Conceptual Edits as Counterfactual Explanations

Giorgos Filandrianos[1], Konstantinos Thomas[1], Edmund Dervakos[1] and Giorgos Stamou[1]

[1]AILS lab, School of Electrical and Computer Engineering, National Technical University of Athens

Abstract

We propose a framework for generating counterfactual explanations of black-box classifiers, which answer the question "What has to change for this to be classified as X instead of Y?" in terms of given domain knowledge. Specifically, we identify minimal and meaningful "concept edits" which, when applied, change the prediction of a black-box classifier to a desired class. Furthermore, by accumulating multiple counterfactual explanations from interesting regions of a dataset, we propose a method to estimate a "global" counterfactual explanation for that region and a desired target class. We implement algorithms and show results from preliminary experiments employing CLEVR-Hans3 and COCO as datasets. The resulting explanations were useful, and even managed to unintendedly reveal a bias in the classifier's training set, which was unknown to us.

Keywords

Counterfactual Explanations, XAI, Knowledge Graphs, Description Logics

1. Introduction

Public concerns about biases within machine learning (ML) models have created an increased demand for transparent AI [1, 2]. End users are quickly realizing that, for them to confidently be able to count on the impressive outputs of AI models, those outputs need to be accompanied by proper explanations. Being able to assess the reasons behind an AI model's suggestion is an essential component of the trust that is needed for any organization, government, or professional to assuredly count on AI tools and incorporate them into their workflow. Unsurprisingly, this "black box" problem which gained popular traction with the introduction of ML tools to end-users, was already a pain point for researchers in the Deep Learning field for many years [3]. Vetting a deep model for flaws and biases has been analogous to trying to perform an autopsy on a brain in the hope of discerning its thoughts.

One of the more interesting techniques that are being tested in efforts of observing the causation behind model outputs are counterfactual explanations. Counterfactual explanations answer the question of "*What would have to change for something to be classified as X instead of Y*". A real, GDPR inspired [4, 5], example would be asking a bank's AI model that declined our loan "*What would I have to change for my loan to be approved?*". Those types of questions

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2022 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence (AAAI-MAKE 2022), Stanford University, Palo Alto, California, USA, March 21–23, 2022.

geofila@islab.ntua.gr (G. Filandrianos); konstantinos.thomas@gmail.com (K. Thomas); eddiedervakos@islab.ntua.gr (E. Dervakos); gstam@cs.ntua.gr (G. Stamou)

0000-0001-7838-3919 (E. Dervakos); 0000-0003-1210-9874 (G. Stamou)

have a whole spectrum of plausible combinations as answers. The algorithm's goal is to find the one requiring the least amount of change, customized for that particular situation, while also being feasible and actionable in the real world. A key element of counterfactuals is their reliance on notions of distance and similarity [6]. In our approach, we propose employing a measure of conceptual distance of data samples combined with the amount of change at the output of a black-box classifier as a criterion for getting counterfactual explanations.

There are many approaches to counterfactual explanations in recent literature. Poyiadzi *et al.* [7] define counterfactual explanations as "feasible paths" in the data, which respect the data distribution and satisfy feasibility and actionability constraints. We take inspiration from this work and attempt to incorporate the constraints in our approach. Goyal *et al.*[8] propose a method for detecting which regions in an image should be changed, by "opening" the black box and utilizing the features extracted in the first layers of a deep neural network. They approach the problem as a "minimum edit" problem which is close to the way we approach the problem, with some important differences being: we do not require access to the model's weights, the explanations we provide have the form of concept edits instead of pixel edits, and our approach is suited for any domain besides images. Another method for the visual domain proposed by Zhao *et al.* [9], uses a text-to-image generative adversarial network to generate counterfactual visual explanations. A similarity of this approach with ours is that they utilize external knowledge and don't solely rely on the given features and classes for a model. For numeric tabular data, Gomez *et al.* [10] propose a heuristic method for detecting the minimal set of changes required for the prediction of a classifier to change, and provide a visualization tool for end-users. This approach is similar to ours in the sense that they compute a minimal set of changes, however instead of being applied on continuous numerical features, ours is applied on concepts that are independent of the features which the classifier accepts at its input. For further reading, we refer to literature surveys on counterfactual explanations [11, 12].

The majority of AI today is data-driven, sub-symbolic machine learning. Models tend to be convoluted, algebraic matrices that are difficult for humans to interpret. Knowledge graphs [13], on the other hand, provide symbolic, background knowledge in a machine-readable and human-understandable format. Knowledge representation techniques seem like a promising complement to machine learning for providing meaningful explanations [14, 15]. For instance Silva *et al.* [16] utilize knowledge graphs such as WordNet [17] in a composite approach for text entailment. They simultaneously outperform the state-of-the-art while explaining their predictions, thanks to the external knowledge. Liartis *et al.* [18] and Dervakos *et al.* [19] provide explanations for black-box classifiers, by attempting to mimic the classifier's behaviour with semantic query answering over external knowledge. Daniels *et al.* [20] propose exploiting the WordNet hierarchy to perform scene classification from images with neural networks in an explainable fashion. Alirezaie *et al.* [21] utilize external ontological knowledge to explain the errors of a satellite image classifier. For further reading on knowledge graphs as a tool for explainability, we refer to the recent survey by Tiddi *et al.* [22]. Following this line of work, our approach to counterfactual explanations makes use of external knowledge graphs, in the form of concept hierarchies. Specifically, our contributions can be summarized as follows:

- We introduce a theoretical framework for representing and computing counterfactual explanations with respect to concepts that characterize data samples and are linked with

external knowledge, in the form of concept hierarchies.

- We propose using a conceptual edit distance which is adaptable by the assignment of costs via user input, in order to satisfy any real-world constraints. This edit distance is based on the concept hierarchy, and is closely related to other semantic distance/similarity measures which have been proposed over the years [23]
- We accumulate multiple counterfactual explanations, in order to generate a "global" explanation for a specific class (which we call generalized counterfactual explanations). To our knowledge, we are the first to explore global counterfactual explanations.
- We propose and implement algorithms for generating explanations in this context, and show results from preliminary experiments.

2. Background

For describing concepts and the relationships between them we use the notation from description logics [24]. Specifically, in this work we utilize data annotations which use a vocabulary linked to *concept hierarchies*, in the form of taxonomies. We choose the formalism of description logics because in the future we plan to expand our framework to work with more expressive knowledge besides taxonomies of concepts. Given a set of concept names CN, each of which represents an *atomic concept*, a *concept* C is defined as: $\mathsf{C} \coloneqq \mathsf{A}|\top|\bot$, where $\mathsf{A} \in \mathsf{CN}$, $\top$ is the universal concept and $\bot$ is the bottom concept. For defining relationships between concepts we use a TBox T, which is a set of terminological axioms of the form: $\mathsf{A} \sqsubseteq \mathsf{B}$, where A and B are atomic concepts, and $\sqsubseteq$ denotes *inclusion*. Inclusion is transitive, meaning: $(A \sqsubseteq B \text{ and } B \sqsubseteq C) \Rightarrow A \sqsubseteq C$. Such a TBox may be represented as a directed graph $G = (V, E)$, where there is a $1-1$ matching between vertices of the graph and concepts: $V \leftrightarrow \mathsf{CN} \cup \{\top\}$, and there is an edge from vertex v_1 which matches to concept $\mathsf{A_1}$ to vertex v_2 which matches to concept $\mathsf{A_2}$ iff $\mathsf{A_1} \sqsubseteq \mathsf{A_2} \in T$. Furthermore, we consider any concept which appears only on the right of inclusion axioms in the given TBox, is connected with an incoming edge from the node corresponding to $\top$. We will refer to this graph as the *TBox graph*. When we ignore the direction of edges in the TBox graph, we will refer to it as an *undirected TBox graph*. Throughout this paper we allow the assignment of positive weights to edges of an undirected TBox graph. Finally, we view black box classifiers F as *functions* $F : \mathcal{D} \to [0, 1]^c$, where $\mathcal{D}$ is the classifier's domain, i.e. what it expects at its input (ex. images, text, vectors), and c is the number of classes. With abuse of notation, when an element x is classified in class C, we write $F(x) = C$, while the output of the classifier for class C is written as $F_C(x)(\in [0, 1])$.

3. Interpreting Black-Box Classifiers with Terminology Based Counterfactual Explanations

An overview of our framework is shown in Figure 1. In order to generate explanations for a black-box classifier, we need a *terminology* in terms of which we express the explanations, in addition to a set of testing items for the classifier. Specifically, we need a dataset, where for each sample we require: a) features which can be fed to the classifier, and b) a semantic description of

the sample in the form of a set of concepts, in terms of which we will provide the explanations. In the general case, this set of concepts is linked to external knowledge, in the form of a TBox.

