# Structure Your Data: Towards Semantic Graph Counterfactuals

**Angeliki Dimitriou** [1]   **Maria Lymperaiou** [1]   **Giorgos Filandrianos** [1]   **Konstantinos Thomas** [1]   **Giorgos Stamou** [1]

## Abstract

Counterfactual explanations (CEs) based on concepts are explanations that consider alternative scenarios to understand which high-level semantic features contributed to particular model predictions. In this work, we propose CEs based on the semantic graphs accompanying input data to achieve more descriptive, accurate, and human-aligned explanations. Building upon state-of-the-art (SotA) conceptual attempts, we adopt a model-agnostic edit-based approach and introduce leveraging GNNs for efficient Graph Edit Distance (GED) computation. With a focus on the visual domain, we represent images as scene graphs and obtain their GNN embeddings to bypass solving the NP-hard graph similarity problem for all input pairs, an integral part of CE computation process. We apply our method to benchmark and real-world datasets with varying difficulty and availability of semantic annotations. Testing on diverse classifiers, we find that our CEs outperform previous SotA explanation models based on semantics, including both white and black-box as well as conceptual and pixel-level approaches. Their superiority is proven quantitatively and qualitatively, as validated by human subjects, highlighting the significance of leveraging semantic edges in the presence of intricate relationships. Our model-agnostic graph-based approach is widely applicable and easily extensible, producing actionable explanations across different contexts. The code is available at https://github.com/aggeliki-dimitriou/SGCE.

## 1. Introduction

The AI landscape, now dominated by advanced Large Multimodal Models such as GPT4, GPT4V, and Gemini, high-
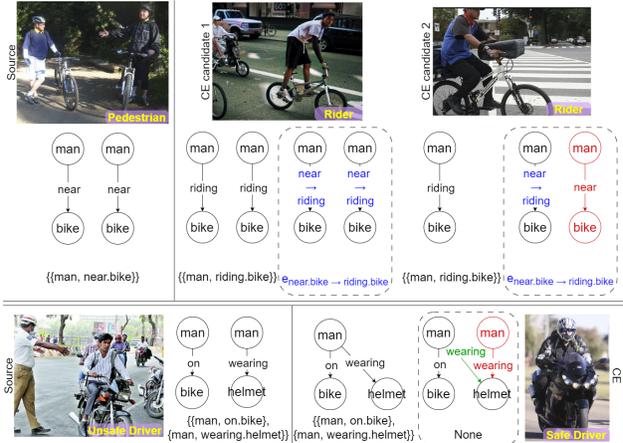


*Figure 1.* Examples where semantic graphs trump concept sets. Example 1 (top) shows the importance of the multiplicity of concepts for edit distance and example 2 (bottom) emphasizes the intricacy of relations. Edits (substitutions, insertions, deletions) are enclosed in striped rectangles. Images sourced from Visual Genome (Krishna et al., 2017), except unsafe driver (Deccan Chronicle, 2016).

lights the widespread use of proprietary models due to their state-of-the-art (SotA) performance across various modalities and datasets[1] (Team et al., 2023). This underscores the need for increased attention to black-box explainability methods, especially with the growing related applications in critical areas like medical image classification (Wu et al., 2023; Hou & Ji, 2024). Users should have the ability to understand decision-making processes without accessing the classifier architecture, emphasizing the importance of autonomy in scrutinizing proprietary models. To address this, there is a rising demand for post-hoc/model-agnostic explainability, an established field with publications in prestigious conferences (Ribeiro et al., 2016; Ying et al., 2019; Dervakos et al., 2023). In this spirit, this paper proposes a black-box method to compute Counterfactual Explanations (CEs) (Wachter et al., 2017) based on semantics. The challenges posed to extract and interpret decision processes of black-box models, although acknowledged as inherent trade-offs (Rudin, 2019), lie beyond the scope of this work.

The effectiveness of Conceptual XAI methods is closely tied to the semantic context of the data they interpret. In fact,

---

---

[1]https://openai.com/research/image-gpt

Browne & Swift (2020) report that 'there is no explanation without semantics', and formally prove that semantics are the distinguishing factor between CEs and adversarial examples. The role of annotations as an integral component in formulating conceptual CEs was first highlighted in Filandrianos et al. (2022), where the term "Explanation Dataset" was introduced. Its curation is also the initial step of the counterfactual computation pipeline proposed by the SotA CE work of Dervakos et al. (2023) (SC), which emphasizes its significance by urging users to 'choose their data wisely'. Given that the ultimate recipients of the explanations are humans, it is crucial to select annotations with precision and to actively engage domain experts in the process. This ensures that the explanations are not only accurate but also meaningful and relevant to the intended audience.

