On Generating Plausible Counterfactual and Semi-Factual Explanations for Deep Learning

Eoin M. Kenny[*], **Mark T. Keane**

University College Dublin, Dublin, Ireland
Insight Centre for Data Analytics, UCD, Dublin, Ireland
VistaMilk SFI Research Centre
eoin.kenny@insight-centre.org, mark.keane@ucd.ie

Abstract

There is a growing concern that the recent progress made in AI, especially regarding the predictive competence of deep learning models, will be undermined by a failure to properly explain their operation and outputs. In response to this disquiet, counterfactual explanations have become very popular in eXplainable AI (XAI) due to their asserted computational, psychological, and legal benefits. In contrast however, semi-factuals (which appear to be equally useful) have surprisingly received no attention. Most counterfactual methods address tabular rather than image data, partly because the non-discrete nature of images makes good counterfactuals difficult to define; indeed, generating plausible counterfactual images which lie on the data manifold is also problematic. This paper advances a novel method for generating plausible counterfactuals and semi-factuals for black-box CNN classifiers doing computer vision. The present method, called PlausIble Exceptionality-based Contrastive Explanations (PIECE), modifies all "exceptional" features in a test image to be "normal" from the perspective of the counterfactual class, to generate plausible counterfactual images. Two controlled experiments compare this method to others in the literature, showing that PIECE generates highly plausible counterfactuals (and the best semi-factuals) on several benchmark measures.

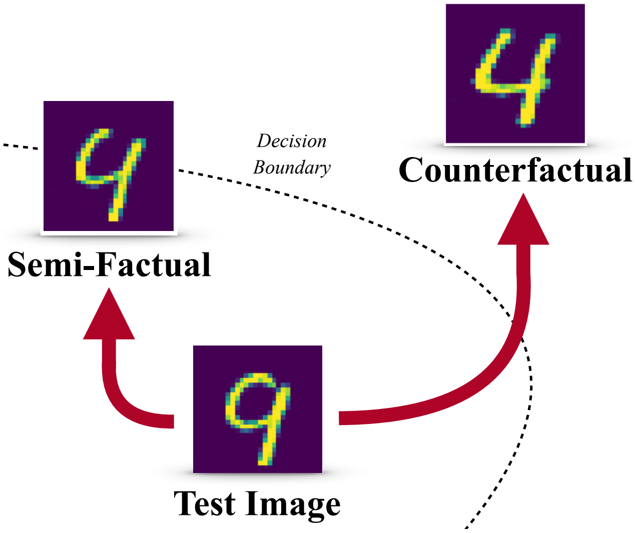

Figure 1: Alternative Explanations: From a given sample test image, PIECE can generate synthetic counterfactual (in a different class) and semi-factual images (within the same class) that are meaningful modifications in the pixel-space, and fall within the data distribution.

Introduction

In the last few years, emerging issues around the the *interpretability* of machine learning models have elicited a major, on-going response from government (Gunning 2017), industry (Pichai 2018), and academia (Miller 2019) on eXplainable AI (XAI) (Guidotti et al. 2018; Adadi and Berrada 2018). As opaque, black-box deep learning models are increasingly being used in the "real world" for high-stakes decision making (e.g., medicine and law), there is a pressing need to give end-users some insight into how these models achieve their predictions. In this paper, we advance a new technique for XAI using counterfactual and semi-factual explanations, applied to deep learning models [i.e., convolutional neural networks (CNNs)]. Recently, the topic of "contrastive explanations" has received considerable interest in AI (Miller 2018; Wachter, Mittelstadt, and Russell 2017), but it tends to focus on counterfactual rather than semi-factual explanation strategies. Counterfactuals have received this attention because they appear to offer computational, psychological, and legal advantages over other explanation strategies; advantages that also appear to accrue to semi-factual explanations (see next section for a review). The code needed to reproduce our algorithm may be found at `https://github.com/EoinKenny/AAAI-2021`.

Contrastive Explanation

Clarifying what constitutes an "explanation" has been an issue for AI, as much as it has been for philosophy and psychology. Here, we follow the proposal that explanations are "contrastive" to convey important causal information about the to-be-explained item (Miller 2019). *Contrastive explanation* is often identified with *counterfactual explanation* (Wachter, Mittelstadt, and Russell 2017), where features that change the original outcome are used to explain

[*] Corresponding Author.

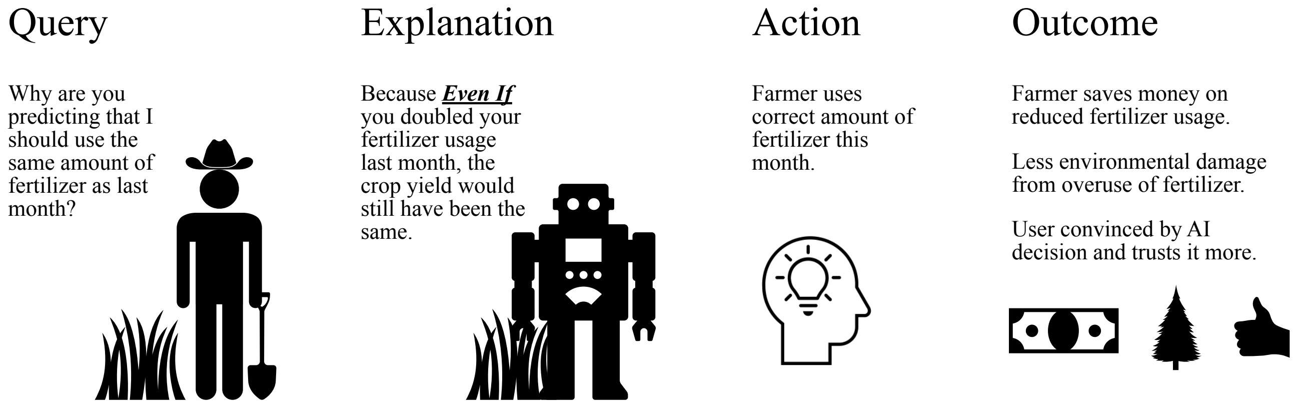
Query
Why are you predicting that I should use the same amount of fertilizer as last month?
Explanation
Because ***Even If*** you doubled your fertilizer usage last month, the crop yield would still have been the same.
Action
Farmer uses correct amount of fertilizer this month.
Outcome
Farmer saves money on reduced fertilizer usage.
Less environmental damage from overuse of fertilizer.
User convinced by AI decision and trusts it more.

Figure 2: A Semi-Factual Explanation in Smart Agriculture: A farmer does not understand the AI Decision Support System's advice on fertilizer-use in the coming month and asks for an explanation. The AI uses a semi-factual explanation to justify the advice and convince the user it is correct about using less fertilizer than the farmer expected to use. The result is an actionable insight that saves the farmer money, improves environmental sustainability, and bolsters trust in the system.

what might have been (e.g., if Hitler had been killed in a Munich street riot in 1930, WWII would not have happened). However, there are other types of contrastive explanation that have received a lot less attention: including, semi-factuals (McCloy and Byrne 2002), pre-factuals (Sanna 1996), and bi-factuals (Miller 2018). Here, we explore semi-factuals which explain how certain features can change *without* changing the original outcome (e.g., even if Hitler had been killed in 1930, WWII would still have happened). We propose an new algorithm for counterfactual explanation, that also can generate semi-factual explanations [what (Nugent, Doyle, and Cunningham 2009) called *a-fortiori* reasoning]. The former involves contrasting explanations that modify the current outcome of a test instance (e.g., the model's prediction changes to another class), whilst the latter involves a contrasting explanation that leaves the outcome intact (i.e., the model's prediction is *not* significantly changed in the explanation generated). Here we consider each of these alternative explanation strategies, in turn.