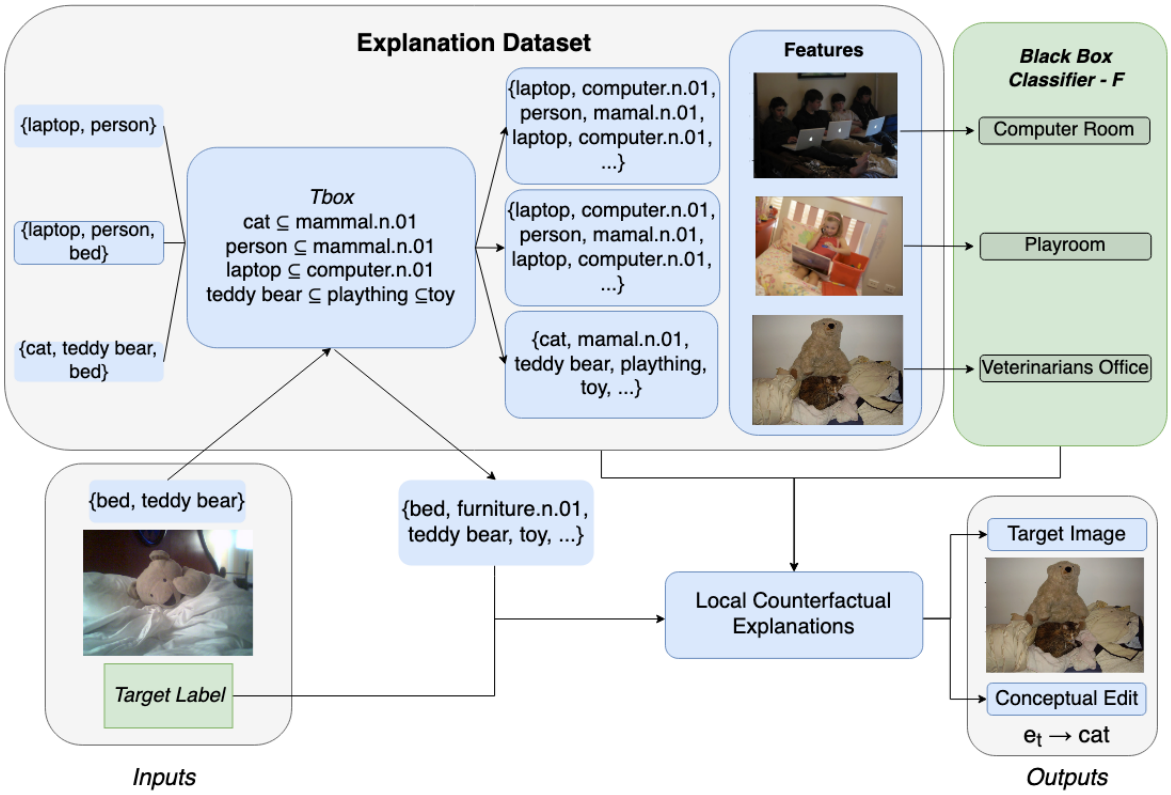

Figure 1: System Architecture

Definition 1 (Explanation Dataset). *Let $F : \mathcal{D} \rightarrow [0, 1]^c$ be a classifier. Given the domain $\mathcal{D}$ of the classifier to be explained, and a set of atomic concepts* CN*, an* ***explanation dataset*** *for F is a set of tuples $\{(x_i, C_i)\}$, where $x_i \in \mathcal{D}$ and $C_i \subseteq$ CN*

For instance, in an explanation dataset for image classifiers, the first element (x_i) of a tuple represents an image, while the second element (C_i) might be a set of concepts that describe objects in the image. In Section 5 we experiment on image classification explanation datasets CLEVR-Hans3 [25] and COCO [26]. In another example, the first element of a tuple might be raw text, which is fed to a black-box natural language model, while the second element might be a set of concepts from external knowledge such as WordNet [17] or ConceptNet[27] , or even domain-specific knowledge such as SNOMED-CT [28] for the medical domain, leading to a dataset similar to the one used in [29].

Given an explanation dataset, we can answer the question "*What has to change in order to be classified as X instead of Y?*" in terms of concepts (C_i), instead of features (x_i). In many cases, this leads to more intuitive explanations, especially in cases in which x_i is sub-symbolic raw

data (pixels, audio signals, etc), and C_i is linked to useful knowledge. More specifically, the explanations which we generate will have the form of edits on a set of concepts, where the cost of each edit is determined by the distance between concepts in the TBox graph.

Definition 2 (Concept Distance). *Let* CN *be a set of atomic concepts,* T *be a TBox for* CN*, and* G_T *be the corresponding undirected TBox graph. The* ***distance*** *from concept* A *to concept* B*, where* $\mathsf{A}, \mathsf{B} \in \mathsf{CN} \cup \{\top\}$ *is defined as the length of the shortest path on* G_T *from the vertex* v_{A} *to the vertex* v_{B}*, where* $v_{\mathsf{A}}, v_{\mathsf{B}}$ *are the vertices corresponding to atomic concepts* A, B*. We write* $d_T(\mathsf{A}, \mathsf{B})$ *to denote concept distance.*

For example if we were given the TBox $\{\mathsf{Cat} \sqsubseteq \mathsf{Mammal}, \mathsf{Dog} \sqsubseteq \mathsf{Mammal}, \mathsf{Ant} \sqsubseteq \mathsf{Insect}, \mathsf{Mammal} \sqsubseteq \mathsf{Animal}, \mathsf{Insect} \sqsubseteq \mathsf{Animal}\}$, then the concept distance from Cat to Dog would be 2, with the path on G_T being $\mathsf{Cat} \to \mathsf{Mammal} \to \mathsf{Dog}$. The concept distance from Cat to Ant would be 4 with the shortest path being $\mathsf{Cat} \to \mathsf{Mammal} \to \mathsf{Animal} \to \mathsf{Insect} \to \mathsf{Ant}$. Finally, the concept distance from Cat to $\top$ would be 3 with the path $\mathsf{Cat} \to \mathsf{Mammal} \to \mathsf{Animal} \to \top$.

We will use the notion of concept distance for assigning cost to edit operations on sets of concepts. These edit operations will end up being part of the counterfactual explantions.

Definition 3 (Concept Set Edit). *Let* CN *be a set of atomic concept names,* T *be a TBox for* CN*, and* $\mathcal{A} \subseteq \mathsf{CN}$ *be a set of concepts. A* ***concept set*** *edit on* $\mathcal{A}$ *is any of:*

- ***Replacement*** *of concept* $\mathsf{A} \in \mathcal{A}$ *with concept* $\mathsf{B} \notin \mathcal{A}$*. We write* $e_{\mathsf{A} \to \mathsf{B}}(\mathcal{A})$ *to denote replacement of* A *from* $\mathcal{A}$ *with* B*.*
- ***Deletion*** *of concept* $\mathsf{A} \in \mathcal{A}$ *from* $\mathcal{A}$*. We write* $e_{\mathsf{A} \to \top}(\mathcal{A})$ *to denote deletion of* A *from* $\mathcal{A}$*.*
- ***Insertion*** *of concept* $\mathsf{B} \notin \mathcal{A}$ *into* $\mathcal{A}$*. We write* $e_{\top \to \mathsf{B}}(\mathcal{A})$ *to denote insertion of* B *into* $\mathcal{A}$

The cost of a concept set edit $e_{x \to y}$ *is defined as the concept distance from* x *to* y*:* $d_T(x, y)$*, where* $x, y \in \mathsf{CN} \cup \{\top\}$*. The resulting set of concepts* $e_{x \to y}(\mathcal{A})$ *is called a transformation of* $\mathcal{A}$*.*

As mentioned in Section 2, we allow for the assignment of positive weights to the undirected TBox graph. This is done to allow for the incorporation of additional constraints to better reflect the actionability and feasibility of changes in the real world. In this work, we do not systematically do this, as we consider it to be given. For example for a given application, it might be useful to make the deletion of an Animal concept ($e_{\mathsf{Animal} \to \top}$) more costly than the replacement of a Cat concept with a Mammal concept ($e_{\mathsf{Cat} \to \mathsf{Mammal}}$), so we would appropriately tweak the edge weights of the undirected TBox graph.

