The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)

CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines

Arjun R. Akula,[1] **Shuai Wang,**[2] **Song-Chun Zhu**[1]
[1]UCLA Center for Vision, Cognition, Learning, and Autonomy
[2]University of Illinois at Chicago
aakula@ucla.edu, shuaiwanghk@gmail.com, sczhu@stat.ucla.edu

Abstract

We present CoCoX (short for Conceptual and Counterfactual Explanations), a model for explaining decisions made by a deep convolutional neural network (CNN). In Cognitive Psychology, the factors (or semantic-level features) that humans zoom in on when they imagine an alternative to a model prediction are often referred to as *fault-lines*. Motivated by this, our CoCoX model explains decisions made by a CNN using fault-lines. Specifically, given an input image I for which a CNN classification model M predicts class c_{pred}, our fault-line based explanation identifies the minimal semantic-level features (e.g., *stripes* on zebra, *pointed ears* of dog), referred to as explainable concepts, that need to be added to or deleted from I in order to alter the classification category of I by M to another specified class c_{alt}. We argue that, due to the conceptual and counterfactual nature of fault-lines, our CoCoX explanations are practical and more natural for both expert and non-expert users to understand the internal workings of complex deep learning models. Extensive quantitative and qualitative experiments verify our hypotheses, showing that CoCoX significantly outperforms the state-of-the-art explainable AI models. Our implementation is available at https://github.com/arjunakula/CoCoX

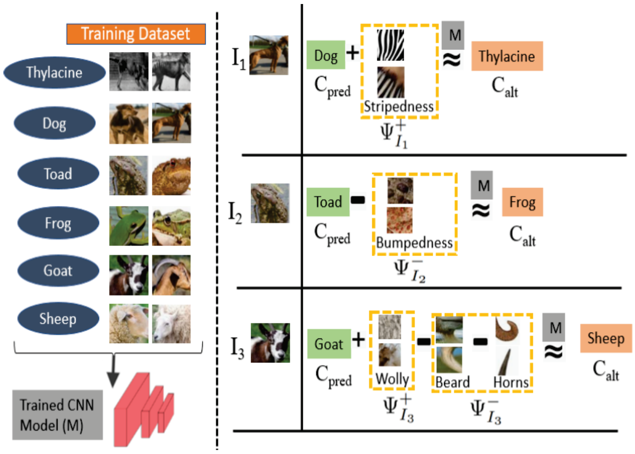


Figure 1: CoCoX Explanations using Fault-Lines: Positive fault-line explanation ($\Psi^+_{I_1}$) suggests adding *stripes* to the animal in the input image (I_1) to alter the model M's prediction from `Dog` class to `Thylacine` class, i.e., the concept of *stripedness* is critical for M to decide between `Dog` and `Thylacine` in I_1. Similarly, negative fault-line $\Psi^-_{I_2}$ suggests removing *bumps* from I_2 to alter the classification category from `Toad` to `Frog`. Changing the classification result of I_3 from `Goat` to `Sheep` requires adding *wool* and removing *beard* and *horns* from I_3, i.e., it needs both positive and negative fault-lines.

Introduction

Artificial Intelligence (AI) systems are becoming increasingly ubiquitous from low-risk environments such as movie recommendation systems and chatbots to high-risk environments such as self-driving cars, drones, military applications, and medical-diagnosis and treatment. However, understanding the behavior of these systems remains a significant challenge as they cannot explain why they reach a specific recommendation or a decision. This is especially problematic in high risk environments such as banking, healthcare, and insurance, where AI decisions can have significant consequences. Therefore, we need explainable AI (XAI) models as tools to understand the decisions made by these AI systems (Miller 2018; Sundararajan, Taly, and Yan 2017; Ramprasaath et al. 2016; Zeiler and Fergus 2014; Smilkov et al. 2017).

XAI models, through explanations, aim at making the underlying inference mechanism of AI systems transparent and interpretable to expert users (system developers) and non-expert users (end-users) (Lipton 2016; Ribeiro, Singh, and Guestrin 2016; Hoffman 2017). In this work, we focus mainly on increasing justified human trust (JT) in AI systems, through explanations (Hoffman et al. 2018; Akula et al. 2019b; 2019a). Justified trust is computed based on human judgments of AI system's prediction (more details on this are described in the Experiments section). Despite an increasing amount of work on XAI (Smilkov et al. 2017; Sundararajan, Taly, and Yan 2017; Zeiler and Fergus 2014; Kim, Rudin, and Shah 2014; Zhang, Nian Wu, and Zhu 2018; R Akula et al. 2019), providing explanations that can increase justified human trust remains an important research problem (Jain and Wallace 2019). To address this problem, we present a new XAI model CoCoX which explains decisions made by a deep convolutional neural network (CNN) using *fault-lines* (Kahneman and Tversky 1981).