In the context of visual CEs, leveraging semantic annotations instead of superficial pixel-based features is significant, but not sufficient if their relations are not represented accurately. Our work addresses such limitations by *structuring the semantics as a graph*. In Fig. 1, we see a coarse representation of depicted concepts and their relations for source images and CE candidates, using our proposed semantic graphs versus the set of sets representation of SC. The explanations we provide include counterfactual images accompanied by the edit graphs from source to counterfactual. Fig. 1 (top) illustrates an example where employing sets under-represents the edit number by treating the two CE candidates as equals, despite the varying number of pedestrians (man riding bike) between them, potentially leading to a CE that is not optimal neither in terms of Graph Edit Distance (GED), nor visually. In other words, SC would consider a picture of a single rider the same as a photo of tens of cyclists. Fig 1 (bottom) depicts another problematic case for SC, where for the same CE the edit path is misleading. Using set representation, it is unclear that the rider lacks a helmet in the source image, creating the false sense of no required semantic edits. In contrast, our graph method recommends adding the 'wearing' role between 'man on bike' and 'helmet'. For 'safe' vs 'unsafe' driving classification, this edit path is crucial for explanations, both locally and globally. These two instances motivate the expressivity of graph-based explanations. By further linking semantics with external knowledge, we constrain edits to establish that concepts such as 'man' and 'woman' are more closely related than concepts like 'man' and 'helmet'; thus, boosting the interpretability and actionability of our method.

This work serves as an advancement of the previously presented method by Dervakos et al. (2023), which emphasizes leveraging semantically rich concepts to obtain CEs within model-agnostic settings, as long as the participating data instances are chosen wisely. However, their data representation lacks the proper incorporation of relationships between concepts, calling for a more intricate approach. We not

only employ graphs to structure semantic information as a direct refinement of prior work, but also leverage Graph Neural Networks (GNNs) for the efficient approximation of GED between graph instances to compute CEs. Our findings confirm the significance of correctly representing the number and interactions of concepts and our method significantly narrows the gap to the golden standard GED, achieving closer proximity with fewer edits. To underscore the efficacy of our approach against methods with access to the underlying model, we expanded the human survey from SC and compared our CEs with the white-box CE method by Vandenhende et al. (2022) (CVE), which, despite being pixel-level, emphasizes preserving semantic consistency. Our evaluation aimed to assess both human preferences and their capacity to comprehend and anticipate the classifier's output.

Our survey revealed that participants preferred our CEs in the majority of instances, and they were also successful in learning to accurately classify images themselves. This indicates that our approach surpasses the explanatory power of the white-box method of CVE in clarifying classifier logic to humans. This finding was further reinforced by replicating this experiment *exclusively providing semantic graphs and edits to the users, without any images*. Participants comprehended the classifier's reasoning and predicted outcomes effectively, even without the corresponding visual data.

We prove that our work surpasses prior SotA approaches. Our improvements have been quantitatively assessed and further substantiated by human surveys - a significant XAI evaluation tool. The compared methodologies are diverse in terms of reliance on the features classifiers exploit and the granularity of information. To combat the challenges of evaluating the explanations, we establish unified quantitative and qualitative metrics, applicable in all cases. Further elaboration is available in the Evaluation section of §4. Our key contributions are:

- Demonstrating **quantitative and qualitative superiority** over previous black- and white-box methods, paired with **enhanced adaptability** due to our model-agnostic design,

- Offering more **interpretable, expressive, and actionable** CEs using semantic graphs,

- Achieving **efficient GED approximation** via GNNs without compromising the representation of concept interactions.

Our method is **novel** in employing graphs and GNNs for counterfactual retrieval. We validate our approach across four diverse datasets (images & audio), using three neural and one non-neural classifier, including two human surveys

and four quantitative and qualitative experiments, yielding superior outcomes and time efficiency relative to SotA.