Counterfactual Explanation

To understand what makes counterfactuals important, consider the difference between factual and counterfactual explanations. An AI loan application system could explain its decision *factually* saying "You were refused because a previous customer with your profile asked for this amount, and was also refused" (Kenny and Keane 2019; Keane and Kenny 2019; Kenny et al. 2020). In contrast, a *counterfactual* explanation could say "If you applied for a slightly lower amount, you would have been accepted". The proponents of counterfactual explanations argue that they have distinctive computational, psychological, and legal benefits. Computationally, counterfactuals provide explanations without having to "open the black box" (McGrath et al. 2018). Psychologically, counterfactuals elicit spontaneous, causal thinking in people, thus making explanations that use them more engaging (Byrne 2019; Miller 2019). Legally, it is argued that counterfactual explanations are GDPR compliant (Wachter, Mittelstadt, and Russell 2017).

Semi-Factual Explanation

Similar arguments can be also be made for the benefits of semi-factuals. In everyday discourse, people typically begin a semi-factual explanation with the words "*Even if...*". So, an AI loan system using semi-factuals might say "*Even if* you had double your current salary, your loan would still have been refused". In some respects, semi-factual explanations appear to have advantages over other explanation types.

Firstly, semi-factual explanations make a prediction seem incontestable and more correct than a factual explanation (Nugent, Doyle, and Cunningham 2009; Byrne 2019). As such, these explanations could be much more convincing; for example, in a SmartAg decision-support system [e.g., (Kenny et al. 2020)], a semi-factual explanation could help convince a user to trust the system whilst also imparting important *causal* information regarding the prediction. Specifically, a farmer could be told "*Even if* you used twice as much fertilizer last month, the crop yield would still have been the same" (see Fig. 2), leading to better farm management and environmental sustainability (Sutton et al. 2011; Kenny et al. 2019).

Secondly, semi-factuals typically work by modifying a *single* feature in the explanation. Counterfactuals often require multiple feature changes to enable the explanation to cross a decision boundary (to change the outcome). This difference is important because it is generally agreed that sparse explanations (with fewer feature-differences) are more comprehensible (Keane and Smyth 2020). So semi-factual explanations produced by an AI system are much more likely to be interpretable than counterfactual ones.

Thirdly, semi-factuals appear to have the advantage of decreasing negative emotions in people by comparison to counterfactuals (McCloy and Byrne 2002), which may give them a role in explanations conveying bad news (e.g., loan rejections or illness diagnoses). The semi-factual tells you there is nothing you could have done to change a bad outcome, whereas the counterfactual potentially *blames you* for not having done something (e.g., "even if you lived healthily you would still have gotten ill" versus "if you had a healthier lifestyle, you would not have gotten ill").

Semi-factuals have been researched for decades in philosophy (Goodman 1983; Bennett 1982; Barker 1991) and psychology (Boninger, Gleicher, and Strathman 1994; Santamaría, Espino, and Byrne 2005; Macbeth and Razumiejczyk 2019; McCloy and Byrne 2002). However, in AI, semi-factual explanations have been largely ignored, even though they can offer good justifications for predictions whilst conveying relevant causal explanatory information (see Fig. 1-3). The closest work we have found in the AI literature, is that on *a-fortiori reasoning* (Nugent, Doyle, and Cunningham 2009; Zurek 2012) where it has been found to better convince people of a classifier's correctness in comparison to factual (i.e., nearest neighbor) explanations (e.g., see Fig. 3). Apart from these two studies, we have not found any other work in AI that explores semi-factual explanation. In *a-fortiori reasoning* one argues that since situation x it true, situation y must be true also. For example, such reasoning might state "Britain cannot afford a space programme, ergo, neither can India". A semi-factual version of this example would state, "*Even if* India was as wealthy as Britain, they still couldn't afford a space program". Computationally, we see these situations as being interchangeable.

Lastly, having argued for the positive aspects of semi-factuals, it should be said there may be negative ones. Semi-factuals can make a prediction seem incontestable (Byrne 2019), and appear to be highly effective in convincing people about AI systems (Nugent, Doyle, and Cunningham 2009). Hence, there is concern that they could be misused to mislead users and engender "inappropriate trust" in a system. So there are ethical issues that need to be flagged (see our Ethics Statement at the end of this paper).

Related Work

Most *post-hoc* explanation-by-example research on counterfactuals has focused on discrete data such as tabular datasets [e.g., see (McGrath et al. 2018)]. These methods aim to generate minimally-different counterfactual instances that can plausibly explain test instances [i.e., instances from a "possible world" (Pawelczyk, Broelemann, and Kasneci 2020)].[1] These counterfactual explanation techniques can be divided into "blind perturbation" and "experience-guided" methods (Keane and Smyth 2020). *Blind perturbation* methods generate candidate counterfactual explanations by perturbing feature values of the test instance to find minimally-different instances from a different/opposing class [e.g., (Wachter, Mittelstadt, and Russell 2017)], using distance metrics to select "close" instances. *Experience-guided* methods rely more directly on the training data by justifying counterfactual selection using training instances (Laugel et al. 2019), analyzing features of the training data (McGrath et al. 2018), or by directly adapting training instances (Keane and Smyth 2020). At present, it is unclear which works best, as there is no agreed standard for computational evaluation, and few papers perform user evaluations [but see (Dodge et al. 2019; Lucic, Haned, and de Rijke 2020)]. With respect to semi-factual explanations, there is one highly relevant paper, using case-based reasoning for *a-fortiori reasoning* (Nugent, Doyle, and Cunningham 2009), but it focuses solely on tabular data.

The applicability of the above techniques to image data remains an open question, largely due to the difference between discrete (e.g., tabular and text) and non-discrete domains (i.e., images). In image datasets, a separate literature examines counterfactuals for adversarial attacks, rather than generating them for XAI. In adversarial attacks, small changes are made (i.e., at the pixel level of an image) to generate synthetic instances to induce misclassifications (Goodfellow, Shlens, and Szegedy 2014). Typically, these micro-level perturbations are constructed to be human-undetectable. In XAI however, counterfactual feature changes need to be human-detectable, comprehensible, and plausible (see Fig. 3c). With this in mind, some notable recent work has used variational autoencoders (VAEs) (Kingma and Welling 2013) and generative adversarial networks (GANs) (Goodfellow et al. 2014) to produce counterfactual images with large featural-changes for XAI. In this literature, the closest related work uses GANs to generate explanations (Samangouei et al. 2018; Seah et al. 2019; Singla et al. 2019; Liu et al. 2019), but only one of these methods is able to offer explanations for pre-trained CNNs in multi-class classification [(Liu et al. 2019); which we compare to our method in Expt. 1]. This preference for binary classification arises partly because choosing a counterfactual class in multi-class classification is non-trivial, and optimization to arbitrary classes is susceptible to local minima (as we shall see, PIECE can solve these problems; see Fig. 6). In addition, none of this previous research has used a method that modifies exceptional features to generate counterfactual explanations, let alone semi-factuals.