Fault-lines are the high-level semantic aspects of reality that humans zoom in on when they imagine an alternative to it. More concretely, given an input image I for which a CNN model M predicts class c_{pred}, our fault-line based explanation identifies a *minimal* set of semantic features, referred to as *explainable concepts* (xconcepts), that need to be added to or deleted from I in order to alter the classification category of I by M to another specified class c_{alt}. For example, let us consider a training dataset for an image classification task shown in Figure 1 containing the classes `Dog`, `Thylacine`, `Frog`, `Toad`, `Goat` and `Sheep`, and a CNN based classification model M which is trained on this dataset. In order to alter the model's prediction of input image I_1 from `Dog` to `Thylacine`, the fault-line ($\Psi^{+}_{I_1,c_{pred},c_{alt}}$) suggests adding *stripes* to the `Dog`. We call this a positive fault-line (PFT) as it involves adding a new xconcept, i.e., *stripedness*, to the input image. Similarly, to change the model prediction of I_2 from `Toad` to `Frog`, the fault-line ($\Psi^{-}_{I_2,c_{pred},c_{alt}}$) suggests removing *bumps* from the `Toad`. We call this a negative fault-line (NFT) as it involves subtracting xconcept, i.e., *bumpedness*, from the input image. In most cases, both PFT and NFT are needed to successfully alter the model prediction.

For example, in Figure 1, in order to change the model prediction of I_3 from `Goat` to `Sheep`, we need to add an xconcept *wool* (PFT) to I_3 and also remove xconcepts *beard* and *horns* (NFT) from I_3. As we can see, these fault-lines can be directly used to make the internal decision making criteria of deep neural network transparent to both expert and non-expert users. For instance, we answer the question *"Why does the machine classify the image I_3 as `Goat` instead of `Sheep`?"* by using PFT $\Psi^{+}_{I_3,c_{pred},c_{alt}}$ and NFT $\Psi^{-}_{I_3,c_{pred},c_{alt}}$ as follows: "Machine thinks the input image is `Goat` and not `Sheep` mainly because `Sheep`'s feature *woolly* is absent in I_3 and `Goat`'s features `beard` and `horns` are present in I_3". It may be noted that there could be several other features of `Sheep` and `Goat` that might have influenced the model's prediction. However, fault-lines only capture the most critical (minimal) features that highly influenced the model's prediction.

What makes fault-lines a good visual explanation? We chose fault-lines as an explanation for the following two important reasons:

1. Firstly, unlike current methods in XAI which mainly focus on pixel-level explanations (viz. saliency maps), fault-line based explanations are **concept-level** explanations. Pixel-level explanations are not effective at human scale, whereas concept level explanations are effective, less ambiguous, and more natural for both expert and non-expert users in building a mental model of a vision system (Kim et al. 2018). Moreover, with conceptual explanations, humans can easily generalize their understanding to new unseen instances/tasks. In our work, as shown in Figure 1, we represent xconcepts (e.g., *stripedness*) using a set of example images (similar to (Kim et al. 2018)).
2. Secondly, fault-lines are **counter-factual** in nature, i.e., they provide a *minimal* amount of information capable of altering a decision. This makes them easily digestible and practically useful for understanding the reasons for a model's decision (Wachter, Mittelstadt, and Russell 2017). For example, consider the fault-line explanation for image I_3 in Figure 1. The explanation provides only the most critical changes (i.e., adding wool and removing beard and horns) required to alter the model's prediction from `Goat` to `Sheep`, though several other changes may be necessary.

While there are recent works on generating pixel-level counter-factual and contrastive explanations (Hendricks et al. 2018; Dhurandhar et al. 2018; Goyal et al. 2019), to the best of our knowledge, this is the first work to propose a method for generating explanations that are counter-factual as well as conceptual.

We identify two main challenges in generating a fault-line explanation, namely: (a) How to identify the set of xconcepts; and (b) How to select the most critical xconcepts that alter the model prediction from c_{pred} to c_{alt}. In this work, we first propose a novel method to mine all the plausible xconcepts from the given dataset automatically. We then identify class-specific xconcepts by using directional derivatives (Kim et al. 2018). Finally, we pose the derivation of a fault-line as an optimization problem which selects a minimal set of these xconcepts to alter the model's prediction. We perform extensive human study experiments to demonstrate the effectiveness of our approach in improving human understanding of the underlying classification model.

Through our human studies, we show that our fault-line based explanations significantly outperform the baselines (i.e., attribution techniques and pixel-level counterfactual explanations) in terms of qualitative and quantitative metrics such as Justified Trust and Explanation Satisfaction (Hoffman et al. 2018).

Concurrent to our work, recent work by (Ghorbani, Wexler, and Kim 2019) also seeks to automatically identify human-friendly xconcepts. However, they use segmentation methods to identify xconcepts, whereas we use Grad-CAM (Selvaraju et al. 2017) based localization maps. Moreover, their explanations are not counter-factual unlike our fault-line based explanations.

The contributions of this work are threefold: (i) we introduce a new XAI framework based on fault-lines to generate conceptual and counterfactual explanations; (ii) we present a new method to mine xconcepts from a given training dataset automatically and derive the fault-lines; (iii) we show that our fault-line explanations qualitatively and quantitatively outperform baselines in improving human understanding of the classification model.