## 2. Related Work

**Counterfactual explanations** of visual classifiers encompass pixel-level edit methods which focus on marking and altering significant image areas that influence the model's predictions (Goyal et al., 2019; Vandenhende et al., 2022; Augustin et al., 2022). Contrary to other feature extraction counterfactual methods, the Counterfactual Visual Explanations (CVE) of Vandenhende et al. (2022) attempt to enforce semantically consistent area exchanges through an auxiliary semantic similarity component between local regions. Their semantically driven approach is the prime choice for pixel-level comparison, also providing a benchmark in contrast to techniques with direct classifier access, a feature distinguishing it from ours. Another vein of research focuses on human-interpretable concept edits to retrieve CEs. Abid et al. (2022) propose conceptual CEs in the event of a misclassification, using a white-box technique based on Concept Activation Vectors. The work by Dervakos et al. (2023) (SC) serves as an extension of Filandrianos et al. (2022), additionally emphasizing the importance of the Explanation Dataset, and expanding upon the original graph bipartite matching CE framework by leveraging the roles between concepts. Our work directly enhances these approaches, retaining advantageous qualities such as their model-agnostic nature and definition of object/relation distance through ontologies. Instead of leveraging Set Edit Distance and ignoring edges altogether (Filandrianos et al., 2022) or rolling up the edges into concepts, thus sacrificing crucial object relation information (Dervakos et al., 2023), we use the more accurate GED. To this end, we introduce semantic graphs for increased expressivity and use GNNs to accelerate GED calculation. A preliminary GNN-based counterfactual analysis was thoroughly discussed in the concurrent work of (Dimitriou et al., 2024), which compares various Graph Machine Learning algorithms. Much like its predecessors, our current paper is centered on the visual domain and applied to other modalities as a use case. Consequently, we do not delve into the literature of audio CEs, for instance. Despite the vast literature on graph CEs (Prado-Romero et al., 2024), comparison to GNN explainers (Bajaj et al., 2021; Lucic et al., 2022) is not applicable here, since we propose utilizing semantic graphs corresponding to non-graph input data for CE computation. It is noteworthy that no existing method leverages GNNs for post-hoc CEs, a novel feature of our approach.

**Graph similarity** methods such as Graph Edit Distance (GED) (Sanfeliu & Fu, 1983) are computationally expensive, prompting the use of approximation algorithms. Considering neural approaches, the ones relevant to our work leverage GNNs (Bai et al., 2019; Li et al., 2019; Ranjan et al., 2022). As our paper focuses on embedding extraction to facilitate CEs instead of the similarity itself, we simply draw inspiration from previous approaches in implementing our GNN model for semantic graphs, by adopting the ideas of Siamese GNNs, graph-to-graph proximity training and Multi-Dimensional Scaling as loss (Bai et al., 2019) to preserve inter-graph distances in the embedding space.

## 3. Method

Since the majority of our experiments are conducted with visual classifiers, we will illustrate our framework within this domain (Fig. 2). Given a query image $I_{(A)}$ belonging to a class $A$, a conceptual CE entails finding another image $I'_{(B)} \neq I_{(A)}$ in a class $B \neq A$, so that the shortest edit path between $I_{(A)}$ and $I'_{(B)}$ is minimized. Even though there are different notions of distance between images, we select a *conceptual* representation, employing scene graphs to represent objects and interactions within images. To this end, the problem of image similarity ultimately reduces to a graph similarity challenge. However, graph edits (insertions, deletions, substitutions) as a deterministic measure of similarity between two graphs $G_{(A)}$ and $G'_{(B)}$ is an NP-hard problem. Optimal edit paths can be found through tree search algorithms with the requirement of exponential time. When searching for a counterfactual graph to $G_{(A)}$ among a set of $N$ graphs, GED needs to be calculated $N-1$ times. To minimize the computational burden, we use lightweight GNNs that accelerate the graph proximity process by mapping all $N$ graphs to the same embedding space. By retrieving the closest embedding to $G_{(A)}$ that belongs to class $B \neq A$, GED is computed only *once* per query during retrieval. Concretely, we approximate the following optimization problem for semantic graphs extracted from any input modality:

$$GED(min|G_{(A)}, G'_{(B)}|), \; such \; that \; A \neq B \qquad (1)$$

**Ground Truth Construction** As our overall approach does not rely on pre-annotated graph distances, we propose a technique to construct well-defined ground truth instances. The graph structure of data imposes the requirement of defining an absolute similarity metric between graph pairs for the training stage. GED is regarded as the optimal choice despite its computational complexity; computing GED for only $N/2$ pairs to construct the training set is adequate for achieving high quality representations, as validated experimentally. To further facilitate GED calculation, we exploit a suboptimal algorithm utilizing a bipartite heuristic that accelerates an already effective in practice LSAP-based algorithm for GED (Jonker & Volgenant, 1987; Fankhauser et al., 2011). Consequently, semantic information of nodes and edges should guide graph edits based on their conceptual similarity. Thus, we choose to deploy the technique
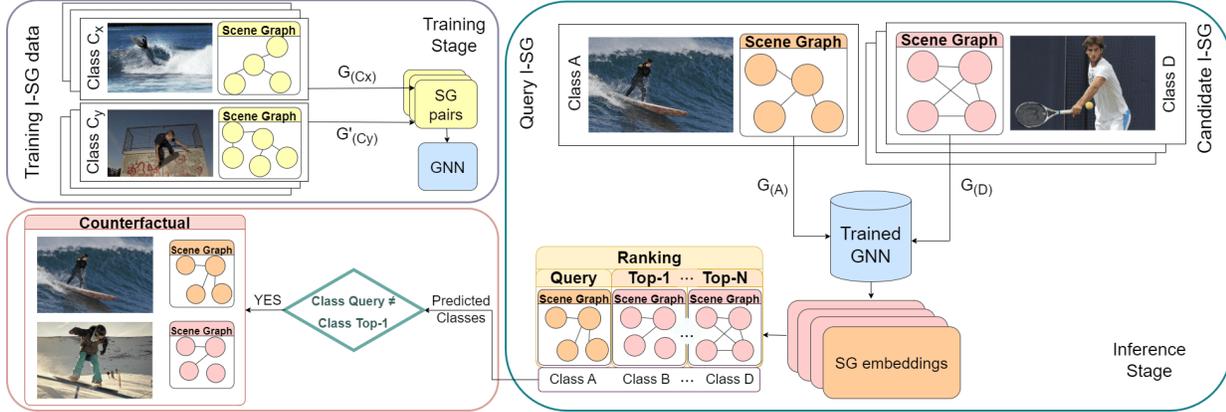
*Figure 2.* Method outline (for image classifiers). Depicted stages directly correspond to Sec. 3 paragraphs. *Predicted* class labels are: A - query, B - target, $C_x, C_y$ - any class, others - random class instances. Graph $G'_{(B)}$ corresponds to counterfactual image $I'_{(B)}$.

proposed in SC (Dervakos et al., 2023) to assign operation costs based on conceptual edit distance, as instructed by the shortest path between two concepts within the WordNet hierarchy (Miller, 1995).

**GNN Training** To accelerate the retrieval of the most similar graph $G'_{(B)}$ to graph $G_{(A)}$, we build a siamese GNN component for graph embedding extraction based on inter-graph proximity. The GNN comprises two identical node embedding units that receive a random graph pair $(G_{(C_x)}, G'_{(C_y)})$ as input ($C_x, C_y$ can be any class). The extracted node representations are pooled to produce global graph embeddings $(h_{G_{(C_x)}}, h_{G'_{(C_y)}})$. Embedding units consist of stacked GNN layers, described by either GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017) or GIN (Xu et al., 2018). We formalize GCN graph embedding computation in Eq. 2 (omitting class notation for simplicity):

$$h_G = \frac{1}{n} \sum_{i=1}^{n} (u_i^{K-1} + \sum_{j \in \mathcal{N}(i)} u_j^{K-1}) \quad (2)$$

where $u_i$ is the representation of node $i$, $\mathcal{N}(i)$ is the neighborhood of $i$, $n$ is the number of nodes for $G$ and $K$ is the number of GCN layers. To preserve the similarity of vectors $(h_{G_{(C_x)}}, h_{G'_{(C_y)}})$, we adopt the dimensionality reduction technique of Multi-Dimensional Scaling (Williams, 2000), as proposed in (Bai et al., 2019). The model is trained transductively to minimize the loss function $\mathcal{L}$:

$$\mathcal{L} = \mathbb{E}(\left\| (h_{G_{(C_x)}} - h_{G'_{(C_y)}}) \right\|_2^2 - GED(G_{(C_x)}, G'_{(C_y)})) \quad (3)$$

Graphs are embedded in a lower dimensional space by choosing a random subset of $\frac{N!}{2(N-2)!}$ pairs with varying cardinality $p$. As node features initialization is significant with regard to semantic similarity preservation, we use GloVe representations (Pennington et al., 2014) of node labels.