Present Contribution. This paper reports PlausIble Exceptionality-based Contrastive Explanations (PIECE), a novel algorithm for generating contrastive explanations for any CNN (both semi-factuals and counterfactuals specifically). PIECE automatically models the distributions of latent features to detect "exceptional features" in a test instance, modifying them to be "normal" in explanation generation. PIECE's explanation generation process can deal with multi-class classification, and is applicable to pre-trained CNNs. Experimental tests show that this method advances the state-of-the-art for counterfactual explanations in quantitative measurements (see Expt. 1). Additionally, PIECE can generate semi-factual explanations (for the first time in deep learning) that are better than benchmark techniques adapted from prior work (see Expt. 2). Alongside our previous work (Kenny and Keane 2019), we see this as completing the algorithmic basis for the three main types of "*post-hoc* explanations-by-example" (Lipton 2018), namely *factual, counterfactual*, and *semi-factual*.

[1] There is a literature using Causal Bayesian Networks to assess fairness of AI systems (Pearl 2000). This is a different use of counterfactuals for another aspect of XAI.

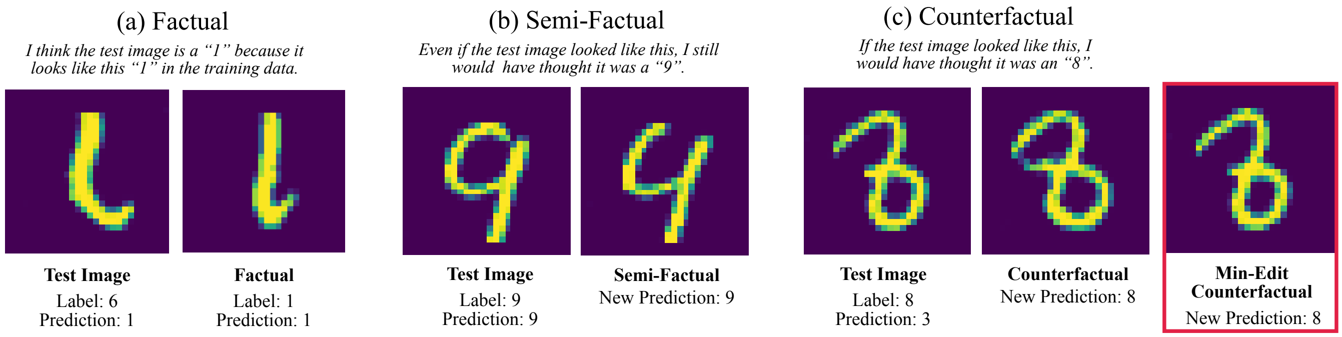

Figure 3: *Post-Hoc* Factual, Semi-Factual, and Counterfactual Explanations on MNIST: (a) a *factual explanation* for a misclassification of "6" as "1" found using the twin-system approach (Kenny and Keane 2019), (b) a *semi-factual explanation* for the correct classification of a "9", that shows a synthetic instance with meaningful feature changes that would *not* alter its classification, and (c) a *counterfactual explanation* for the misclassification of an "8" as a "3", that shows a synthetic instance with meaningful feature changes that would cause the CNN to correct its classification (n.b., for comparison a counterfactual using the *Min-Edit* method (see Expt. 1) is shown with its human-undetectable feature-changes).

PlausIble Exceptionality-based Contrastive Explanations (PIECE)

Plausibility is a major challenge facing contrastive explanations for XAI (Yang et al. 2020). A good counterfactual explanation needs to be plausible, informative, and actionable (Poyiadzi et al. 2020). For example, good counterfactual explanations in a loan application system should not propose implausible feature-changes (e.g., "If you earned $1M more, you would get the loan"). For images, plausible counterfactuals also need to modify human-detectable features (see Fig. 3c); indeed, some methods can generate synthetic instances that are not even within the data distribution (Laugel et al. 2019). Accordingly, an explanation-instance's proximity to the data distribution is now used as a proxy for evaluating plausibility (Van Looveren and Klaise 2019; Samangouei et al. 2018), as used here. PIECE uses an *experience-guided* approach, exploiting the distributional properties of the data to help guarantee plausibility.

Fig. 3 illustrates some of PIECE's plausible contrastive explanations for a CNN's classifications on the MNIST dataset (LeCun, Cortes, and Burges 2010), with a factual explanation provided for comparison (Kenny et al. 2021). In Fig. 3c, the test image of an "8" misclassified as a "3", is shown alongside its counterfactual explanation, showing the feature changes that would cause the CNN to classify it as an "8" (i.e., the cursive stroke making the plausible "8" image). An implausible counterfactual, generated by a minimal-edit method (i.e., the *Min-Edit* method in Expt. 1), is also shown, with human-undetectable feature-changes that would also cause the CNN to classify the image as an "8". Fig 3b shows a semi-factual, with meaningful changes to the test image that do *not* change the CNN's prediction. That is, *even if* the "9" had a very open loop, so it looked more like a "4", the CNN would *still* classify it as a "9". This type of explanation has the potential to convince people that the original classification was definitely correct (Byrne 2019; Nugent, Doyle, and Cunningham 2009), although that is likely less needed in a domain such as MNIST were people are mostly experts. Finally, though these examples show two explanations for incorrect predictions (factual and counterfactual), and one for a correct prediction (semi-factual), these three explanation types may be generated for either predictive outcome.

PIECE generates counterfactuals and semi-factuals by identifying "exceptional" features in the test image, and then modifying these to be "normal". This idea is inspired by people's spontaneous use of counterfactuals, specifically the *exceptionality effect*, were people change exceptional events into what would *normally* have occurred (Byrne 2019; Icard, Kominsky, and Knobe 2017). For example, when people are told that "Bill died in a car crash taking an unusual route home from work", they typically respond counterfactually, saying "if only he had taken his *normal* route home, he might have lived" (Byrne 2016). So, PIECE identifies feature-values in the test image that are probabilistically-low in the counterfactual class (i.e., exceptional features) and modifies them to be their expected values (i.e., normal features) in that counterfactual class (in order from the lowest probability feature); by doing this, PIECE shows how the original test image would have to change to be considered a good semi-factual or example of the counterfactual class.

The Algorithm: PIECE

PIECE involves two distinct systems, a CNN with predictions to be explained, and a GAN that helps generate counterfactual or semi-factual explanatory images (see Section S1 of the supplement for model architectures). This algorithm will work with any CNN post-training, provided there is a GAN trained on the same dataset as the CNN. PIECE has three main steps: (i) "exceptional" features are identified in the CNN for a test image from the perspective of the counterfactual class, (ii) some of these are then modified to be their expected values, and (iii) the resulting latent-feature representation of the explanatory counterfactual is visualized in the pixel-space using the GAN. To produce semi-factuals, the algorithm is identical, but the feature modifications in the second step are stopped prematurely before the model's prediction crosses the counterfactual decision boundary.