Approach

In this section, we detail our ideas and methods for generating fault-line explanations. Without loss of generality, we consider a pre-trained CNN (M) for image classification. Given an input image I, the CNN predicts a log-probability output $\log P(\mathbf{Y}|I)$ over the output classes $\mathbf{Y}$. Let χ denote a dataset of training images, where $\chi_c \subset \chi$ represents the subset that

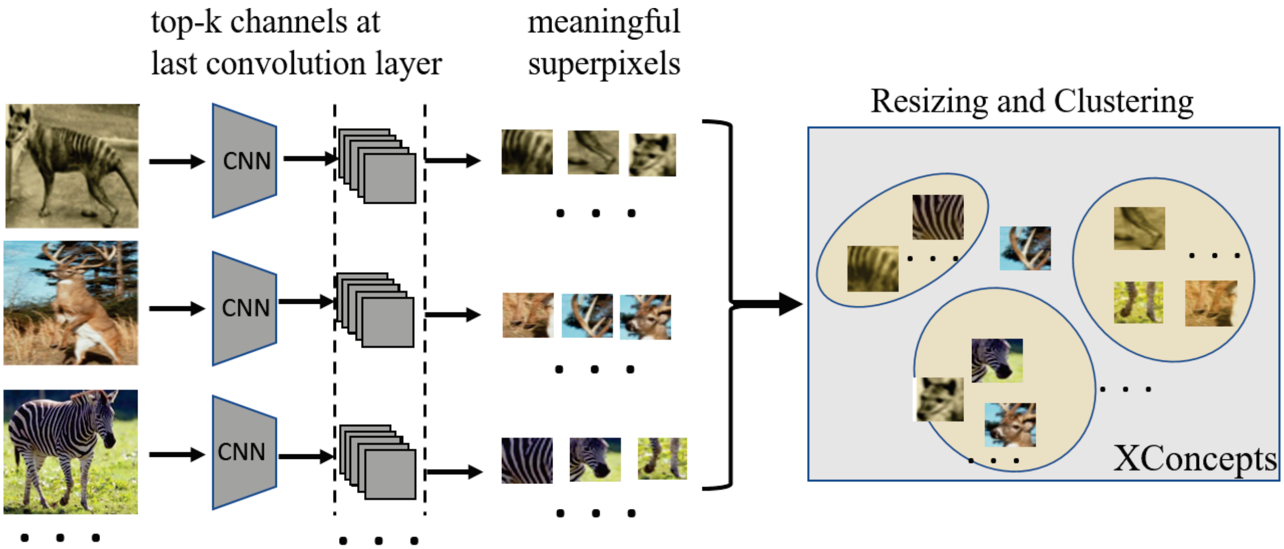


Figure 2: We consider feature maps from the last convolutional layer as instances of xconcepts and obtain their localization maps (i.e., superpixels) by computing the gradients of the output with respect to the feature maps. We select highly influential superpixels and then apply K-means clustering with outlier removal to group these superpixels into clusters where each cluster represents an xconcept.

belongs to category $c \in \mathbf{Y}$, $(c = 1, 2, \ldots, C)$. We denote the score (logit) for class c (before the softmax) as y^c and the predicted class label as c_{pred}. Our high-level goal is to find a fault-line explanation (Ψ) that alters the CNN prediction from c_{pred} to another specified class c_{alt} using a minimal number of xconcepts. We follow (Kim et al. 2018) in defining the notion of xconcepts where each xconcept is represented using a set of example images. This representation of xconcepts provides great flexibility and portability as it will not be constrained to input features or a training dataset, and one can utilize the generated xconcepts across multiple datasets and tasks.

We represent the quadruple $<I, c_{pred}, c_{alt}>$ as a human's query Q that will be answered by showing a fault-line explanation Ψ. We use Σ to represent all the xconcepts mined from χ. The xconcepts specific to the class c_{pred} and c_{alt} are represented as Σ_{pred} and Σ_{alt} respectively. Our strategy will be to first identify the xconcepts Σ_{pred} and Σ_{alt} and then generate a fault-line explanation by finding a minimal set of xconcepts from Σ_{pred} and Σ_{alt}. Formally, the objective is to find a fault-line that maximizes the posterior probability:

$$\arg\max_{\Psi} P\left(\Psi, \Sigma_{pred}, \Sigma_{alt}, \Sigma \,\middle|\, Q\right) \tag{1}$$

Mining Xconcepts

We first compute $P\left(\Sigma \,|\, \chi, M\right)$ by identifying a set of semantically meaningful superpixels from every image and then perform clustering such that all the superpixels in a cluster are semantically similar. Each of these clusters represent an xconcept. We then identify class specific xconcepts i.e., $P\left(\Sigma_{pred} \,\middle|\, \Sigma, \chi, I, c_{pred}, M\right)$ and $P\left(\Sigma_{alt} \,|\, \Sigma, \chi, I, c_{alt}, M\right)$.

A. Finding Semantically Meaningful Super-pixels as Xconcepts Figure 2 shows the overall algorithm for computing $P\left(\Sigma \,|\, \chi, M\right)$. As deeper layers of the CNN capture richer semantic aspects of the image, we construct the xconcepts by making use of feature maps from the last convolution layer. Let f denote the feature extractor component of the CNN and g denote the classifier component of the CNN that takes the output of f and predicts log-probabilities over output classes $\mathbf{Y}$. We denote the m feature maps produced at layer L of the CNN as $A^{m,L} = \{a^L | a^L = f(I)\}$ which are of width u and height v. We consider each feature map as an instance of an xconcept and obtain its localization map (i.e., super-pixels of each feature map). To produce the localization map, we use Grad-CAM (Selvaraju et al. 2017) to compute the gradients of y^c with respect to the feature maps $A^{m,L}$ and are then spatially pooled using Global Average Pooling (GAP) to obtain the importance weights ($\alpha^c_{m,L}$) of a feature map m at layer L for a target class c:

$$\alpha^c_{m,L} = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^{m,L}_{ij}} \tag{2}$$

Using the importance weights, we select top p super-pixels for each class. Given that there are C output classes in the dataset χ, we get $p * C$ super-pixels from each image in the training dataset. We apply K-means clustering with outlier removal to group these super-pixels into G clusters where each cluster represents an xconcept (as shown in Figure 2). For clustering, we consider the spatial feature maps $f(I)$ instead of the super-pixels (i.e., actual image regions) themselves. We use the silhouette score value of a different range of clusters to determine the value of K.