As is apparent from the notation $e_{x \to y}$, we treat the deletion of concept A from a set as if being equivalent to its replacement with the universal concept $\top$, while the insertion of concept B is treated as if replacing a $\top$ concept with B. This entails that inserting or deleting a concept is more costly the further away it is from the $\top$ vertex in the TBox graph, which is a measure of how *specific* the concept is. Continuing the previous example, given a set $\mathcal{A}$, then inserting a Cat concept into the set would have a cost of $d_T(\top, \mathsf{Cat}) = 3$, while inserting an Animal concept into the set would have a cost of $d_T(\top, \mathsf{Animal}) = 1$. Using the concept set edit, we can define a concept set edit distance between sets of concepts, as the minimum cost of a set of concept set edits which when applied to the first set of concepts, transform it into the second.

Definition 4 (Concept Set Edit Distance). *Let* CN *be a set of concept names,* $\mathcal{T}$ *be a TBox on* CN *and* $\mathcal{A}, \mathcal{B}$ *be sets of concepts* $\mathcal{A}, \mathcal{B} \subseteq \mathsf{CN}$*. The* ***concept set edit distance*** *from* $\mathcal{A}$ *to* $\mathcal{B}$ *is defined as the minimum cost of a set of concept edits which transform* $\mathcal{A}$ *into* $\mathcal{B}$*.*

Intuitively, the concept set edit distance represents the minimum cost of converting every concept present in the first set, into every concept present in the second. For example, given two sets of concepts $\mathcal{A} = \{\mathsf{Cat}, \mathsf{Insect}\}$ and $\mathcal{B} = \{\mathsf{Animal}\}$, and the TBox from the previous example, then their concept set edit distance will be $D_{\mathcal{T}}(\mathcal{A}, \mathcal{B}) = \min\{[d_{\mathcal{T}}(\mathsf{Cat}, \mathsf{Animal}) + d_{\mathcal{T}}(\mathsf{Insect}, \top)], [d_{\mathcal{T}}(\mathsf{Cat}, \top) + d_{\mathcal{T}}(\mathsf{Insect}, \mathsf{Animal})]\} = \min\{(2+2), (3+1)\} = 4$.

In the context of our framework, the concept set edit distance is used to measure how conceptually similar two elements of an explanation dataset are and is one of the two key components for generating counterfactual explanations. The second component involves the black-box classifier which we want to explain. Specifically, we want counterfactual explanations to represent small conceptual changes (small concept set edit distance) which lead to large changes in the output of the classifier. For this reason, we define the *significance* of transforming an element of an explanation dataset into another.

Definition 5 (Significance of Transformation). *Let* F *be a classifier,* $\mathcal{T}$ *be a TBox and* $a = (x_a, C_a)$*,* $b = (x_b, C_b)$ *be two elements of an explanation dataset for* F*. The* ***significance of transforming*** a *into* b *is defined as:* $\sigma(a, b) = \frac{|F(x_a) - F(x_b)|}{D_{\mathcal{T}}(C_a, C_b)}$

Significance of transformations is the measure we use to determine what constitutes a good counterfactual explanation. The local explanations will have the form of a sequence of samples of the explanation dataset (similarly to the approach by Poyiadzi *et al.* [7]), but they will be accompanied by sets of concept set edits, which show what has to change conceptually in the data sample for the classification to change. In practice, we construct a directed graph where there is a node for each sample in the explanation dataset, and edges between pairs of nodes a, b have a cost of $\frac{1}{\sigma(a,b)}$ and as a label the set of edits corresponding to the conceptual distance $D_{\mathcal{T}}$. We then compute the shortest path from the given sample to any sample in the desired class, as described in Section 4.

Definition 6 (Local Counterfactual Explanation). *Let* $D = \{x_i, C_i\}$ *be an explanation dataset for a classifier* F*, and* $G = (V, E)$ *be a directed graph, where there is a 1-1 correspondence between elements of* D *and the set of vertices* V*. The set of edges* E *contains an edge of weight* $1/\sigma(a, b)$ *for every ordered pair of elements* $a, b \in D$*. Each edge* (a, b) *also has as a label a sets of concept set edits which optimally transform* C_a *into* C_b*. A* ***counterfactual explanation*** *from element* e *to a class* H *is a path from the node corresponding to* e *to any element* f *for which* $F(f) = H$*. Counterfactual explanations corresponding to a shortest path from* e *to any* f *are called* ***optimal counterfactual explanations****.*

The shorter the distance of a path corresponding to a local counterfactual explanation, the better the explanation is considered since short distances represent significant transformations.

Finally, besides acquiring a *local* explanation on how a single sample should be changed for it to be classified in a specific class, we are also concerned with more general explanations which give us an overview of which type of edits are more likely to lead towards being predicted

as a specific class, given a generalization of the initial sample. For example, a counterfactual explanation for a PhD student who is also a musician and was declined a loan might not be informative enough. This extension to *global* explanations would be able to answer the question "*What do musicians usually change to have their loan accepted*" and "*What do PhD students usually change to have their loan accepted*", which combined with its local explanation might be useful for the user to better understand why the black-box is making these decisions.

Definition 7 (Region of Explanation Dataset). *Let* CN *be a set of concept names,* $\mathcal{Q}$ *be a set of concepts* $\mathcal{Q} \subseteq \mathsf{CN}$*, and* $D = \{x_i, C_i\}$ *be an explanation dataset. A* ***region*** *of* D *with description* $\mathcal{Q}$ *is the subset* $R_{\mathcal{Q}} \subseteq D$ *of the explanation dataset for which:* $(x_i, C_i) \in R_{\mathcal{Q}} \iff \forall c_1 \in \mathcal{Q}, \exists c_2 \in C_i : c_2 \sqsubseteq c_1$

A region of an explanation dataset is a subset of it which satisfies specific constraints, it is essentially a query. For example given a region description $\mathcal{C} = \{\mathsf{Animal}\}$, then the region $R_{\mathcal{Q}}$ will contain any samples (x_i, C_i) of the explanation dataset which in their semantic description C_i contain any concept c which is included in Animal according to the TBox. Generalized counterfactual explanations will then be statistical measures on all optimal local counterfactual explanations from elements of a region. Specifically, they will measure how often a concept is introduced (either via replacement or insertion) and subtract how often they are removed.

Definition 8 (Generalized Counterfactual Explanation). *Let* $R_{\mathcal{Q}}$ *be a region of an explanation dataset, and* $E_{R_{\mathcal{Q}}}$ *be the multi set containing the labels of optimal local counterfactual explanations from each element of* $R_{\mathcal{Q}}$ *to the desired class. Given a set of concepts* $\mathcal{C} \subseteq \mathsf{CN}$*, a* ***generalized counterfactual explanation*** *is an assignment of importance to every concept* $\mathsf{C} \in \mathcal{C}$*. , where the importance of a concept* C *is defined as:* $\frac{|\{e_{x \to \mathsf{C}} \in E_{R_{\mathcal{Q}}}\}| - |\{e_{\mathsf{C} \to x} \in E_{R_{\mathcal{Q}}}\}|}{|R_{\mathcal{Q}}|}$*, where* $x \in \mathsf{CN}$

For example consider an explanation dataset for a classifier which determines if an image depicts a *bedroom* or a *veterinarian's office*. A region of this explanation dataset with a description of $\{\mathsf{Animal}\}$ might contain three elements: $(x_1, \{\mathsf{Cat}, \mathsf{Dog}\}), (x_2, \{\mathsf{Insect}\}), (x_3, \{\mathsf{Human}, \mathsf{Sofa}\})$. Let the first image be classified as *veterinarian's office*, while the other two are classified as *bedroom*. The optimal local counterfactual explanations from each element to the class *veterinarians office* might have labels: $E_1 = \emptyset$ (since x_1 is already classified in the desired class), $E_2 = \{e_{\top \to \mathsf{Human}}, e_{\mathsf{Insect} \to \mathsf{Cat}}\}$ and $E_3 = \{e_{\mathsf{Human} \to \mathsf{Cat}}, e_{\mathsf{Sofa} \to \top}\}$. Then the multiset $E_{R_{\mathcal{C}}}$ containing the labels of all optimal counterfactual explanations will be $E_{R_{\mathcal{C}}} = \{e_{\top \to \mathsf{Human}}, e_{\mathsf{Insect} \to \mathsf{Cat}}, e_{\mathsf{Human} \to \mathsf{Cat}}, e_{\mathsf{Sofa} \to \top}\}$. Then, a generalized counterfactual explanation for this region would be: a) Cat with importance $\frac{2}{3} = \frac{2}{3} - 0$, b) Insect and Sofa with importance $-\frac{1}{3} = 0 - \frac{1}{3}$, and c) Human with importance $0 = \frac{1}{3} - \frac{1}{3}$. Negative importance indicates that the concept is usually removed, while positive importance that it is introduced.