**Ranking and Counterfactual Retrieval** Once graph embeddings have been extracted, they are compared using cosine similarity to produce rankings. For each query image $I_{(A)}$ and subsequently its scene graph $G_{(A)}$, we obtain the instance $G'_{(B)}$ with the highest rank given the constraint that $I'_{(B)}$ is classified in $B \neq A$. $I'_{(B)}$ is proposed as a CE of $I_{(A)}$ since it constitutes the instance with the minimum graph edit path from it, classified in a different target category $B$. Specifically, we retrieve a scene graph $G'_{(B)}$ as:

$$G'_{(B)} = G^i_{(B)}, \ \arg\max_i \left( \frac{h_{G^i_{(B)}} \cdot h_{G_{(A)}}}{\left\| h_{G^i_{(B)}} \right\| \left\| h_{G_{(A)}} \right\|} \right) \ if \ B \neq A \quad (4)$$

where $i = 1, ..., N$. Selecting target class $B$ is correlated with the characteristics of the dataset in use and the goal of the explanation itself. Precisely, if the data instances have ground truth labels, the target class can be defined as the most commonly confused compared to the source image class (Vandenhende et al., 2022). Another valid choice is to arbitrarily pick $B$ to facilitate a particular application, i.e. explanation of classifier mistakes, in which case $B$ is the true class of the query image (Abid et al., 2022). We choose the first approach when ground truth class labels are available; otherwise, we define the target class as the one with the most highly ranked instance not classified as A.

## 4. Experiments

**Evaluation** comprises quantitative metrics, as well as human-in-the-loop experiments. Quantitative results are extracted by comparing the ranks retrieved based on our obtained graph embeddings to the ground truth ranks retrieved by GED. This type of analysis is not present for SC, despite its significance for objectively assessing CEs beyond intuitive metrics. The reported metrics are: 1) *average Precision@k (P@k)*: all top-k GED retrieved results are

considered relevant, 2) *binary P@k* and *binary NDCG@k*: only top-1 GED result is relevant and its position in retrieved ranks is emphasized through NDCG, 3) *average number of edits*: average number of node/edge insertions, deletions, and substitutions with different concepts, calculated post-hoc through GED to ensure fairness.

The use of GED rankings as the golden standard for evaluation is clearly motivated by previous work (SC) and reinforced by features like: a) its purely semantic nature, b) completeness in distance representation due to its reliance on graphs which accurately encompass both objects and relations, c) deterministic nature and applicability regardless of modality and granularity of the technique under evaluation. Wide applicability is especially important because baselines include pixel-based methods which define units of information differently (significant rectangular areas vs concepts). Thus, investigating the effectiveness of GED in depth or comparing it with other similar metrics would divert from the paper's main focus, as it is already accepted in the research community.

Human evaluation highlights several aspects of our contributions. First, to validate the quality of our retrieved CEs against SotA, we ask our evaluators (student volunteers of engineering backgrounds) to select among two CE alternatives of a query image; an image retrieved from our method versus an image retrieved either by SC or CVE. We also test the understandability of our CEs by replicating the machine-teaching human experiment of CVE, adjusted to accommodate our graph-based explanations. We design the same stages (pre-learning, learning, and testing) and equally divide our annotators into two independent learning stage variants, namely 'visually-informed' and 'blind'. The 'blind' variant is the only different setting from CVE: the annotators of the 'blind' learning stage are only provided with scene graph pairs and graph edits but no images. This evaluation method is being used for the first time to measure the reliance of humans on graph concepts rather than visual cues to understand the reasoning for classification. More information about human evaluation is provided in App. A.

**Experimental settings and objectives**   Our presented results involve $p \sim N/2$ training graph pairs and the GCN variant unless mentioned otherwise. We produce graph representations using a single Tesla K80 GPU, while all other computations are done on a 12-core Intel Core i7-5930K CPU. We utilize PyG (Fey & Lenssen, 2019) for the implementation of GNNs and DGL (Wang et al., 2019) for approximate GED label calculation. Comparison with CVE showcases the abilities of our model-agnostic method compared to theirs, which requires white-box model access and relies on pixel-level edits. On the other hand, comparison with SC demonstrates the power of graph representations compared to set-level edits in the black-box conceptual set-

ting. An important clarification is that SC proposes the use of roles only in the corresponding experiments of §4.3, meaning that for §4.1, 4.2 they solely rely on concepts. More details in Appendices C, D, E.