B. Identifying Class-Specific Xconcepts For each output class c, we learn the most common xconcepts that are highly influential in the prediction of that class over the entire training dataset χ. We use the TCAV technique (Kim et al. 2018) to identify these class-specific xconcepts. Specifically, we construct a vector representation of each xconcept, called a CAV (denoted as v_X), by using a direction normal to a linear classifier trained to distinguish between the xconcept activations from the random activations. We then compute directional derivatives ($S_{c,X}$) to produce estimates of how important the concept X was for a CNN's prediction of a target class c, e.g., how important the xconcept `stripedness` is for predicting the zebra class.

$$S_{c,X} = \nabla g_c(f(I)) \cdot v_X \tag{3}$$

where g_c denote the classifier component of the CNN that takes the output of f and predicts log-probability of output class c. We argue that these class-specific xconcepts facilitate in generating meaningful explanations by pruning out incoherent xconcepts. For example, the xconcepts such as `wheel` and `wings` are irrelevant in explaining why the network's prediction is a $zebra$ and not a cat.

Fault-Line Identification

In this subsection, we describe our approach to generate a fault-line explanation using the class-specific xconcepts. Let us consider that n_{pred} and n_{alt} xconcepts have been identified for output classes c_{pred} and c_{alt} respectively, i.e., $\left|\Sigma_{pred}\right| = n_{pred}$ and $|\Sigma_{alt}| = n_{alt}$. We denote CAVs of the n_{pred} xconcepts belonging to the class c_{pred} as $v_{pred} = \{v^i_{pred}, i = 1, 2, \ldots, n_{pred}\}$ and CAVs of the n_{alt} xconcepts belonging to the class c_{alt} as $v_{alt} = \{v^i_{alt}, i = 1, 2, \ldots, n_{alt}\}$. We formulate finding a fault-line explanation as the following optimization problem:

$$\begin{aligned}
&\underset{\delta_{pred},\delta_{alt}}{\text{minimize}} \quad \alpha D(\delta_{pred}, \delta_{alt}) + \beta \left\|\delta_{pred}\right\|_1 + \lambda \left\|\delta_{alt}\right\|_1 ;\\
&D(\delta_{pred}, \delta_{alt}) = max\{g^{pred}(I^{'}) - g^{alt}(I^{'}), -\tau\};\\
&I^{'} = A^{m,L} \circ v^{\top}_{pred}\delta_{pred} \circ v^{\top}_{alt}\delta_{alt};\\
&\delta^i_{pred} \in \{-1, 0\},\ \delta^i_{alt} \in \{0, 1\}\ \forall i \text{ and } \alpha, \beta, \lambda, \tau \geq 0.
\end{aligned} \tag{4}$$

We elaborate on the role of each term in the Equation 4 as follows. Our goal here is to derive a fault-line explanation that gives us the minimal set of xconcepts from Σ_{pred} and Σ_{alt} that will alter the model prediction from c_{pred} to c_{alt}. Intuitively, we try creating new images (I') by removing xconcepts in Σ_{pred} from I and adding xconcepts in Σ_{alt} to I until the classification result changes from c_{pred} to c_{alt}. To do this, we do not directly perturb the original image but change the activations obtained at last convolutional layer $A^{m,L}$ instead. In order to perturb the activations, we take the Hadamard product ($\circ$) between the activations ($A^{m,L}$), $v^{\top}_{pred}\delta_{pred}$ and $v^{\top}_{alt}\delta_{alt}$. The difference between the new logit scores for c_{pred} (i.e., $g^{pred}(I')$) and c_{alt} (i.e,. $g^{alt}(I')$) is controlled by the parameter τ. We apply a projected fast iterative shrinkage-thresholding algorithm (FISTA) (Beck and Teboulle 2009; Dhurandhar et al. 2018) for solving the above optimization problem. We outline our method in Algorithm 1.

Algorithm 1: Generating Fault-Line Explanations

```
Input: input image $I$, classification model $M$, predicted class label $c_{pred}$, alternate class label $c_{alt}$ and training dataset $\chi$
  1. Find semantically meaningful superpixels in $\chi$,
       $\alpha^c_{m,L} = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A^{m,L}_{ij}}$
  2. Apply K-means clustering on superpixels and obtain xconcepts ($\Sigma$).
  3. Identify class specific xconcepts ($\Sigma_{pred}$ and $\Sigma_{alt}$) using TCAV,
       $S_{c,X} = \nabla g_c(f(I)) \cdot v_X$
  4. Solve Equation 4 to obtain fault-line $\Psi$,
       $\Psi \leftarrow \min_{\delta_{pred},\delta_{alt}} \alpha D(\delta_{pred}, \delta_{alt}) + \beta \left\|\delta_{pred}\right\|_1 + \lambda \left\|\delta_{alt}\right\|_1$
return $\Psi$.
```

Experiments

We conducted extensive human subject experiments to quantitatively and qualitatively assess the effectiveness of the proposed fault-line explanations in helping expert human users and non-expert human users understand the internal workings of the underlying model. We chose an image classification task for our experiments (although the proposed approach is generic and can be applied to any task). We use the following metrics (Hoffman et al. 2018; Hoffman 2017) to compare our method with the baselines[1].