4. Computing Counterfactual Explanations

For generating the proposed counterfactual explanations in practice, we need a classifier F, an explanation dataset D, and a TBox T. There are three steps to computing explanations. The first step is to create the graph mentioned in Definition 6, the second step is to find appropriate

paths in this graph, and the third step (only for generalized counterfactual explanations) is to accumulate these paths and compute importance (from Definition 8). We show an outline of how we create the graph in Algorithm 1. For computing the concept distance between two concepts we find the shortest path on the undirected TBox graph using Dijkstra's algorithm with a complexity of $O(|\mathsf{CN}| + |T| \log |\mathsf{CN}|)$. For computing Concept Set Edit Distance (Definition 4) from set of concepts $\mathcal{A}$ to set of concepts $\mathcal{B}$, we first remove common elements from both sets, then we create a bipartite graph in $O(|\mathcal{A}||\mathcal{B}|)$, where each element of $\mathcal{A}$ is connected with an edge to all elements of $\mathcal{B}$ with a weight for each edge corresponding to the concept distance, and then we compute the minimum weight full matching of the bipartite graph by using an implementation of Karp's algorithm[30] for the problem with a time complexity of $O(|\mathcal{A}||\mathcal{B}| \log |\mathcal{B}|)$. Thus, to create the graph with Algorithm 1, the total time required will end up being $O((n + t \log n)m^4k^2 \log m)$, where $n = |\mathsf{CN}|$, m is the maximum cardinality of a set of concepts, k is the size of the explanation dataset and t is the size of the TBox. The creation of this graph is only done once per explanation dataset and TBox. To then compute local counterfactual explanations (Definition 6), we use Dijkstra's algorithm to find the shortest path, on the already constructed graph (including edge costs and labels) [1].

5. Experiments

The experimental objective we found was most in tune with counterfactual explanations was to test a classifier for biases. For example, if we trained a classifier on a set of images that included a "grey cube" in all the images of a specific class, would our counterfactual explanation result in an "add a grey cube" answer when asked how should we alter an image for it to belong in that class? This is precisely what we did with the CLEVR-Hans3 [25] dataset. Due to the control it provides on the generated images, and their accompanying description, it was the logical first step. Indeed, we found that our counterfactual algorithm was consistently able to detect these biases. Once this more technical task was accomplished, we sought to experiment on a more intuitive task. The advantage of such a task would be that it simulates a more real-world problem than 3D colored objects but, as with many real-world problems, it does not have an impartially correct bias. For example, what defines a bedroom as being a bedroom, and what makes a kitchen, a kitchen, in the eyes of the classifier? According to our counterfactual model, for instance, the classifier seems to think that a "bed" and a "refrigerator" are the defining factors of the above classes.

5.1. CLEVR-Hans3

The goal when experimenting with this dataset was twofold. First, we observe some image sequences representing local counterfactuals (ignoring the edit labels), to compare the results with results generated by the implementation provided for FACE [7]. Second, we want to see if the generalized counterfactuals can easily detect the (known) bias of the classifier.

[1]Code is available at: https://github.com/geofila/Conceptual-Edits-as-Counterfactual-Explanations

Algorithm 1: Explanation Graph Construction

Data: A classifier F, an explanation dataset D, an undirected TBox Graph G_T
Result: Explanation Graph G_E
//the explanation graph will have a node for each element in the explanation dataset
Initialize Directed Graph $G_E = (V_E = D, E_E = \emptyset)$;
foreach $(x_i, C_i) \in D$ **do**
  foreach $(x_j, C_j) \in D \setminus \{(x_i, C_i)\}$ **do**
    Initialize Graph $G_C = (V_C = C_i \cup C_j, E_C = \emptyset)$;
    foreach $k \in C_i$ **do**
      foreach $l \in C_j$ **do**
        //Compute concept distance using TBox graph
        $d_T(k, l) = |\mathsf{ShortestPath}(G_T, k, l)|$
        //Add an edge to G_C with weight d_T
        $E_C = E_C \cup \{(k, l, d_T)\}$
      end
    end
    //Compute minimum weight full matching of the bipartite graph G_C
    $\{(c_m, c_n)\}, w = \mathsf{MinFullMatch}(G_C)$
    //Concept Set Edit Distance
    $D_T(C_i, C_j) = w$
    //Compute inverse significance
    $\frac{1}{\sigma(i,j)} = \frac{D_T(C_i, C_j)}{|F(x_i) - F(x_j)|}$
    //Add an edge to the explanation graph G_E with weight $\frac{1}{\sigma}$ and as a label the edits corresponding to the minimum weight full match
    $E_E = E_E \cup \{(v_i, v_j, \frac{1}{\sigma(i,j)}, \{e_{c_m \to c_n}\})\}$
  end
end
return G_E

5.1.1. Setting

CLEVR-Hans3 is a dataset of images of sets of 3D geometric shapes which are split into three classes. For each image, information is provided regarding the objects which are present concerning their shape (Sphere, Cube, Cylinder), their size (Large, Small), their material (Metallic, Rubber) and their color (Blue, Yellow, Brown, Grey, Green, Purple, Cyan, Red). The three classes contain images that depict: a) a Large Cube and a Large Cylinder, b) a Small Metal Cube and a Small Sphere, and c) a Large Blue Sphere and a Small Yellow Sphere. Furthermore, the first two classes are confounded in the training set, with an intentionally added bias. For the first class, in the training set, the Large Cube is always Grey, while in the test set the color of the Large Cube is random. For the second class, the material of the Small Sphere is Metal in the training set, while in the test set the material of the Small Sphere is random. This means that we expect classifiers trained on the training set to be biased towards the confounding factor. As a classifier we trained a resnet34 [31] model which achieved 99% accuracy on the validation set (which is confounded), while for the test set the per-class F1 scores were class 0: 0.27, class 1: 0.54, class 2: 0.92. As expected the performance is poor for the confounded classes.

We created two explanation datasets, one from the training set (in order to be compared with

FACE which is intended to be run on the training set), and one from the test set in order to attempt to detect the biases acquired in training. As a set of concept names CN, we defined a concept for every combination of shape, size, material, and color (including the absence of any of the above), leading to $|\mathsf{CN}| = 4 \times 3 \times 3 \times 9 = 324$. As a TBox, we added an inclusion axiom from each concept in CN to any other concept with the same description where one element is missing. For example $\mathsf{GrayCube} \sqsubseteq \mathsf{Gray}$ and $\mathsf{GrayCube} \sqsubseteq \mathsf{Cube}$. This way we assigned sets of concepts to each element in the dataset, based on the descriptions provided by the creators of the dataset in the corresponding json files.

5.1.2. Local Counterfactuals

In Figure 2 we show local counterfactual explanations generated for three randomly selected images (first column) which were classified in class 1 (Small Metal Cube, Small Sphere - where the Small Sphere is always Metal only in the train set) with the target class being class 0 (Large Cube, Large Cylinder, where the Large Cube is always Grey only in the train set), at first using the FACE algorithm [7] (second column) and then using our proposed algorithm (third column). A first observation is that neither of the results is very intuitive, and we argue that the form of the explanations (sequence of samples from the training set) is the reason. A second observation is that our approach tends to keep the number of objects present in an image constant, which makes sense due to the cost of adding and deleting concepts instead of replacing them, while FACE which relies on the distribution of the dataset and operates on pixels, having no knowledge of the objects depicted, tends to transition to images which contain many objects. A final observation is that in both methodologies the color of the Large Cube in the target image is always Grey, which is expected since this experiment ran on the training set.

5.1.3. Generalized Counterfactuals

In Figures 3,4 we show two generalized counterfactual explanations. The first (fig.3) shows the importance of concepts for the region of the explanation dataset constructed from the test set of CLEVR-Hans3 which classified in class 1, with the target class being class 0, while the second (fig. 4) shows the importance of concepts for the same region, with the target class being class 2. As mentioned in section 3, negative importance indicates that a concept tends to be removed for the given region and target class, while positive importance indicates that it tends to be inserted.

