose a simple metric CaCE, and show it captures more closely what we expect of explanations of models: the true effect of the absence or presence of a concept on the classifier’s output. We then show how we can estimate CaCE, leveraging the recent development of powerful conditional VAEs. We demonstrate that our method can closely match the true CaCE in controlled settings. Note that while we use VAEs in our work, our approach to calculate CaCE is applicable in combination with any other generative model such as GAN, without many changes.

We hope that CaCE is a starting point towards targeting succinct and causal explanations to unveil the causal processes in classifiers.

References

- [1] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2018.
- [2] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6967–6976, 2017.
- [3] Been Doshi-Velez, Finale; Kim. Towards a rigorous science of interpretable machine learning. In *eprint arXiv:1702.08608*, 2017.
- [4] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.
- [5] Yash Goyal, Ziyang Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, 2019.
- [6] Joseph Y Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part ii: Explanations. *The British journal for the philosophy of science*, 56(4):889–911, 2005.
- [7] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Generating counterfactual explanations with natural language, 2018.
- [8] Been Kim, Cynthia Rudin, and Julie A Shah. The Bayesian Case Model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, pages 1952–1960, 2014.
- [9] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2673–2682, 2018.
- [10] Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017.
- [11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.
- [12] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.
- [14] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- [15] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, 2015.
- [16] B. Ustun and C. Rudin. Methods and models for interpretable linear classification. *ArXiv*, 2014.
- [17] James Woodward. *Making things happen: A theory of causal explanation*. Oxford university press, 2005.
- [18] Sherry Yang and Been Kim. BIM: Towards quantitative evaluation of interpretability methods with ground truth. *ArXiv*, 2019.
- [19] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.