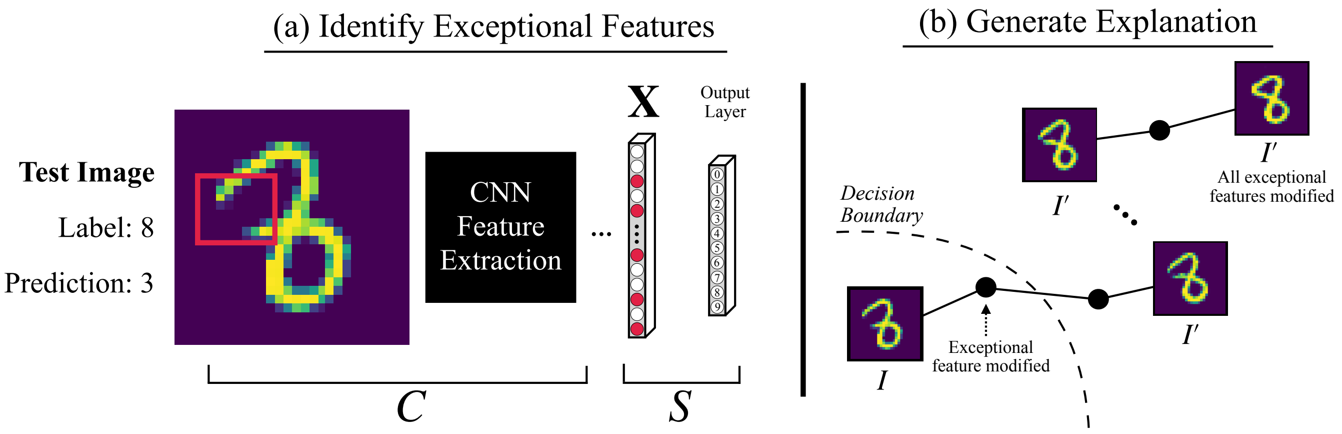

Figure 4: PIECE Counterfactual Explanation for an *Incorrect* Prediction: (a) After the feature extraction half C of the CNN outputs the latent features for image I in layer $\mathbf{X}$ (which is the penultimate layer and part of the output linear classifier S), PIECE identifies the exceptional features for the counterfactual-class "8" in this same layer, these features primarily reside in the region highlighted by the red box in the test image (and are highlighted in red in layer $\mathbf{X}$). (b) These features in image I are then modified to be "normal" from the perspective of the "8" counterfactual-class and the explanation is generated (successive feature-changes to I are shown in the I' images).

Setup and Notation. Allow all layers in a CNN up to just before the penultimate extracted feature layer $\mathbf{X}$ be C, and the output linear classifier S (which includes $\mathbf{X}$; see Fig. 4). The extracted features from a test image I at layer $\mathbf{X}$ will be denoted as x, this connects to an output SoftMax layer to give a probability vector Y which predicts a class c. To denote that c is the class in Y with the largest probability (i.e., the predicted class), Y_c will be used. Let the generator in the GAN be G, and its latent input z. The counterfactuals to a test image I, in class c, with latent features x, are denoted as I', c' and x', respectively.

Identify the Counterfactual Class. The initial steps involve locating a given test image I in G, and then identifying the counterfactual class c'. First, to find the input vector z for G, such that $G(z) \approx I$, we solve the following optimization with gradient descent:

$$z = \arg\min_{z_0} \|C(G(z_0)) - C(I)\|_2^2 + \|G(z_0) - I\|_2^2 \quad (1)$$

where z_0 is a sample from the standard normal distribution. More efficient methods exist to do this involving encoders (Zhu et al. 2020; Seah et al. 2019), but Eq. (1) was sufficient for present purposes. Second, the counterfactual class c' for I may need to be generated for a prediction were the classifier is correct or incorrect. When the CNN is incorrect in classifying I and the label is known, c' can be trivially selected as being the actual label (e.g., see Fig. 4). However, when the classifier is correct (or the label is unknown) for I, identifying c' becomes non-trivial. We use a novel method here involving gradient *ascent* to solve this problem and run:

$$\arg\max_{z} \|S(C(G(z))) - Y_c\|_2^2 \quad (2)$$

where Y_c is binary encoded as all 0s, and a 1 for the class c. During this optimization process, the first time a decision boundary is crossed, the new class is selected as c'. Whilst hard-coding c' can result in the optimization becoming "stuck" (Liu et al. 2019), the present automated method has never failed to generate the desired counterfactual in all of our tests.

Step 1: Identifying Exceptional Features

Here, when the CNN classifies a test image I as class c, taking the perspective of the counterfactual class c', we find exceptional features in x by examining their statistical probabilities in the training distributions for c' (see Fig. 5). So, assuming the use of ReLU activations in $\mathbf{X}$, we can model each neuron $\mathbf{X}_i$ for c', as a hurdle model (note each neuron $\mathbf{X}_i$ will have a hurdle model for each class). A statistical hurdle model is a two-part process that specifies one process for zero counts and another process for positive counts. In this case, we are modelling latent features in $\mathbf{X}$, which use ReLU activation functions. Due to the nature of ReLU, there will be many zero values, alongside positive ones. As such, a hurdle model can deal with the resultant data well, were the probability of the neuron not activating (i.e., a 0 value) or activating (i.e., a value > 0) can be modelled as a Bernoulli distribution (the initial hurdle process), and when the neuron does activate (the second process), the data can be modelled as a probability density function (PDF; see Fig. 5). The hurdle models are defined as:

$$p(x_i) = (1 - \theta_i)\delta_{(x_i)(0)} + \theta_i f_i(x_i), \quad \text{s.t.} \quad x_i \geq 0 \quad (3)$$

where $p(x_i)$ is the probability of the latent feature value x_i for c' in neuron $\mathbf{X}_i$, θ_i is the probability of the neuron $\mathbf{X}_i$ activating for the class c' (i.e., Bernoulli trial success in the hurdle model), f_i is the subsequent PDF modelled for when $x_i > 0$ (i.e., when the "hurdle" is passed), the constraint of $x_i \geq 0$ refers to the ReLU activations, and $\delta_{(x_i)(0)}$ is the Kronecker delta function, returning 0 for $x_i > 0$, and 1 for $x_i = 0$. Moving forward, X_i will signify the random variable associated with f_i.

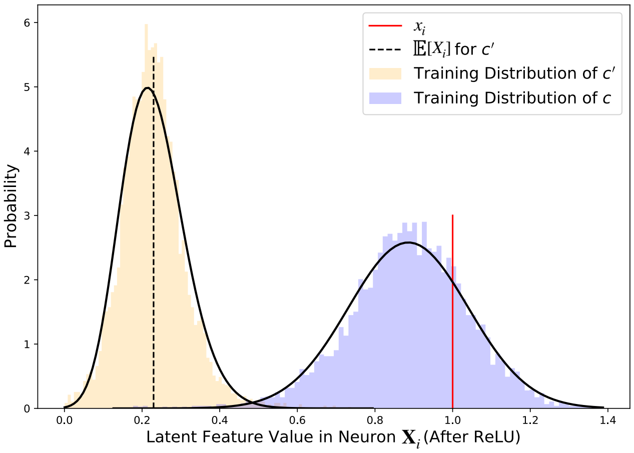

Figure 5: Exceptional Feature Modification [Specifically Eq. (7)]: The second process of two hurdle models is shown fitted to the data. A high-probability feature value for class c is shown in red, which also has a very low-probability in class c'. Since $\mathbf{X}_i$ has a negative connection weight to c', decreasing x_i to $\mathbb{E}[X_i]$ for c' brings the classification closer to c' and hence the generation of a counterfactual. The initial process (i.e., Bernoulli trial) in of the hurdle model is omitted here for simplicity.