1. ***Justified Trust*** (Quantitative Metric). Justified Trust is computed by evaluating the human's understanding of the model's (M) decision-making process. In other words, given an image, it evaluates whether the users could reliably predict the model's output decision. More concretely, let us consider that M predicts images in a set C correctly and makes incorrect decisions on the images in the set W. Justified trust is given as sum of the percentage of images in C that the human subject thinks M would correctly predict and the percentage of images in W that the human subject thinks M would fail to predict correctly.

2. ***Explanation Satisfaction (ES)*** (Qualitative Metric). We measure human subjects' feeling of satisfaction at having achieved an understanding of the machine in terms of usefulness, sufficiency, appropriated detail, confidence, and accuracy (Hoffman et al. 2018; Hoffman 2017). We ask the subjects to rate each of these metrics on a Likert scale of 0 to 9.

[1] We empirically observed that the metrics Justified Trust and Explanation Satisfaction are effective in evaluating the core objective of XAI, i.e. to evaluate whether the user's understanding of the model improves with explanations. These metrics are originally defined at a high-level in the work by (Hoffman et al. 2018) and we adapt them for the image classification task.

We used ILSVRC2012 dataset (Imagenet) (Russakovsky et al. 2015) and considered VGG-16 (Simonyan and Zisserman 2014) as the underlying network model. We randomly chose 40 classes in the dataset for our experiments and identified 46 xconcepts using our algorithm[2].

We applied between-subject design and randomly assigned subjects into ten groups. We perform this separately with expert user pool and non-expert user pool. Subjects in non-expert pool have no background in computer vision, whereas subjects in expert pool are experienced in training an image classification model using CNN. Each group in the non-expert pool are assigned 6 subjects and each group in the expert pool are assigned 2 subjects. Within each group, each subject will first go through a familiarization phase where the subjects become familiar with the underlying model through explanations (with 15 training images), followed by a testing phase where we apply our evaluation metrics and assess their understanding (on 5 test images) in the underlying model. Specifically, in the familiarization phase, human will be shown the input image I and the CNN's prediction c_{pred} and asked to provide c_{alt} as input. We will then show an explanation to the human user for the model's prediction c_{pred}. For example, in CoCoX group, we show the fault-line explaining why the model chose c_{pred} instead of c_{alt}. In the testing phase, human will be given only I and will not see c_{pred}, c_{alt}, and explanations, and we evaluate whether the human can correctly identify c_{pred} based on his/her understanding of the model gained in the familiarization phase.

For the first group, called NO-X (short for no-explanation group), we show the model's classification output on all the 15 images in the familiarization phase but we do not provide any explanation for the model's prediction. For the subjects in groups two to nine, in addition to the model's classification output, we also provide explanations in the familiarization phase for the model's prediction generated using the following state-of-the-art XAI models respectively: CAM (Zhou et al. 2016), Grad-CAM (Selvaraju et al. 2017), LIME (Ribeiro, Singh, and Guestrin 2016), LRP (Bach et al. 2015), SmoothGrad (Smilkov et al. 2017), TCAV (Kim et al. 2018), CEM (Dhurandhar et al. 2018), and CVE (Goyal et al. 2019). For the subjects in the tenth group, we show the fault-line explanations generated by our CoCoX model in addition to the classification output. It may be noted that, in the testing phase, human will be shown only the image I and will not be provided c_{pred}, c_{alt}, and explanations.

Results

Table 1 compares the Justified Trust (JT) and Explanation Satisfaction (ES) of all the ten groups in expert subject pool and non-expert subject pool. As we can see, JT and ES values of attention map based explanations such as Grad-CAM, CAM, and SmoothGrad do not differ significantly from the NO-X baseline, i.e., attention based explanations are not effective at increasing human trust and reliance (we did not evaluate ES for NO-X group as these subjects are not shown any explanations). This finding is consistent with the recent study by (Jain and Wallace 2019) which shows that attention is not an explanation. On the other hand, concept based explanation framework TCAV and counterfactual explanation frameworks CEM, and CVE performed significantly better than the NO-X baseline (in both expert and non-expert pool). Our CoCoX model, which is both conceptual and counterfactual, significantly outperformed all the baselines with 69.1% JT in non-expert pool and 70.5% JT in expert pool ($p < 0.01$). Interestingly, expert users preferred LRP (JT = 51.1%) to LIME (JT = 42.1%) and non-expert users preferred LIME (JT = 46.1%) to LRP (JT = 31.1%).

Furthermore, human subjects in the CoCoX group, compared to all the other baselines, found that explanations are highly useful, sufficient, understandable, detailed and are more confident in answering the questions in the testing phase. These findings verify our hypothesis that fault-line explanations are lucid and easy for both expert and non-expert users to understand.

Gain in Justified Trust over Time: We hypothesized that subjects' justified trust in the AI system might improve over time. This is because it can be harder for humans to fully understand the machine's underlying inference process in one single session. Therefore, we conduct an additional experiment with eight human subjects (non-experts) for each group where the subjects' reliance was measured after every session. Note that each session consists of a familiarization phase followed by a testing phase. The results are shown in Figure 3(a). As we can see, the subjects' JT in CoCoX group increased at a higher-rate compared to other baselines. However, we did not find any significant increase in JT after fifth session across all the groups. This is consistent with our expectation that it is difficult for humans to focus on a task for longer periods [3]. It should be noted that the increase in JT with attention map based explanations such as Grad-CAM and CAM is not significant. This finding again demonstrates that attention maps are not effective to improve human trust.