A first observation is that the bias of the classifier is immediately detected for the confounded class 0. As mentioned previously, the confounding factor for class 0 is that the Large Cube is always Grey in the train set. This is apparent from the first three bars of the plot on the left, where the most important insertions seem to be the concepts: (Gray, GrayLargeCube, GrayLarge). The reason for which GrayLargeCube has a larger importance than GrayLarge is because, for some local counterfactuals, GrayLarge objects (which are not necessarily Cube) might be removed, thus lowering the importance of this concept. Class 2 on the other hand (Large Blue Sphere, Small Yellow Sphere) is not confounded, and the classifier is not expected to be biased (test F1 score of 0.92). The most important removals seem to be: combinations of Cube, Small, Metal - which makes sense since the source region contains images classified in

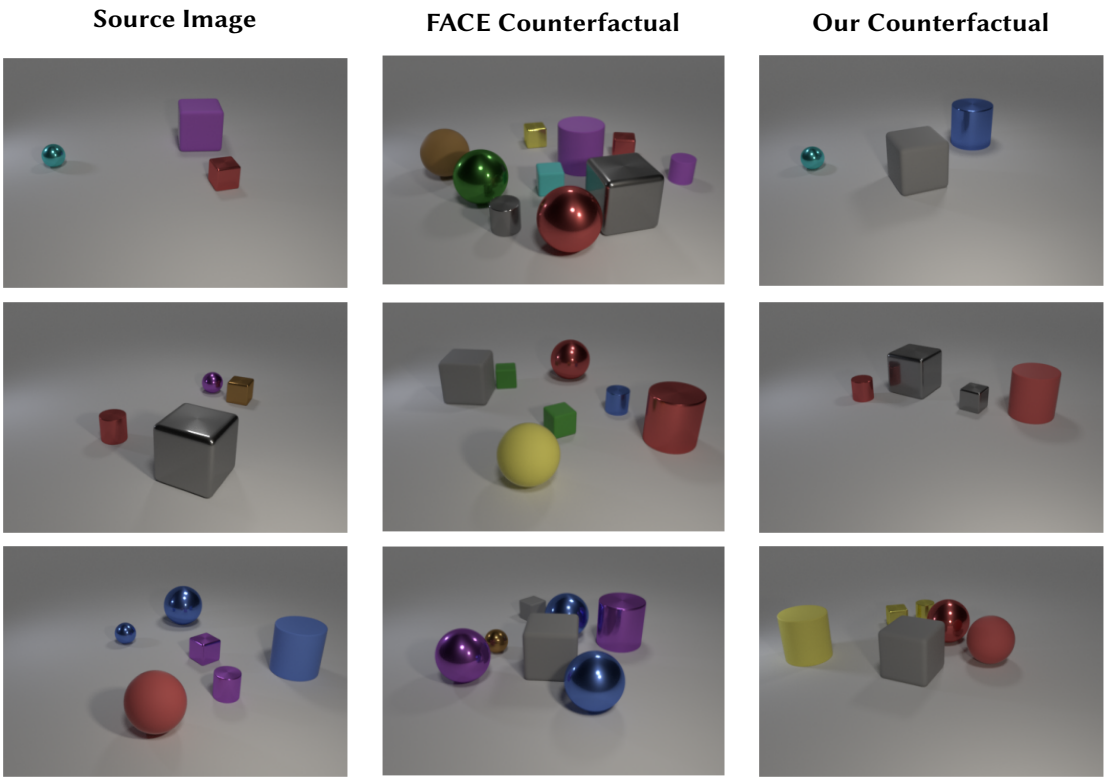

Figure 2: Counterfactuals for 3 randomly selected images (first column) which classified in class 1 with target class 0, using the FACE algorithm (second column) and our proposed method (third column)

class 1 (Small Metal Cube, Small Sphere, where the Small Sphere is always Metal in the train set). The most important insertions seem to be: Blue, Yellow, Sphere, and combinations of Blue, Large, Sphere which coincides with the definition of the class.

5.2. COCO

As a second experiment, we decided to explore some more intuitive examples and thus decided to take advantage of the COCO dataset [26], which contains real-world images, annotated with objects, which we can automatically link to external knowledge such as WordNet.

5.2.1. Setting

Examining COCO's labels in the process of determining a class transformation that will utilize them, we concluded that the two classes that should be used are "Restaurant" related and "Bedroom" related images. Specifically, for the *restaurant-related* class we gathered all images from COCO that contained the concepts: 1. {dining table, person, pizza} (1000+ images) 2. {dining table, person, wine glass} (1200+ images). For the *bedroom-related* class we gathered all images that contained the label combinations of: 1. {bed, person} (1300+ images) 2. {bed, book} (800+ images) 3. {bed, teddy bear} (300+ images). On top of that, we wanted to make sure that

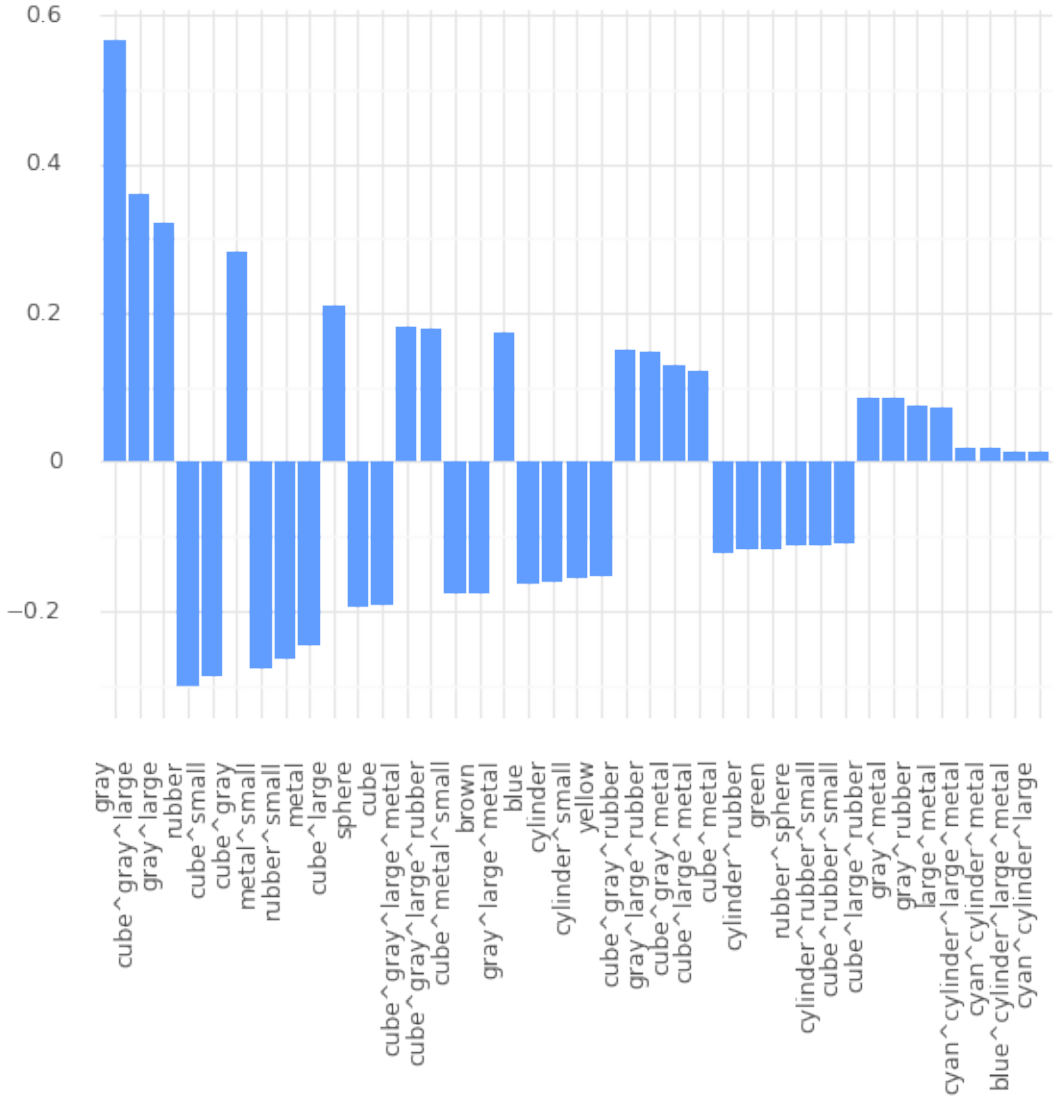

Figure 3: Generalized Counterfactual Explanation for the region of the explanation dataset for CLEVR-Hans3 which is classified in class 1, with the target class being class 0

we included some images that might be puzzling for the classifier. Those images were the ones including COCO label combinations of: 1. {bed, fork} (10 images) 2. {bed, spoon} (20 images) 3. {bed, wine glass} (20 images) 4. {bed, pizza} (10 images) 5. {dining table, bed} (170 images). For each image in COCO, a description of the objects present in that image is provided. To create the explanation dataset, we automatically linked these object descriptions with WordNet synsets by using the NLTK python package[2]. We used WordNet synsets as the set of concept names CN, and the hyponym-hypernym hierarchy as a TBox. We then acquired the image