To model this, x is gathered from all training data into the latent dataset L (by passing it through C), and considering the n output classes, we divide L into $\{L_i\}_{i=1}^n$ where $\forall x \in L_i, S(x) = Y_i$. Now considering the counterfactual class data $L_{c'}$, let all data for some neuron $\mathbf{X}_i$ be $\{x_j\}_{j=1}^m \in L_{c'}$, where m is the number of instances. If we let the number of these m instances where $x_j \neq 0$ be q, the probability of success θ_i in the Bernoulli trail can be modelled as $\theta_i = q/m$, and the probability of failure as $1 - \theta_i$. The subsequent PDF f_i from Eq. (3) is modelled with $\{x_j\}_{j=1}^m \in L_{c'}, \forall x_j > 0$. Importantly, the hurdle models use what S *predicted* each instance to be (rather than the *label*), because we wish to model what the CNN has learned, irrespective of whether it is objectively correct or incorrect.

We found empirically that a hurdle model's second process PDF will typically approximate a Gaussian, Gamma, or Exponential distribution (see Fig. 5). Hence, we automated the modelling process by fitting the data with all three distributions (with and without a fixed location parameter of 0) using maximum likelihood estimation. Then, using the Kolmogorov-Smirnov test for goodness of fit across all these distributions, we chose the one of best fit. In all generated explanations, the average p-value for goodness of fit was p $>$ 0.3 across all features. With the modelling process finished, a feature value x_i is considered an exceptional feature x_e for the test image I if:

$$x_i = 0 \mid p(1 - \theta_i) < \alpha \tag{4}$$

$$x_i > 0 \mid p(\theta_i) < \alpha \tag{5}$$

Glossed, Eq. (4) dictates that it is exceptional if a neuron $\mathbf{X}_i$ does not activate, given the probability of it not activating is less than α for c' typically. Eq.(5) illustrates that it is exceptional if a neuron activates, given that the probability of it activating is less than α for c' typically. The other two exceptional feature events are:

$$\theta_i F_i(x_i) < \alpha \mid x_i > 0 \tag{6}$$

$$(1 - \theta_i) + \theta_i F_i(x_i) > 1 - \alpha \mid x_i > 0 \tag{7}$$

where F_i is the cumulative distribution function for f_i. Eq. (6) dictates that, given the neuron has activated, it is exceptional (i.e., a probability $< \alpha$) to have such a low activation value for c'. Eq. (7) relays that, given the neuron has activated, it is exceptional to have such a high activation value for c' (i.e., the example in Fig. 5). In defining the α threshold, the statistical hypothesis-testing standard was adopted, categorizing any feature value which has a probability less than $\alpha = 0.05$ as being exceptional in both experiments.

Algorithm 1: Modify exceptional features in x to produce x'

```
Input: $x$: The latent features of the test image $I$
Input: $w$: The weight vector connecting $\mathbf{X}$ to $c'$
Input: $\{x_e\}_{e=1}^n \in x$: The exceptional features
       (ordered lowest to highest probability)
foreach $x_e$ in $\{x_e\}_{e=1}^n \in x$ do
    if $w_e > 0$ and $x_e$ discovered with Eq. (4),
     Eq. (5), or Eq. (6) then
        $x_e \leftarrow \mathbb{E}[X_e]$          // Using PDF
         modelled for $c'$ in Eq. (3)
    else if $w_e < 0$ and $x_e$ discovered with Eq. (5)
     or Eq. (7) then
        $x_e \leftarrow \mathbb{E}[X_e]$          // Using PDF
         modelled for $c'$ in Eq. (3)
return $x$ (now modified to be $x'$)
```

Step 2: Changing the *Exceptional* to the *Expected*

The exceptional features $\{x_e\}_{e=1}^n \in x$ (where n is the number of exceptional features identified) divide into those that negatively or positively affect the classification of c' in I, PIECE only modifies the former (see Algorithm 1). Importantly, features are only modified if they meet the criteria regarding their connection weight, and identification process (i.e., found using Eqs. (4)-(7)). Glossed, the algorithm only modifies the exceptional feature values to their expected values if doing so brings the CNN closer to modifying the classification to c'. These exceptional features are ordered from the lowest probability to the highest, as this ordering plays a key role in semi-factual explanations where the modification of features is stopped short of the decision boundary.

Step 3: Visualizing the Explanation

Finally, having constructed x', the explanation is visualized by solving the following optimization problem with gradient descent:

$$z' = \arg\min_z \|C(G(z)) - x'\|_2^2 \tag{8}$$

and inputting z' into G to visualize the explanation I'. For the computational cost and machine specs involved, see the supplement Section S3 and S4.

Experiment 1: Counterfactuals

In this experiment, PIECE's performance is compared against other known methods for counterfactual explanation generation. The tests compare PIECE against other sufficiently general methods which are applicable to color datasets (Liu et al. 2019; Wachter, Mittelstadt, and Russell 2017) [here we use CIFAR-10], and then with the addition of other relevant works which focused on MNIST (Dhurandhar et al. 2018; Van Looveren and Klaise 2019). The methods compared in Expt. 1 are:

- **PIECE.** The present algorithm, using Eq. (8), where all exceptional features were categorized with $\alpha = 0.05$, and subsequently modified.
- **Min-Edit.** A simple minimal-edit perturbation method with a direct optimization towards c', where the optimization used gradient descent and was immediately stopped when the decision boundary was crossed, defined by: $z' = \arg\min_{z} \|S(C(G(z))) - Y_{c'}\|_2^2$.
- **Constrained Min-Edit (C-Min-Edit).** A modified version of (Liu et al. 2019),[2], and inspired by (Wachter, Mittelstadt, and Russell 2017), this optimized with gradient descent and stopped when the decision boundary was crossed, defined as: $z' = \underset{z}{\operatorname{argmin}} \max_{\lambda} \lambda \|S(C(G(z))) - Y_{c'}\|_2^2 + d(C(G(z)), x)$.
- **Contrastive Explanations Method (CEM).** Pertinent negatives from (Dhurandhar et al. 2018), which are a form of counterfactual explanation, implemented here using (Klaise et al. 2020).
- **Interpretable Counterfactual Explanations Guided by Prototypes (Proto-CF).** The method by (Van Looveren and Klaise 2019), implemented here using (Klaise et al. 2020).

Hyperparameter choices are presented in Section S2 of the supplementary material. Although other similar techniques are reported in the literature (Singla et al. 2019; Samangouei et al. 2018; Seah et al. 2019), they are not applicable as they cannot explain CNNs which are pre-trained on multi-class classification problems.

Setup, Test Set, and Evaluation Metrics

For MNIST, a test-set of 163 images classified by the CNN was used which divided into: (i) correct classifications (N=60) with six examples per number-class, (ii) close-correct classifications (N=62), that had an output SoftMax probability < 0.8, where the CNN "just" got the classification right,[3] and (iii) incorrect classifications (N=41) by the CNN (i.e., every instance misclassified by the CNN). For CIFAR-10, the test-set was divided into: (i) correct classifications (N=30) with three examples per class, and (ii) incorrect classifications (N=30) with three examples per class. All instances were randomly selected, with the exception of MNIST's incorrect classifications, which were not randomly selected as there was only 41 of them.