### 4.1. Counterfactuals on CUB

We experiment with Caltech-UCSD Birds (CUB) (Wah et al., 2011), despite its lack of ground truth scene graphs. Nevertheless, they can easily be constructed by leveraging given structured annotations: we create a central node to represent the bird and establish 'has' edges connecting it to its parts. Each part is linked to its respective attributes using edges labeled with the corresponding feature type (color, shape, etc.). To be consistent with CVE, we use ResNet50 (He et al., 2015) as the classifier under explanation.

**Quantitative results**   We examine the agreement between the counterfactuals $I'_{(B)}$ retrieved by each method (CVE, SC, ours) and the ground truth GED. Our approach outperforms CVE on every ranking metric (Tab. 1). As for SC, metrics are only valid for $k = 1$ since it produces a single CE instead of a rank. Therefore, P@1 for SC is 0.02, much lower than ours. In addition, we observe that our approach leads to the *lowest number of overall edits*: In Tab. 2, we can see that our method produces about 1 and 2 fewer edits on average for SC and CVE respectively, strengthening the claim that our CEs correspond to the **minimum number of edits**.

*Table 1.* Comparison of counterfactual retrieval results with ground truth GED rankings on CUB. **Bold** denotes best results.

|  | P@k↑ | | P@k (binary)↑ | | NCDG@k (bin.)↑ | |
|---|---|---|---|---|---|---|
|  | k=1 | k=4 | k=1 | k=4 | k=1 | k=4 |
| CVE | 0.02 | 0.10 | 0.02 | 0.11 | 0.11 | 0.26 |
| Ours | **0.19** | **0.34** | **0.19** | **0.49** | **0.23** | **0.36** |

*Table 2.* Average number of node, edge & total edits on CUB. **Bold** for best results (lowest number of edits).

|  | Node ↓ | Edge ↓ | Total ↓ |
|---|---|---|---|
| CVE | 8.43 | 4.70 | 13.13 |
| SC | 8.07 | **3.66** | 11.73 |
| Ours | **6.16** | 4.34 | **10.5** |

Additionally, our CEs are tied to **minimum-cost edits**; more specifically, the resulting GED between the query and the retrieved counterfactual scene graph obtain lower GED scores in comparison to both CVE and SC. The related analysis is presented in App. D.2.

**Qualitative results**   for CUB are presented in Fig. 3 for three images of class A (Rusty Blackbird), accompanied by
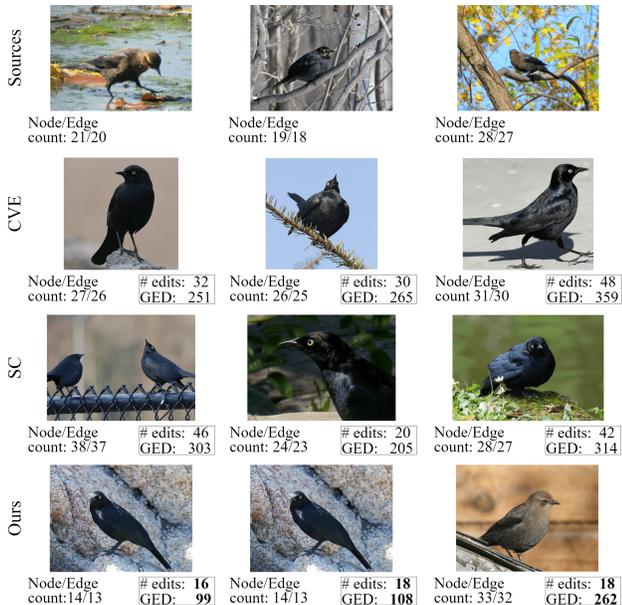
*Figure 3.* Results for Rusty → Brewer Blackbird. **Bold** denotes best results (lowest number of edits and GED scores).

the number of edits and GED needed to transition to class B (Brewer Blackbird). Overall, our approach produces the **fewest concept edits**. SC leads to clear fallacies like suggesting CEs with additional birds (SC, left), or with a portion of the bird in view (SC, middle); thus leading to unnecessary costly deletions and additions. In contrast, our approach mitigates such errors via graphs, where concept instances are uniquely tied to nodes, and their interconnections strongly guide graph similarity through GED, ultimately producing a more accurate and expressive notion of distance than flat unstructured sets. CVE generally fails in finding CEs conceptually similar to the query $I_{(A)}$, as highlighted by the elevated GED and number of edits. Their approach avoids SC's mistakes to an extent by implicitly taking visual features like zoom into account. However, it offers no semantic guarantees, unlike our GED-based approach.