Subjective Evaluation of Justified Trust: In addition to the quantitative evaluation of the justified trust, we also collect subjective trust values (on a Likert scale of 0 to 9) from the subjects. This helps in understanding to what extent the users think they trust the AI system. The results are shown in Figure 3(b). As we can see, these results are consistent with our quantitative trust measures except that qualitative trust in Grad-CAM, CAM, and SmoothGrad is lower compared to the NO-X group.

Case Study: Figure 4 shows examples of the xconcepts (cropped and rescaled for better view) identified using our approach. As we can see, our method successfully extracts semantically coherent xconcepts such as *pointed curves* of `deer`, *stripedness* of `zebra`, and *woolliness* of `deerhound` from the training dataset. Also the fault-lines generated by our method correctly identify the most critical

[2] We manually removed noisy xconcepts and fault-lines. We couldn't find an automatic approach to filter them. We found that xconcepts generated by (Ghorbani, Wexler, and Kim 2019) are less noisy and might help in generating more meaningful fault-lines. We leave this for future exploration.

[3] In the future, we also intend to experiment with subjects by arranging sessions over days or weeks instead of having continuous back to back sessions.

	XAI Framework	Justified Trust (±std)	Explanation Satisfaction (±std)				
			Confidence	Usefulness	Appropriate Detail	Understandability	Sufficiency
Non-Expert Subject Pool	Random Guessing	6.6 %	N/A	N/A	N/A	N/A	N/A
	NO-X	21.4 % ± 2.7 %	N/A	N/A	N/A	N/A	N/A
	CAM (Zhou et al. 2016)	24.0 % ± 1.9 %	4.2 ± 1.8	3.6 ± 0.8	2.2 ± 1.9	3.2 ± 0.9	2.6 ± 1.3
	Grad-CAM (Selvaraju et al. 2017)	29.2 % ± 3.1 %	4.1 ± 1.1	3.2 ± 1.9	3.0 ± 1.6	4.2 ± 1.1	3.2 ± 1.0
	LIME (Ribeiro, Singh, and Guestrin 2016)	46.1 % ± 1.2 %	5.1 ± 1.8	4.2 ± 1.6	3.9 ± 1.1	4.1 ± 2.0	4.3 ± 1.6
	LRP (Bach et al. 2015)	31.1 % ± 2.5 %	1.1 ± 2.2	2.8 ± 1.0	1.6 ± 1.7	2.8 ± 1.0	2.1 ± 1.8
	SmoothGrad (Smilkov et al. 2017)	37.6 % ± 2.9 %	1.4 ± 1.0	2.2 ± 1.8	2.8 ± 1.0	3.1 ± 0.8	2.9 ± 0.8
	TCAV (Kim et al. 2018)	49.7 % ± 3.3 %	3.6 ± 2.1	3.2 ± 1.8	3.3 ± 1.6	3.6 ± 2.1	3.9 ± 1.1
	CEM (Dhurandhar et al. 2018)	51.0 % ± 2.1 %	4.1 ± 1.4	3.4 ± 1.4	3.1 ± 2.1	2.9 ± 0.9	3.3 ± 1.6
	CVE (Goyal et al. 2019)	50.9 % ± 3.0 %	3.8 ± 1.9	3.1 ± 0.9	3.6 ± 2.1	4.1 ± 1.2	4.2 ± 1.2
	CoCoX (Fault-lines)	**69.1 % ± 2.1 %**	**6.2 ± 1.2**	**6.6 ± 0.7**	**7.2 ± 0.9**	**7.1 ± 0.6**	**6.2 ± 0.8**
Expert Subject Pool	NO-X	28.1 % ± 4.1 %	N/A	N/A	N/A	N/A	N/A
	CAM (Zhou et al. 2016)	37.1 % ± 3.9 %	3.2 ± 1.8	3.3 ± 1.4	3.1 ± 2.1	3.1 ± 1.8	2.9 ± 1.9
	Grad-CAM (Selvaraju et al. 2017)	39.1 % ± 2.1 %	3.7 ± 1.2	3.1 ± 2.2	2.7 ± 1.9	3.7 ± 1.1	3.4 ± 1.6
	LIME (Ribeiro, Singh, and Guestrin 2016)	42.1 % ± 3.1 %	3.1 ± 2.2	3.0 ± 1.2	2.8 ± 1.9	3.1 ± 2.2	2.8 ± 1.7
	LRP (Bach et al. 2015)	51.1 % ± 3.1 %	3.2 ± 4.1	3.5 ± 1.6	4.2 ± 1.5	4.3 ± 1.0	3.9 ± 0.9
	SmoothGrad (Smilkov et al. 2017)	40.7 % ± 2.1 %	3.1 ± 1.0	2.9 ± 1.2	3.8 ± 1.5	3.3 ± 1.1	3.1 ± 1.0
	TCAV (Kim et al. 2018)	55.1 % ± 3.3 %	3.9 ± 2.8	3.6 ± 1.6	4.1 ± 1.3	4.9 ± 1.2	3.9 ± 0.8
	CEM (Dhurandhar et al. 2018)	61.1 % ± 2.2 %	4.8 ± 1.6	3.7 ± 1.6	4.0 ± 1.2	3.7 ± 1.0	4.0 ± 1.1
	CVE (Goyal et al. 2019)	64.5 % ± 3.7 %	4.1 ± 2.3	3.9 ± 1.5	4.6 ± 1.5	4.5 ± 1.4	3.9 ± 1.2
	CoCoX (Fault-lines)	**70.5 % ± 1.3 %**	**5.7 ± 1.1**	**4.9 ± 0.8**	**5.8 ± 1.2**	**6.9 ± 1.1**	**6.4 ± 1.0**