To evaluate an explanation's plausibility, most researchers use some measure of proximity to the data distribution as a basic requirement. For example, a person's income in a loan application system should not be a negative value in an explanation, as this is nonsensical and far from the real distribution. In MNIST, an explanation image should not have any unusual artifacts, and should, put simply, look like an actual handwritten digit. These considerations motivate the evaluation metrics used; that is, the metrics evaluate how well generated synthetic images resemble the underlying distribution. To do this, one related work proposed IM1 and IM2, which trains multiple autoencoders (AEs) to test the generated counterfactual's relative reconstruction error (Van Looveren and Klaise 2019). However, as there can be issues interpreting IM2 (Mahajan, Tan, and Sharma 2019), we replaced it with Monte Carlo Dropout (Gal and Ghahramani 2016) (MC Dropout), a commonly used method for out-of-distribution detection (Malinin and Gales 2018), with 1000 forward passes. Additionally, we use R%-Substitutability (Samangouei et al. 2018) which measures how well generated explanations can substitute for the actual training data. As there are relatively few explanations generated compared to the actual training datasets (163 compared to 60,000), we use k-NN on the pixel space of MNIST, as the classifier works well with small amounts of training data, and the centred nature of the MNIST dataset means it performs well normally (i.e., $\sim$ 97% accuracy). In the current experiment, the measures used were:

- **MC-Mean.** Posterior mean of MC Dropout on the generated counterfactual image (higher is better).
- **MC-STD.** Posterior standard deviation of MC Dropout on the generated counterfactual (lower is better).
- **NN-Dist.** The distance of the counterfactual's latent representation at layer $\mathbf{X}$ from the nearest training instance measured with the L_2 norm [i.e., the closest "possible world" (Wachter, Mittelstadt, and Russell 2017)].
- **IM1.** From (Van Looveren and Klaise 2019), an AE is trained on class c (i.e., AE_c) and c' (i.e., $AE_{c'}$) to compute IM1 $= \frac{\|I' - AE_{c'}(I')\|\|_2^2}{\|I' - AE_c(I')\|_2^2}$, where a lower score is considered better.
- **Optim. Time.** Time taken to optimize each image.
- **Substitutability (R%-Sub).** Inspired by (Samangouei et al. 2018), the method's generated counterfactuals are fit to a k-NN classifier (in pixel space) which predicts the MNIST test set. The original training set gives $\sim$ 97% accuracy with k-NN, if a method produces half that accuracy, its R%-Sub score is 50% (a higher score is considered better, as it can replace the training data).

[2] They used the pixel rather than latent-space in $d(.)$. We tested both but found no significant difference. However, the latent-space required a smaller λ to find z', and was more stable (Russell 2019).

[3] SoftMax probability is not considered reliable for CNN certainty, but it gives a baseline (Hendrycks and Gimpel 2016).

2*Method	MC Mean		MC STD		NN-Dist		IM1		Optim. Time		R%-Sub
	# 1	# 2	# 1	# 2	# 1	# 2	# 1	# 2	#1	#2	#1
Min-Edit	0.52	0.61	0.24	0.13	1.02	1.48	0.91	1.17	**1.84**	**00.75**	42.87
C-Min-Edit	0.50	0.45	0.25	0.14	1.03	1.50	0.93	1.21	3.41	48.06	40.33
Proto-CF	0.53	N/A	0.23	N/A	1.02	N/A	1.28	N/A	84.89	N/A	34.75
CEM	0.62	N/A	0.22	N/A	0.99	N/A	1.13	N/A	91.37	N/A	43.87
PIECE	**0.99**	**0.96**	**0.02**	**0.02**	**0.41**	**1.17**	**0.72**	**1.15**	26.36	85.51	**69.32**

Table 1: The average performance over the test-sets of the five counterfactual explanation methods for dataset #1 (MNIST) and dataset #2 (CIFAR-10) in Expt. 1, where the best results are highlighted in bold. R%-Sub is tested on MNIST only.

Results and Discussion

PIECE generates counterfactual explanations with better results on all plausiblity metrics (see Table 1); a statistical analysis using the Anderson-Darling test (AD) showed these values to be reliably different on all metrics, $AD > 22$, $p < .001$ (except for IM1 on CIFAR-10). PIECE generates much more visibly plausible counterfactual explanations than Min-Edit (e.g., see Fig. 6 and Fig. 7). Notably, Proto-CF/CEM were the only methods that failed to find a counterfactual explanation for 20/25 images out of a total of 163 on MNIST, respectively. Interestingly, for all results on MNIST, a plot of the NN-Dist measure against the MC-Mean/MC-STD scores show a significant linear relationship r = -0.8/0.82. So, the more a generated counterfactual is grounded in the training data, the more likely it is to be plausible [as (Laugel et al. 2019) argued; see Section S5 of the supplementary material for these plots]. In addition, though PIECE is not the fastest method, its optimization time can be significantly reduced without loss of plausibility by either reducing the number of epochs or utilizing a GPU.

Lastly, whilst allowing Min-Edit (or C-Min-Edit) to optimize until the SoftMax probability approaches 1.0 may appear to be a simple remedy to improve plausibility (as it optimizes beyond the decision boundary), we found it is not reliable. Specifically, we illustrate the potential pitfalls of doing so in Fig. 6, were it fails to generate a plausible explanation. This is largely due to its reliance on a "blind perturbation" approach that optimizes towards a specific class with no constraints. In contrast, PIECE modifies exceptional features to expected statistical values, leaving all others intact, helping to avoid such implausible outputs.

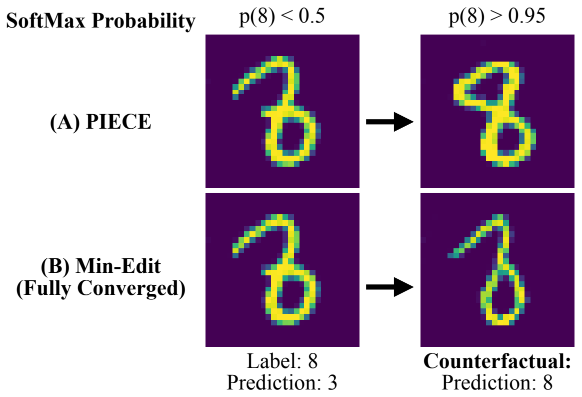

Figure 6: PIECE Versus Min-Edit: (A) By changing exceptional features to expected statistical values PIECE keeps within the data distribution to make a plausible explanation. (B) By contrast, the *Min-Edit* method, when fully converged (i.e., which takes it far over the decision boundary), goes out-of-distribution to make a less plausible explanation despite the CNN having a high confidence (i.e., p(8) > 0.95).

Experiment 2: Semi-Factuals

(Nugent, Doyle, and Cunningham 2009) argued that semi-factual explanations (they called it *a-fortiori* reasoning) should involve the largest possible feature modifications without changing classification (e.g., "*Even if* you trebled your salary, you would still not get the loan"). However, they did not consider semi-factuals for image datasets, or perform controlled experiments. As such, a new evaluation method is needed to measure "good semi-factuals" in terms of how far the generated semi-factual instance is from the test instance, without crossing the decision boundary into c'. To accomplish this in an image domain, we use the L_1 distance between the test image and synthetic explanatory semi-factual in the pixel-space (n.b., the greater the distance the better the method). In the present experiment, PIECE is only compared to the minimal-edit methods from Expt. 1

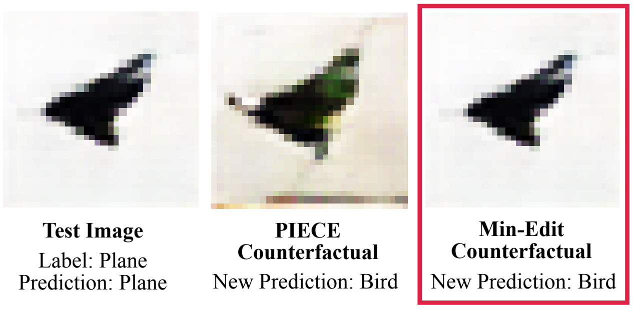

Figure 7: PIECE's Counterfactual Explanations on CIFAR-10: PIECE explains the prediction by contrasting the query with a counterfactual image. Glossed, the explanation reads *I think the query is a plane because in order to be something else (e.g., a bird) it would have to look like this*. The less plausible Min-Edit explanation is shown for comparison.