**Human evaluation** Analyzing the results from the comparative human survey (Tab. 3), we deduce that our CEs are **more human-interpretable** than both SC and CVE by a landslide: annotators prefer ours at nearly twice the rate of the CVE alternative. Compared to SC, despite the increased amount of undecided annotators, our CEs were preferred 2.6 times more frequently. This proves that despite the closeness of the two conceptual methods, ours is more intuitive to humans, confirming the meaningful addition of linking concepts within a graph. A chi-square test revealed significant differences in user preferences between our method and SC (p = 0.003) as well as our method and CVE (p = 9.21e-08), indicating a notable deviation from the expected distribution and further validating the reported results.

As for the machine teaching experiment, we obtain the test set accuracy scores (Tab. 4), as the ratio of correctly human-classified test images over the total number of test images. Our visually-informed accuracy clearly outperforms reported CVE scores, highlighting that concept-based CEs are more powerful in guiding humans towards understanding discriminative concepts between classes compared to non-conceptual pixel-level CEs. The "blind" results show an expected decrease compared to the visually-informed ones, but still outperform CVE. The higher accuracy of concept-based over visual CEs affirms the significance humans place on higher-level features for classification. Details regarding human evaluation are presented in App. A.

*Table 3.* Human preference; Win%=% times our method was preferred, Lose% for vice-versa, Tie% when equally preferred. **Bold** denotes higher human preference per method.

| Ours | Win% | Lose% | Tie% |
|---|---|---|---|
| SC | **48.86** | 19.32 | 31.82 |
| CVE | **48.42** | 26.27 | 25.31 |

*Table 4.* Human test accuracy scores for correct classification of samples in classes A and B. **Bold** for top accuracy score.

| Human experiment | Test accuracy %↑ |
|---|---|
| Ours (visually-informed) | **93.88** |
| Ours (blind) | 89.28 |
| CVE | 82.1 |

**Actionability concerns** CVE may lead to non-actionable CEs, despite training on visual semantic preservation. To elaborate, we observe the following: CVE suggests that only adding a striped pattern in a Gray Catbird's wing is adequate to classify it as a Mockingbird. However, by exhaustively generating all annotated attribute combinations of this new bird instance, we easily find several occurring attribute pairs that are not representative of the Mockingbird class; namely, no other Mockingbird has an eyering head pattern and grey breast color. Actionability dictates the prescription of attainable goals achieved through CEs that accurately represent the underlying data distribution (Poyiadzi et al., 2020). To this end, our approach not only selects CEs drawn from the existing target class distribution but also considers all edits needed to convert query to counterfactual image. Therefore, through GED we formalize a more holistic approach to distance and path between counterfactual pairs and simultaneously leverage relations between depicted objects, both visual (relations on the image) and semantic (relations mapped to WordNet synsets). Further analysis in App. E.3.

**Global counterfactuals** in terms of *graph edits* require a standardized unit to be changed, in our case referring either to graph triples in a *(concept-edge-concept)* format or merely to *concept edits* as parts of graph triples. Both approaches regard the aggregation of local edits to explain the given classifier from a higher-level perspective. In the case of CUB, global CEs highly correlate with human perception: by considering the Parakeet Auklet → Least Auklet class transition, some key characteristics of the source class (such as the *('beak', 'shape', 'specialized')* triplet) need to be deleted, while others (such as the *('beak', 'shape', 'cone')* triplet) should be added. Further details in App. D.4.

## 4.2. Towards conceptual counterfactuals

We focus our analysis on conceptual counterfactuals since the previous sections exhibited the indisputable merits of such approaches against the SotA pixel-level method of CVE. In the interest of experimenting on a less controlled dataset, we employ Visual Genome (VG) (Krishna et al., 2017), a dataset containing over 108k human-annotated scene graphs, describing scenes of multiple objects and their in-between interactions. We construct two manageable subsets of 500 scene graphs each, corresponding to ∼125k possible training graph pairs for our GNNs. The first subset denoted as VG-RANDOM is randomly selected, while the second one, named VG-DENSE, is chosen to favor higher graph densities and less isolated nodes to highlight the importance of object interconnections. Details are provided in App. B. VG instances lack ground truth classification labels, allowing us to test our counterfactual retrieval method without the definition of a certain target class. We classify instances using a pre-trained Places365 classifier (Zhou et al., 2017), and regard as counterfactual classes the closest ones in rank. Specifically, we employ a pre-trained ResNet50 (He et al., 2015), as proposed in the original Places365 paper.