[2]https://www.nltk.org/howto/wordnet.html

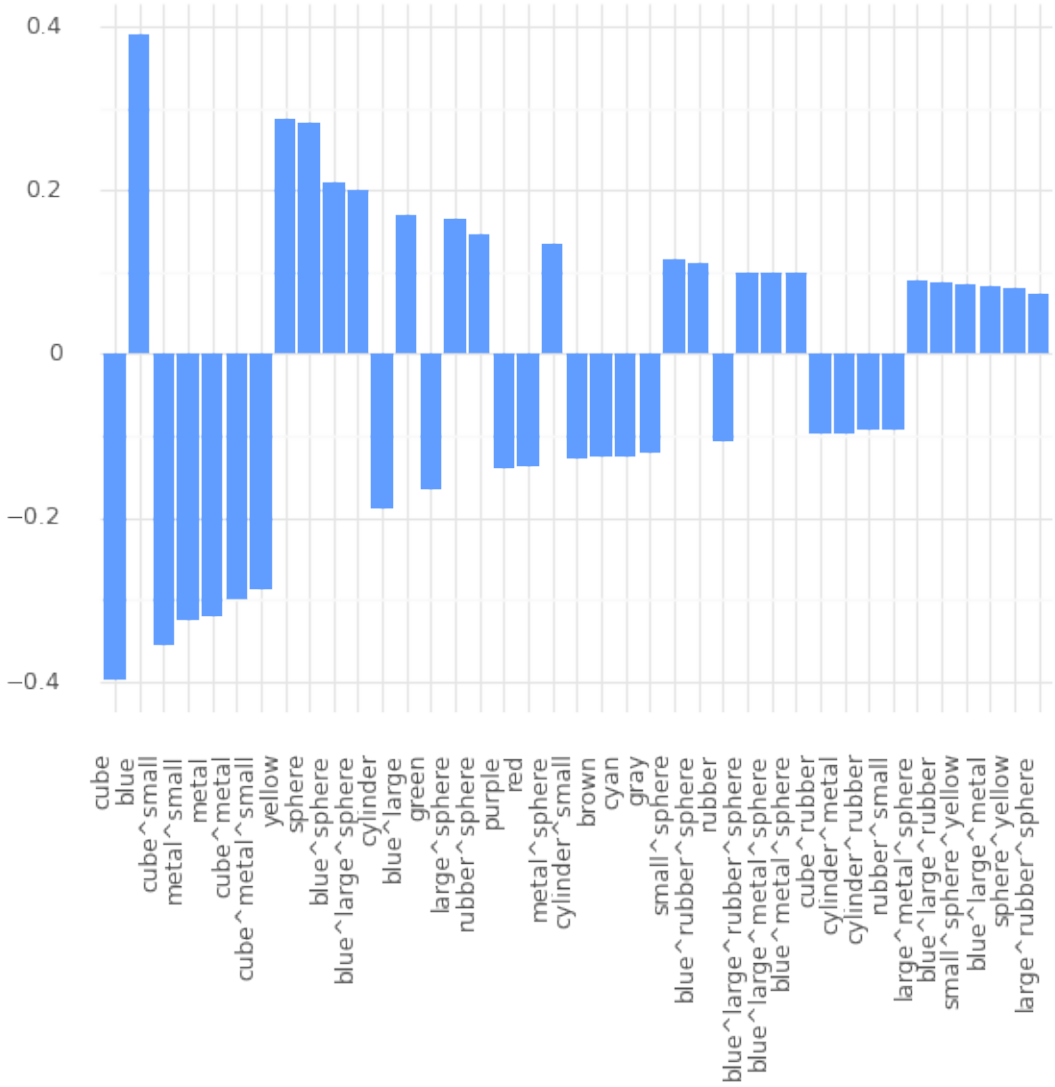

Figure 4: Generalized Counterfactual Explanation for the region of the explanation dataset for CLEVR-Hans3 which is classified in class 1, with the target class being class 2

classifier pre-trained on the PLACES dataset [32], provided by the creators of the dataset [3], for scene classification, and made predictions on the aforementioned subset of COCO. This is the black-box classifier for which we provide explanations.

5.2.2. Local Counterfactuals

In the first row of Figure 5 we show a local counterfactual explanation for an image classified as a "Bedroom" to the target class "Playhouse", which requires only one Concept Edit ($e_{\top \to \mathsf{Child}}$).

[3] http://places2.csail.mit.edu/index.htm

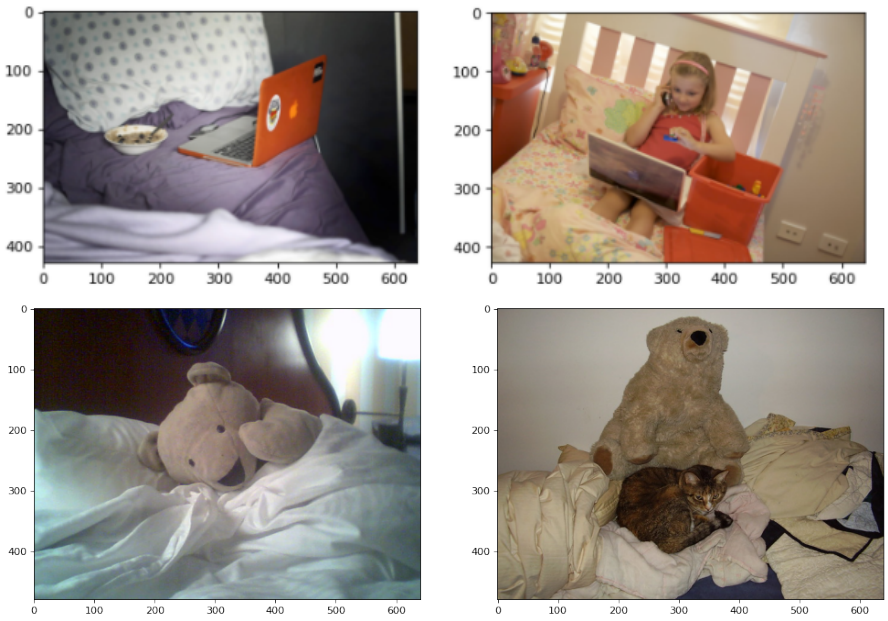

Figure 5: Counterfactual explanation for changing the prediction of the image on the left from 'Bedroom' to 'Playhouse' is simply to add a child ($e_{\top \to \mathsf{Child}}$) (top) and from 'Bedroom' to 'veterinarians office' is simply to add a cat ($e_{\top \to \mathsf{Cat}}$) (bottom).

This example is interesting because "Playhouse" is an erroneous prediction (the ground truth for the second image should be "Bedroom"), thus immediately we detect a potential bias of the classifier, that if a Child is added to an image of a "Bedroom" it might be classified as a "Playhouse". Similarly, in the second row of Figure 5 we show a local counterfactual explanation for an image which is classified as "Bedroom" to the target class "Veterinarian's Office", and the resulting target image is an erroneous prediction. The resulting edit is simply to add a Cat. Finally, in Figure 6 we show a counterfactual explanation, where the path on the graph has two steps. The source image is classified as a "Bedroom" and the target class is "Computer Room". This shows a smooth transition from the source image to the target class, by first adding a person (there are already two laptops in the source image), and then adding two more people and two more laptops.

5.2.3. Generalized Counterfactuals

In Figures 7,8 we see two examples of generalized counterfactual explanations on the COCO dataset. As before, each bar's numeric value shows the importance of the insertion (positive) or removal (negative) of that specific concept, in the process of transforming from a source region of an explanation dataset, to a target class.