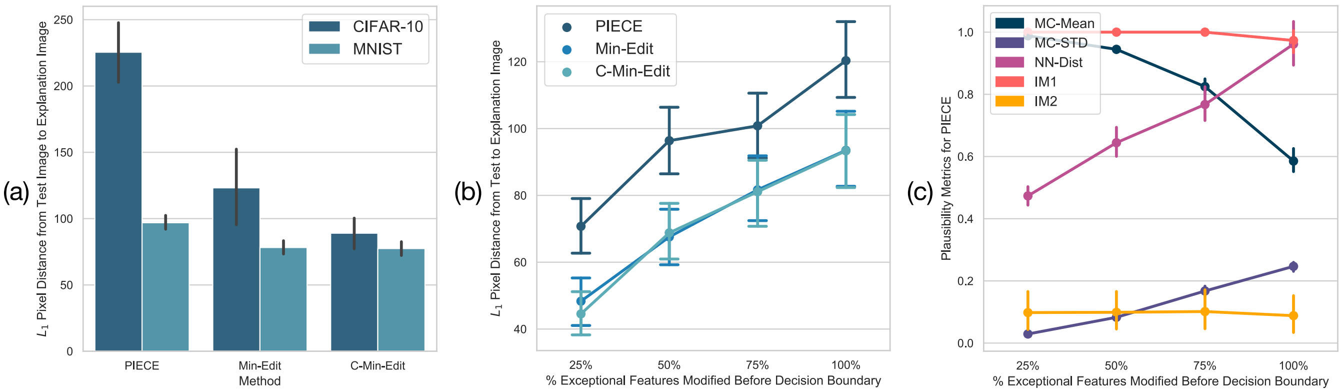

Figure 8: Expt. 2 Results: (a) the L_1 pixel-space change between the test- and semi-factual images for the three methods on two datasets, (b) the same L_1 metric for the three methods under progressive proportions of feature-changes on MNIST, (c) the plausibility measures for PIECE, under the same progressive proportions of feature-changes on MNIST.

(i.e., Min-Edit and C-Min-Edit), as the other methods (i.e., CEM and Proto-CF) cannot generate semi-factuals. To thoroughly evaluate all methods, three distinct tests were carried out (see Fig. 8). First, a max-edit run was performed on a set of test images, where each of the three methods produced their "best semi-factual". Specifically, Min-Edit and C-Min-Edit were allowed optimize until the next step would push them over the decision boundary into c', and PIECE followed its normal protocol, but stopped Algorithm 1 when the next exceptional feature modification to x would alter the CNN classification such that $S(x) \neq Y_c$. Second, the performance of the methods, on the same test set, for different proportions of feature changes were recorded. Specifically, PIECE modifies 25%, 50%, 75%, and 100% of the exceptional-features from the first test, whilst the min-edit methods were allowed to optimize to the same distance as PIECE (measured using L_2 distance) in the latent-space for each of these four distances. This second test allows us to view the full spectrum of results. Third, plausibility measures were applied to PIECE's explanations in the second test for a full profile of its operation.

Setup, Test-Set, and Evaluation Metrics

PIECE was run as in Expt. 1, with the counterfactual class c' being selected in the same way, and with all exceptional features being identified using $\alpha = 0.05$. For full details on hyperparameter choices see Section S2 of the supplementary material. A test set of 90 test images were used (i.e., the "correct" set from MNIST and the "correct" set from CIFAR-10 in Expt. 1), with the plausibility of PIECE being evaluated using the same metrics from Expt. 1 (but we add IM2 since it has not been tested on semi-factuals). The semi-factual's goodness was measured using the L_1 pixel distance between the test image and the semi-factual image generated, the larger this distance, the better the semi-factual.

Results and Discussion

Fig. 8 shows the results of the first comparative tests of semi-factual explanations in XAI. First, PIECE produces the best semi-factuals on both datasets, with significantly higher L_1 distance scores than the min-edit methods (see Fig. 8a; *AD* > 2.5, $p < .029$ for MNIST, *AD* >11.75, $p < .001$ for CIFAR-10). Second, upon a closer examination of the MNIST results, all methods produce better semi-factuals at every distance measured (see Fig. 8b), but PIECE's semi-factuals are significantly better at every distance tested (*AD* > 3.3, $p < .015$). Third, when different plausibility measures are applied to progressive incremental changes of the exceptional features by PIECE on MNIST, there are significant changes across some (i.e., MC-Mean, MC-STD, and NN-Dist), but not all measures (i.e., IM1/IM2), perhaps suggesting the former metrics are more sensitive than the latter (see Fig. 8c). Notably, like Expt. 1, there is a clear trade-off between plausibility (measured in MC-Dropout measures), and NN-Dist. Additionally, as semi-factuals get better (with larger changes in the pixel space), they may sacrifice some plausibility. Note that CIFAR-10 is omitted from Fig. 8b/c to aid presentation, but similar results were found on both datasets.

Conclusion

A novel method, PlausIble Exceptionality-based Contrastive Explanations (PIECE), has been proposed that produces plausible counterfactuals to provide *post-hoc* explanations for a CNN's classifications. Competitive tests have shown that PIECE adds significantly to the collection of tools currently proposed to solve this XAI problem. Future work will extend this effort to more complex image datasets. In addition, another obvious direction would be to use recent advances in text and tabular generative models to extend the framework into these domains, alongside pursuing semi-factual explanations more extensively, as there remains a rich, substantial, untapped research area involving them.

Acknowledgements

This paper emanated from research funded by Science Foundation Ireland (SFI) to the Insight Centre for Data Analytics (12/RC/2289 P2), SFI and the Department of Agriculture, Food and the Marine on behalf of the Government of Ireland to the VistaMilk SFI Research Centre (16/RC/3835).

Ethics Statement

A major aim of explainable AI research is to create techniques and task scenarios that support people in making fairness, accountability, and trust judgements about AI systems. The present work is part of this research effort. By providing people with counterfactual/semi-factual explanations, there is a risk of revealing "too much" about how a system operates (e.g., they potentially convey exactly how a proprietary algorithm works). Notably, the balance of this risk is more on the side of the algorithm-proprietors than on algorithm-users, which may be where we want it to be in the interests of fairness and accountability. Indeed, these methods have the potential to reveal biases in datasets and algorithms as they reveal how data is being used to make predictions. The psychological evidence shows that counterfactual and semi-factual explanations elicit spontaneous causal thinking in people; hence, they may have the benefit of reducing the passive use of AI technologies, enabling better human-in-the-loop systems, where people have appropriate (rather than inappropriate) trust. However, semi-factual explanations may have the potential to convince people than an algorithm is more correct in its decisions than it actually is; if this is true, semi-factuals could engender inappropriate trust on the part of end-users in a poorly-performing system. At present, this is only an hypothesis. However, it is a valid concern and one which warrants further study both computationally and psychologically.