Table 1: Quantitative (Justified Trust) and Qualitative (Explanation Satisfaction) comparison of CoCoX with random guessing baseline, no explanation (NO-X) baseline, and other state-of-the-art XAI frameworks such as CAM, Grad-CAM, LIME, LRP, SmoothGrad, TCAV, CEM, and CVE.

xconcepts that can alter the classification result from c_{pred} to c_{alt}. For example, consider the image of `deerhound` shown in the Figure 4. Our fault-line explanation suggests removing *woolliness* and adding *black and white pattern* to alter the model's classification on the image from `deerhound` to `greyhound`.

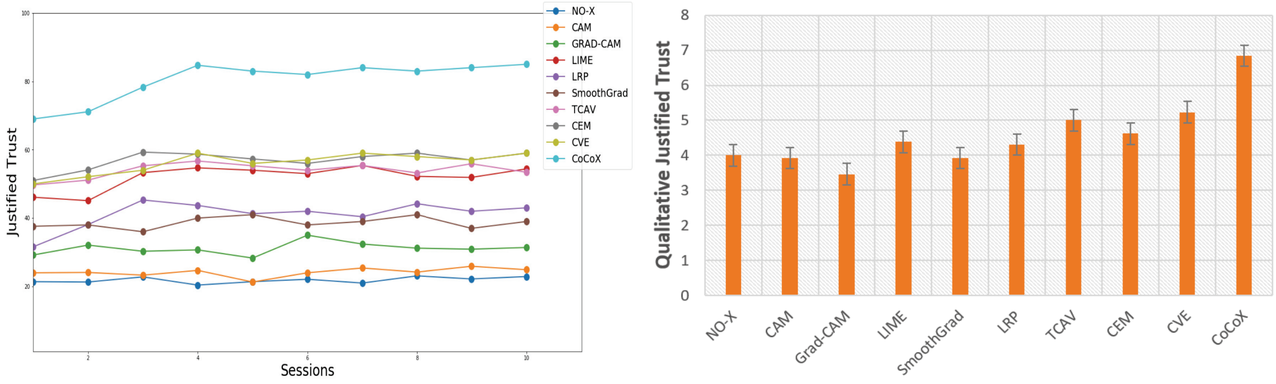
Related Work

Most prior work has focused on generating explanations using feature visualization and attribution.

Feature visualization techniques typically identify qualitative interpretations of features used for making predictions or decisions. For example, gradient ascent optimization is used in the image space to visualize the hidden feature layers of unsupervised deep architectures (Erhan et al. 2009). Also, convolutional layers are visualized by reconstructing the input of each layer from its output (Zeiler and Fergus 2014). Recent visual explanation models seek to jointly classify the image and explain why the predicted class label is appropriate for the image (Hendricks et al. 2016). Other related work includes a visualization-based explanation framework for Naive Bayes classifiers (Szafron et al. 2003), an interpretable character-level language models for analyzing the predictions in RNNs (Karpathy, Johnson, and Fei-Fei 2015), and an interactive visualization for facilitating analysis of RNN hidden states (Strobelt et al. 2016).

Attribution is a set of techniques that highlight pixels of the input image (saliency maps) that most caused the output classification. Gradient-based visualization methods (Zhou et al. 2016; Selvaraju et al. 2017) have been proposed to extract image regions responsible for the network output. The LIME method proposed by (Ribeiro, Singh, and Guestrin 2016) explains predictions of any classifier by approximating it locally with an interpretable model.

There are few recent works in the XAI literature that go beyond the pixel-level explanations. For example, the TCAV technique proposed by (Kim et al. 2018) aims to generate explanations based on high-level user defined concepts. Contrastive explanations are proposed by (Dhurandhar et al. 2018) to identify minimal and sufficient features to justify the classification result. (Goyal et al. 2019) proposed counterfactual visual explanations that identify how the input



(a)

(b)

Figure 3: (a) Gain in Justified Trust over time. (b) Average Qualitative Justified Trust (on a Likert scale of 0 to 9). Error bars denote standard errors of the means.