Without revealing the source region and the target class for each figure, we can try to work out what those are, just by looking at the most frequent additions and removals. On the first (fig.7), which is the more trivial of the two, we see that the most common removals from the source images were concepts relevant to {furniture, bed, animal, carnivore, dog}, while the most

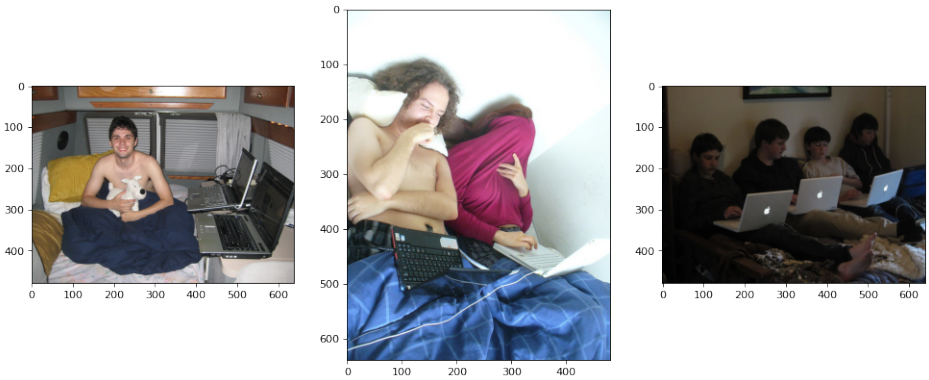

Figure 6: Counterfactual explanation for changing the prediction of the image on the left from “Bedroom” to “Computer Room”, which requires two steps

common additions were the concepts {home appliance, refrigerator, white goods, consumer goods}. From this, we can assume that the source region was likely bedroom images (with a bias towards pets) and the target class was probably a kitchen. The true classes were, indeed, "bedroom" and "kitchen". On the second (fig.8), we see that most frequent removals revolved around {instrumentality, artifact, electronic, furniture, telecommunications, TV, broadcasting, kitchen} and the most common additions around {carnivore, animal, mammal, feline, cat, dog}. Knowing that we are dealing with a classifier of rooms and places, we’d probably guess a kitchen for the source and a location with domestic animals for the target. The actual classes were "bedroom" targeting "veterinarian", which raises an interesting question: why did we see "kitchen" instead of "bed" in the bedroom class? The answer is that no beds were actually removed, since veterinarian office images tend to include beds. On the other hand, our dataset contains a number of studio-apartment bedroom images which had part of the kitchen appearing in the photo - kitchens that are mostly missing from a vet’s office and had to be removed. Another thing to note is that those examples were not cherry-picked. During our experiments we could, most of the time, estimate the source region and target class by looking at the edit frequencies. Notably, the most confusing results were when we tested the "computer room" target and found out that the generalized counterfactual explanation was very often adding people, but never laptops or computers. After investigating what seemed like a bug, we realized that most images from our dataset which were classified as a "computer room" had no computers in them, but people working in lab-appearing rooms.

5.3. Discussion

In our experiments we got interesting results, where both local and generalized counterfactual explanations seem to be informative, understandable and usable. In the CLEVR-Hans3 case (sec.5.1) we were able to detect the foreknown biases of the classifier, while in the COCO case (sec.5.2) we even detected unknown biases (for example the depiction of people was more important than that of laptops for the class “computer room”), and further insight into the classifier, which we had not thought about (for example that the classifier expects veterinarian’s

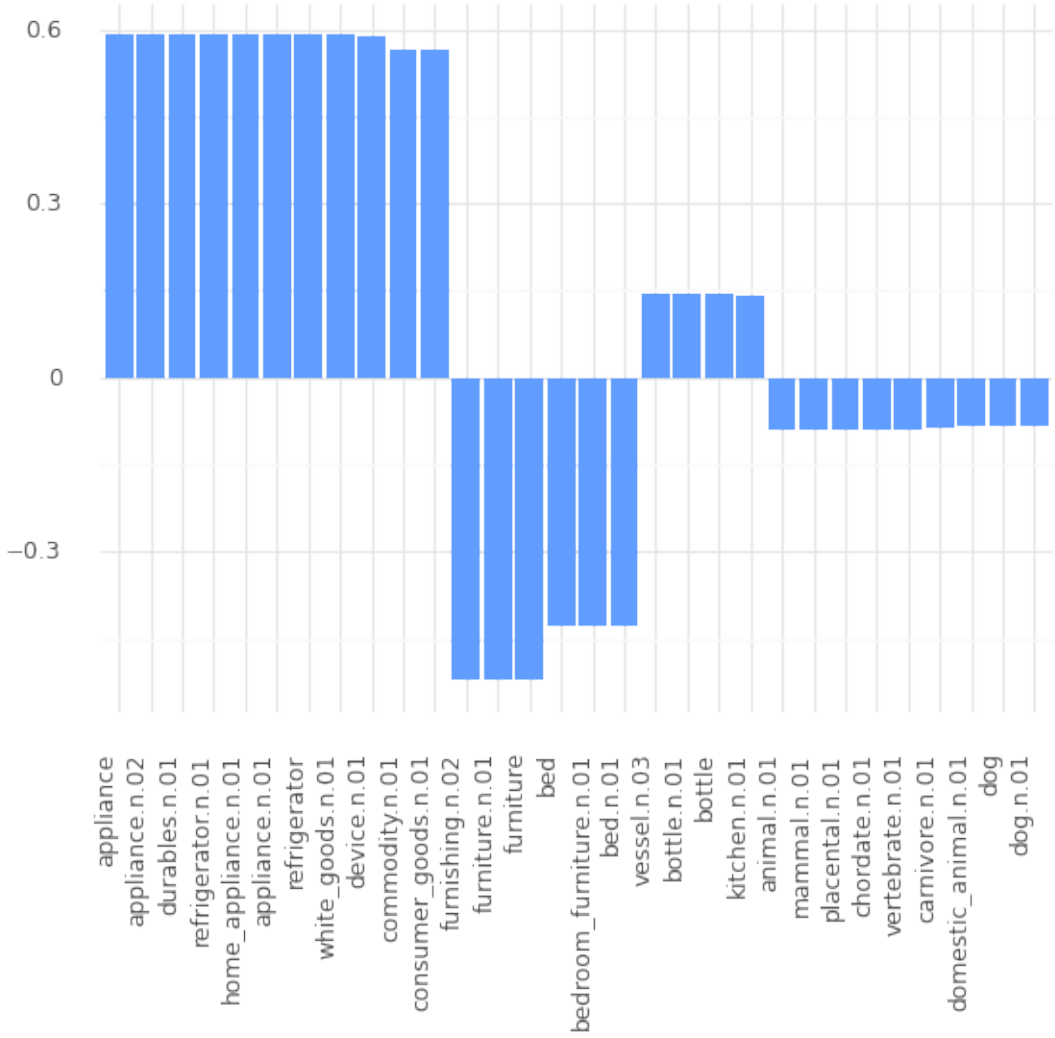

Figure 7: Generalized Counterfactual Explanations for the region of the explanation dataset for COCO which is classified as "bedroom", with the target class being "kitchen"

offices to depict beds among other objects). By comparing with the FACE algorithm (sec.5.1.2), we got a hint of the merits of considering explanations using high-level external terminology instead of low-level features, since even without stating the concept edits corresponding to counterfactual paths, we found the resulting images more intuitive and easy to compare with the source images.

An apparent drawback of the proposed framework, for it to be used in practice, is its reliance on the existence of semantically annotated data (i.e. an explanation dataset). Such datasets do exist for various domains, but they are not abundant. We have identified two ways of mitigating this drawback, which will be explored further in future work. The first is to semantically annotate data automatically by employing information extraction methods, such as object