References

Adadi, A.; and Berrada, M. 2018. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6: 52138–52160.

Barker, S. 1991. Even, still and counterfactuals. *Linguistics and Philosophy* 14(1): 1–38.

Bennett, J. 1982. Even if. *Linguistics and Philosophy* 5(3): 403–418.

Boninger, D. S.; Gleicher, F.; and Strathman, A. 1994. Counterfactual thinking: From what might have been to what may be. *Journal of personality and social psychology* 67(2): 297.

Byrne, R. M. 2016. Counterfactual thought. *Annual review of psychology* 67: 135–157.

Byrne, R. M. 2019. Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 6276–6282.

Dhurandhar, A.; Chen, P.-Y.; Luss, R.; Tu, C.-C.; Ting, P.; Shanmugam, K.; and Das, P. 2018. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in neural information processing systems*, 592–603.

Dodge, J.; Liao, Q. V.; Zhang, Y.; Bellamy, R. K.; and Dugan, C. 2019. Explaining models: an empirical study of how explanations impact fairness judgment. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 275–285.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* .

Goodman, N. 1983. *Fact, fiction, and forecast*. Harvard University Press.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51(5): 1–42.

Gunning, D. 2017. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web* 2.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136* .

Icard, T. F.; Kominsky, J. F.; and Knobe, J. 2017. Normality and actual causal strength. *Cognition* 161: 80–93.

Keane, M. T.; and Kenny, E. M. 2019. How case-based reasoning explains neural networks: A theoretical analysis of XAI using post-hoc explanation-by-example from a survey of ANN-CBR twin-systems. In *International Conference on Case-Based Reasoning*, 155–171. Springer.

Keane, M. T.; and Smyth, B. 2020. Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI). In *International Conference on Case-Based Reasoning*. Springer.

Kenny, E. M.; Ford, C.; Quinn, M.; and Keane, M. T. 2021. Explaining Black-Box classifiers using Post-Hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies. *Artificial Intelligence* 103459.

Kenny, E. M.; and Keane, M. T. 2019. Twin-systems to explain artificial neural networks using case-based reasoning: comparative tests of feature-weighting methods in ANN-CBR twins for XAI. In *Twenty-Eighth International Joint Conferences on Artificial Intelligence (IJCAI), Macao, 10-16 August 2019*, 2708–2715.

Kenny, E. M.; Ruelle, E.; Geoghegan, A.; Shalloo, L.; O'Leary, M.; O'Donovan, M.; and Keane, M. T. 2019. Predicting Grass Growth for Sustainable Dairy Farming: A CBR System Using Bayesian Case-Exclusion and Post-Hoc, Personalized Explanation-by-Example (XAI). In *International Conference on Case-Based Reasoning*, 172–187. Springer.

Kenny, E. M.; Ruelle, E.; Geoghegan, A.; Shalloo, L.; O'Leary, M.; O'Donovan, M.; Temraz, M.; and Keane, M. T. 2020. Bayesian Case-Exclusion and Personalized Explanations for Sustainable Dairy Farming. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

Klaise, J.; Van Looveren, A.; Vacanti, G.; and Coca, A. 2020. Alibi: Algorithms for monitoring and explaining machine learning models. URL https://github.com/SeldonIO/alibi.

Laugel, T.; Lesot, M.-J.; Marsala, C.; Renard, X.; and Detyniecki, M. 2019. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *arXiv preprint arXiv:1907.09294* .

LeCun, Y.; Cortes, C.; and Burges, C. 2010. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2.

Lipton, Z. C. 2018. The mythos of model interpretability. *Queue* 16(3): 31–57.

Liu, S.; Kailkhura, B.; Loveland, D.; and Yong, H. 2019. Generative Counterfactual Introspection forExplainable Deep Learning. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Lucic, A.; Haned, H.; and de Rijke, M. 2020. Why does my model fail? contrastive local explanations for retail forecasting. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 90–98.

Macbeth, G.; and Razumiejczyk, E. 2019. Implicit facilitation effect on counterfactual and semifactual thinking. *Education Sciences & Psychology* 54(4).

Mahajan, D.; Tan, C.; and Sharma, A. 2019. Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers. *arXiv preprint arXiv:1912.03277* .

Malinin, A.; and Gales, M. 2018. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, 7047–7058.

McCloy, R.; and Byrne, R. M. 2002. Semifactual "even if" thinking. *Thinking & Reasoning* 8(1): 41–67.

McGrath, R.; Costabello, L.; Van, C. L.; Sweeney, P.; Kamiab, F.; Shen, Z.; and Lecue, F. 2018. Interpretable credit application predictions with counterfactual explanations. *arXiv preprint arXiv:1811.05245* .

Miller, T. 2018. Contrastive explanation: A structural-model approach. *arXiv preprint arXiv:1811.03163* .

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267: 1–38.

Nugent, C.; Doyle, D.; and Cunningham, P. 2009. Gaining insight through case-based explanation. *Journal of Intelligent Information Systems* 32(3): 267–295.

Pawelczyk, M.; Broelemann, K.; and Kasneci, G. 2020. Learning Model-Agnostic Counterfactual Explanations for Tabular Data. In *Proceedings of The Web Conference 2020*, 3126–3132.

Pearl, J. 2000. Causality: Models, reasoning and inference cambridge university press. *Cambridge, MA, USA*, 9: 10–11.

Pichai, S. 2018. AI at Google: our principles. https://www.blog.google/technology/ai/ai-principles/. [Online; accessed 01-June-2020].

Poyiadzi, R.; Sokol, K.; Santos-Rodriguez, R.; De Bie, T.; and Flach, P. 2020. FACE: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 344–350.

Russell, C. 2019. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 20–28.

Samangouei, P.; Saeedi, A.; Nakagawa, L.; and Silberman, N. 2018. ExplainGAN: Model Explanation via Decision Boundary Crossing Transformations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 666–681.

Sanna, L. J. 1996. Defensive pessimism, optimism, and stimulating alternatives: Some ups and downs of prefactual and counterfactual thinking. *Journal of personality and social psychology* 71(5): 1020.

Santamaría, C.; Espino, O.; and Byrne, R. M. 2005. Counterfactual and semifactual conditionals prime alternative possibilities. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 31(5): 1149.

Seah, J. C.; Tang, J. S.; Kitchen, A.; Gaillard, F.; and Dixon, A. F. 2019. Chest radiographs in congestive heart failure: visualizing neural network learning. *Radiology* 290(2): 514–522.

Singla, S.; Pollack, B.; Chen, J.; and Batmanghelich, K. 2019. Explanation by Progressive Exaggeration. In *International Conference on Learning Representations*.

Sutton, M. A.; Oenema, O.; Erisman, J. W.; Leip, A.; van Grinsven, H.; and Winiwarter, W. 2011. Too much of a good thing. *Nature* 472(7342): 159–161.

Van Looveren, A.; and Klaise, J. 2019. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584* .

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31: 841.

Yang, L.; Kenny, E.; Ng, T. L. J.; Yang, Y.; Smyth, B.; and Dong, R. 2020. Generating Plausible Counterfactual Explanations for Deep Transformers in Financial Text Classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6150–6160.

Zhu, J.; Shen, Y.; Zhao, D.; and Zhou, B. 2020. In-domain gan inversion for real image editing. *arXiv preprint arXiv:2004.00049* .

Zurek, T. 2012. Modelling of a'fortiori reasoning. *Expert Systems with Applications* 39(12): 10772–10779.