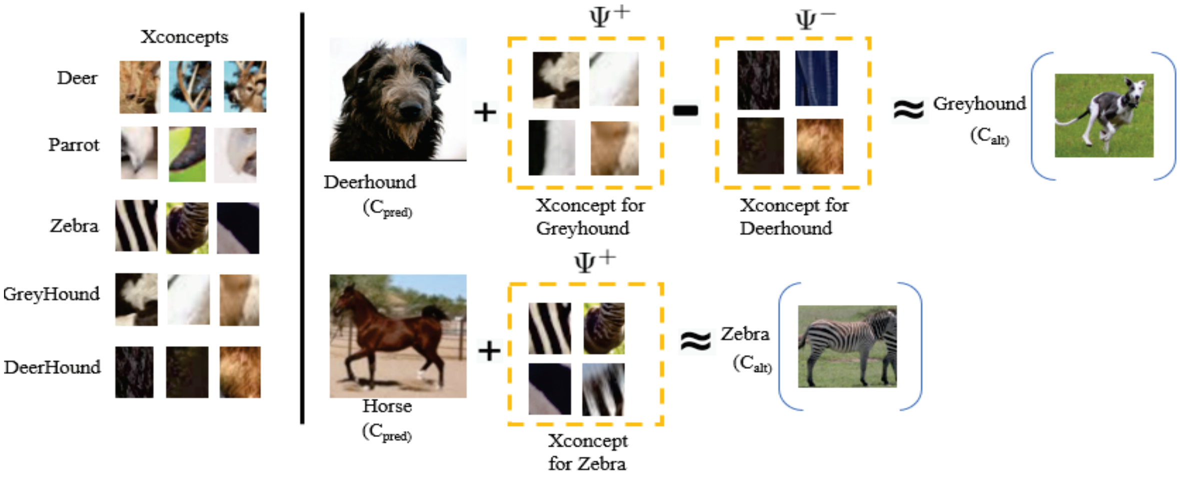


Figure 4: Examples of xconcepts (**Left**) and fault-line explanations (**Right**) identified by our method.

could change such that the underlying vision system would make a different decision. More recently, few methods have been developed for building models which are intrinsically interpretable (Zhang, Nian Wu, and Zhu 2018). In addition, there are several works (Miller 2018) on the goodness measures of explanations which aim to assess the underlying characteristics of explanations.

Conclusions

In this paper, we introduced a new explainable AI (XAI) framework, CoCoX, based on fault-lines. We argue that due to their conceptual and counterfactual nature, fault-line based explanations are lucid, clear and easy for humans to understand. We proposed a new method to automatically mine explainable concepts from a given training dataset and to derive fault-line explanations. Using qualitative and quantitative evaluation metrics, we demonstrated that fault-lines significantly outperform baselines in improving human understanding of the underlying classification model.

Acknowledgments

The authors thank Prof. Joyce Y Chai (UMich), Prof. Sinisa Todorovic (OSU), Prof. Hongjing Lu (UCLA), Prof. Devi Parikh (Georgia Tech), Prof. Dhruv Batra (Georgia Tech), Prof. Stefan Lee (OSU), Lawrence Chen (UCLA), and Yuhe Gao (UCLA) for helpful discussions. This work reported herein is supported by DARPA XAI N66001-17-2-4029.

References

Akula, A.; Liu, C.; Todorovic, S.; Chai, J.; and Zhu, S.-C. 2019a. Explainable ai as collaborative task solving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 91–94.

Akula, A. R.; Liu, C.; Saba-Sadiya, S.; Lu, H.; Todorovic, S.; Chai, J. Y.; and Zhu, S.-C. 2019b. X-tom: Explaining with theory-of-mind for gaining justified human trust. *arXiv preprint arXiv:1909.06907*.

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10(7):e0130140.

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* 2(1):183–202.

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.

Erhan, D.; Bengio, Y.; Courville, A.; and Vincent, P. 2009. Visualizing higher-layer features of a deep network. *University of Montreal* 1341(3):1.

Ghorbani, A.; Wexler, J.; and Kim, B. 2019. Automating interpretability: Discovering and testing visual concepts learned by neural networks. *arXiv preprint arXiv:1902.03129*.

Goyal, Y.; Wu, Z.; Ernst, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Counterfactual visual explanations. In *ICML 2019*.

Hendricks, L. A.; Akata, Z.; Rohrbach, M.; Donahue, J.; Schiele, B.; and Darrell, T. 2016. Generating visual explanations. In *European Conference on Computer Vision*, 3–19. Springer.

Hendricks, L. A.; Hu, R.; Darrell, T.; and Akata, Z. 2018. Generating counterfactual explanations with natural language. In *ICML Workshop on Human Interpretability in Machine Learning*.

Hoffman, R. R.; Mueller, S. T.; Klein, G.; and Litman, J. 2018. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

Hoffman, R. 2017. A taxonomy of emergent trusting in the human–machine relationship. *Cognitive systems engineering: The future for a changing world*.

Jain, S., and Wallace, B. C. 2019. Attention is not explanation. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Kahneman, D., and Tversky, A. 1981. The simulation heuristic. Technical report, STANFORD UNIV CA DEPT OF PSYCHOLOGY.

Karpathy, A.; Johnson, J.; and Fei-Fei, L. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, 2673–2682.

Kim, B.; Rudin, C.; and Shah, J. A. 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, 1952–1960.

Lipton, Z. C. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Miller, T. 2018. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*.

R Akula, A.; Todorovic, S.; Y Chai, J.; and Zhu, S.-C. 2019. Natural language interaction with explainable ai models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 87–90.

Ramprasaath, R.; Abhishek, D.; Ramakrishna, V.; Michael, C.; Devi, P.; and Dhruv, B. 2016. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CVPR 2016*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. ACM.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115(3):211–252.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 618–626.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Strobelt, H.; Gehrmann, S.; Huber, B.; Pfister, H.; and Rush, A. M. 2016. Visual analysis of hidden state dynamics in recurrent neural networks. *arXiv preprint arXiv:1606.07461*.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3319–3328. JMLR. org.

Szafron, D.; Greiner, R.; Lu, P.; Wishart, D.; MacDonell, C.; Anvik, J.; Poulin, B.; Lu, Z.; and Eisner, R. 2003. Explaining naïve bayes classifications. *TR03-09, Department of Computing Science, University of Alberta*.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology* 31(2):2018.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, Q.; Nian Wu, Y.; and Zhu, S.-C. 2018. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8827–8836.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2921–2929. IEEE.