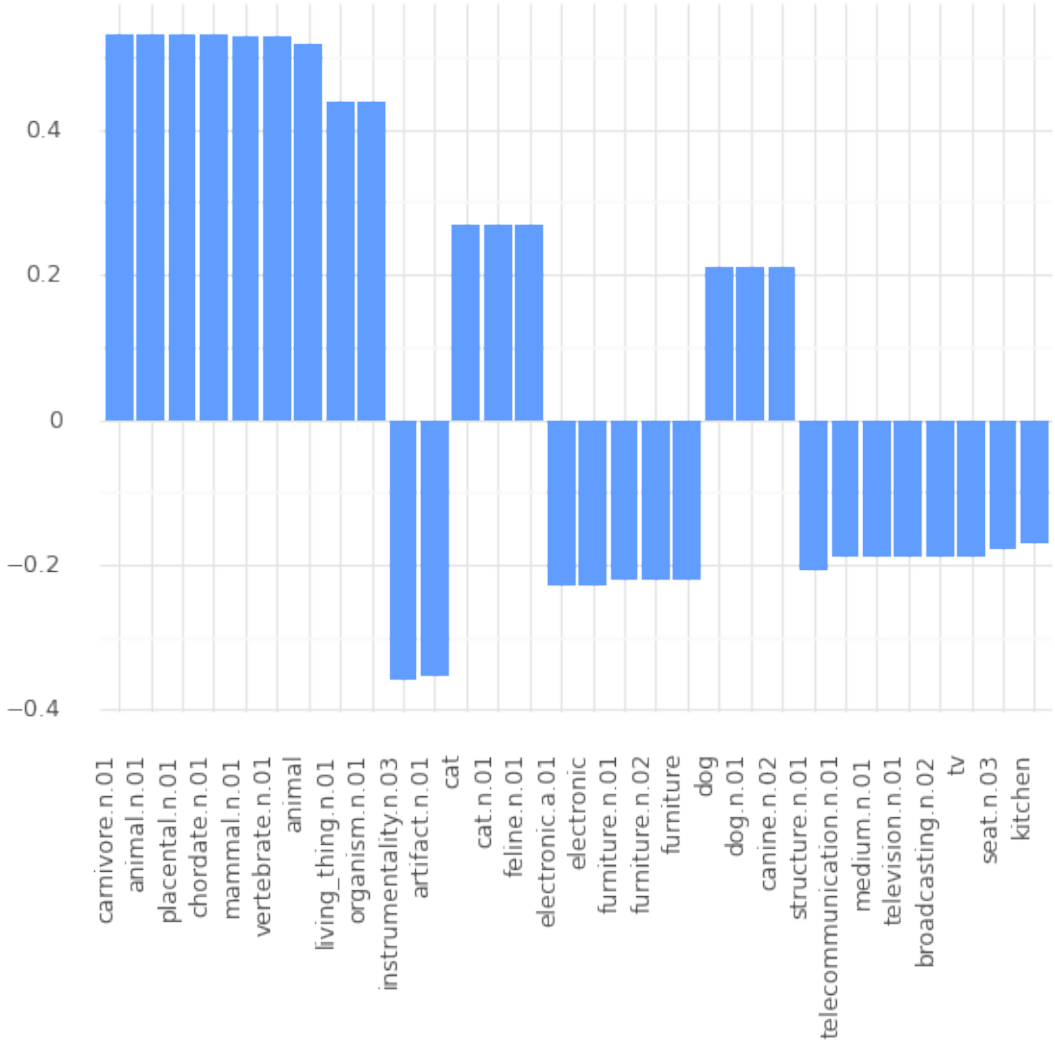

Figure 8: Generalized Counterfactual Explanations for the region of the explanation dataset for COCO which is classified as "bedroom", with target class "veterinarian"

detection or scene-graph generation for images, or other methods which automatically link entities to knowledge (for instance from text to encyclopedic knowledge [33]). The second way of mitigating this drawback, which is more suited for decision-critical domains such as medicine, is to invest resources for the manual annotation and curation of explanation datasets. We believe that having data manually characterized by domain experts could improve user awareness and trust of the generated explanations.

6. Conclusion

We have introduced a novel framework for representing and computing counterfactual explanations and have shown some preliminary results. There are many directions which we plan to explore in future work. First of all we plan to expand this framework further into Description Logics, to include roles and individuals, allow for more complex axioms in the TBox, and to explore how this effects the resulting explanations, both theoretically and practically. Furthermore, we plan to expand our evaluation framework to include datasets from multiple domains and applications, focusing on those where explainability is imperative, such as medical applications. We aim to experiment providing explanations for text, audio and tabular data. Ideally, the evaluation framework will include human evaluators in the future. Finally, we aim to study the properties of explanation datasets, as they are defined in our framework, and as they have been approached in other works [18],[19]. We will explore the effects of the size of an explanation dataset, by for example using the full COCO dataset. We will also experiment with using the same explanation dataset, linked to a different TBox (for example ConceptNet instead of WordNet), which will require us to also experiment with different notions of "conceptual" or "semantic" distance.

References

[1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2019) 93:1–93:42.

[2] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[3] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (XAI), IEEE Access 6 (2018) 52138–52160.

[4] B. Goodman, S. R. Flaxman, European union regulations on algorithmic decision-making and a "right to explanation", AI Mag. 38 (2017) 50–57.

[5] S. Wachter, B. D. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, CoRR abs/1711.00399 (2017).

[6] D. Lewis, Counterfactuals, John Wiley & Sons, 2013.

[7] R. Poyiadzi, K. Sokol, R. Santos-Rodríguez, T. D. Bie, P. A. Flach, FACE: feasible and actionable counterfactual explanations, in: AIES, ACM, 2020, pp. 344–350.

[8] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, S. Lee, Counterfactual visual explanations, in: ICML, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 2376–2384.

[9] W. Zhao, S. Oyama, M. Kurihara, Generating natural counterfactual visual explanations, in: IJCAI, ijcai.org, 2020, pp. 5204–5205.

[10] O. Gomez, S. Holter, J. Yuan, E. Bertini, Vice: visual counterfactual explanations for machine learning models, in: IUI, ACM, 2020, pp. 531–535.

[11] S. Verma, J. P. Dickerson, K. Hines, Counterfactual explanations for machine learning: A review, CoRR abs/2010.10596 (2020).

[12] I. Stepin, J. M. Alonso, A. Catalá, M. Pereira-Fariña, A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence, IEEE Access 9 (2021) 11974–12001.
[13] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutiérrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. F. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, CoRR abs/2003.02320 (2020).
[14] F. Lecue, On the role of knowledge graphs in explainable ai, Semantic Web 11 (2019) 1–11. doi:10.3233/SW-190374.
[15] G. Futia, A. Vetrò, On the integration of knowledge graphs into deep learning models for a more comprehensible ai—three challenges for future research, Information 11 (2020) 122.
[16] V. S. Silva, A. Freitas, S. Handschuh, Exploring knowledge graphs in an interpretable composite approach for text entailment, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 7023–7030.
[17] G. A. Miller, Wordnet: A lexical database for english, Commun. ACM 38 (1995) 39–41. URL: https://doi.org/10.1145/219717.219748. doi:10.1145/219717.219748.
[18] J. Liartis, E. Dervakos, O. Menis-Mastromichalakis, A. Chortaras, G. Stamou, Semantic queries explaining opaque machine learning classifiers, in: DAO-XAI, volume 2998 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021.
[19] E. Dervakos, O. Menis-Mastromichalakis, A. Chortaras, G. Stamou, Computing rule-based explanations of machine learning classifiers using knowledge graphs, arXiv preprint arXiv:2202.03971 (2022).
[20] Z. A. Daniels, L. D. Frank, C. J. Menart, M. Raymer, P. Hitzler, A framework for explainable deep neural models using external knowledge graphs, in: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II, volume 11413, International Society for Optics and Photonics, 2020, p. 114131C.
[21] M. Alirezaie, M. Längkvist, M. Sioutis, A. Loutfi, A symbolic approach for explaining errors in image classification tasks, in: Working Papers and Documents of the IJCAI-ECAI-2018 Workshop on, 2018.
[22] I. Tiddi, S. Schlobach, Knowledge graphs as tools for explainable machine learning: a survey, Artificial Intelligence (2021) 103627.
[23] A. A. Sánchez-Ruiz-Granados, S. Ontañón, P. A. González-Calero, E. Plaza, Refinement-based similarity measure over DL conjunctive queries, in: ICCBR, volume 7969 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 270–284.
[24] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.
[25] W. Stammer, P. Schramowski, K. Kersting, Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations, in: CVPR, Computer Vision Foundation / IEEE, 2021, pp. 3619–3629.
[26] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: common objects in context, in: ECCV (5), volume 8693 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 740–755.
[27] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general

knowledge, in: AAAI, AAAI Press, 2017, pp. 4444–4451.
[28] M. Q. Stearns, C. Price, K. A. Spackman, A. Y. Wang, SNOMED clinical terms: overview of the development process and project status, in: AMIA, AMIA, 2001.
[29] E. Dervakos, G. Filandrianos, K. Thomas, A. Mandalios, C. Zerva, G. Stamou, Semantic enrichment of pretrained embedding output for unsupervised IR, in: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, volume 2846 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021.
[30] R. M. Karp, An algorithm to solve the m× n assignment problem in expected time o (mn log n), Networks 10 (1980) 143–152.
[31] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, IEEE Computer Society, 2016, pp. 770–778.
[32] B. Zhou, À. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 1452–1464.
[33] R. Mihalcea, A. Csomai, Wikify! linking documents to encyclopedic knowledge, in: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 2007, pp. 233–242.