**Quantitative Results** We first compare the average number of edits for our method and SC (Tab. 5). Initially, numerical results between the two methods seem similar, but upon closer inspection in conjunction with average GED results of Tab. 6, our method's superiority is evident. VG contains concepts which are much more diverse than CUB, and despite the knowledge-based contraints we enforced during GED computation, edit distance is expected to be higher *between concepts*. Note that this is not true for mean GED since CUB has a higher number of average edits. To this end, our method leads to lower GED in all cases, even when number of edits is higher (VG-RANDOM). Some extra analysis is provided in App. D.2.

Regarding CE approximation to ground truth GED, results for our approach are presented in Tab. 7 denoted as GCN-70K. As for SC, P@1 is 0.25 on VG-DENSE and 0.20 on VG-RANDOM, compared to 0.25 and 0.21 retrieved by

*Table 5.* Average number of node, edge & total edits on VG. **Bold** denotes best results (lowest number of edits).

|      | VG-DENSE | | | VG-RANDOM | | |
|------|--------|--------|--------|--------|--------|--------|
|      | Node↓ | Edge↓ | Total↓ | Node↓ | Edge↓ | Total↓ |
| SC   | **4.91** | 7.29 | 12.2 | **12.15** | 7.52 | **19.67** |
| Ours | 4.95 | **7.15** | **12.11** | 12.18 | 7.54 | 19.72 |

*Table 6.* Average top-1 GED (VG) for CEs when methods disagree. **Bold** for best (lowest) GED scores for each dataset split.

|      | VG-DENSE ↓ | VG-RANDOM ↓ |
|------|-----------|-------------|
| SC   | 128.67 | 186.77 |
| Ours | **122.41** | **180.67** |

our method. GED approximation is satisfactory but close between methods due to a general agreement in CE retrieval. Despite this fact, our approach still leads in all reported metrics, especially for VG-DENSE, displaying superiority in cases of disagreement. Ranking results consistent with our analysis are also obtained using GQA (Hudson & Manning, 2019), a VG variant which focuses on question-answering, as reported in App. D.3. Reported findings place great significance in examining qualitative results.

*Table 7.* Ranking results on the two VG variants for various GNNs. **Bold** numbers indicate best ranking metrics.

| Models | P@k ↑ | | P@k (binary)↑ | | NDCG@k (binary)↑ | |
|--------|------|------|------|------|------|------|
|        | k=1 | k=4 | k=1 | k=4 | k=1 | k=4 |
| | VG-DENSE | | | | | |
| Kernel | 0.13 | 0.17 | 0.13 | 0.26 | 0.19 | 0.33 |
| GIN-70K | 0.16 | 0.27 | 0.16 | 0.38 | 0.20 | 0.34 |
| GAT-70K | 0.18 | 0.32 | 0.18 | 0.44 | 0.22 | 0.35 |
| GCN-70K | **0.25** | **0.37** | **0.25** | **0.49** | **0.28** | **0.41** |
| | VG-RANDOM | | | | | |
| Kernel | 0 | 0.01 | 0 | 0.01 | 0.10 | 0.25 |
| GIN-70K | 0.03 | 0.07 | 0.03 | 0.07 | 0.22 | 0.38 |
| GAT-70K | 0.18 | 0.29 | 0.18 | 0.38 | 0.11 | 0.27 |
| GCN-70K | **0.21** | **0.30** | **0.21** | **0.42** | **0.25** | **0.38** |

**Qualitative results** By examining counterfactual images retrieved for VG-DENSE in Fig. 4 (left), there is a clear indication that by considering the complex relations between concepts, our method achieves more **detail-oriented results**: in the 1st column, our approach not only retrieves an image with 'man', 'board', 'water' concepts, but also the relation 'man on board'. In the 3rd column, we consider the relation of toppings and retrieve the pizza, while SC simply retrieves an image with similar concepts, ('bun' and 'bread' or 'meat'
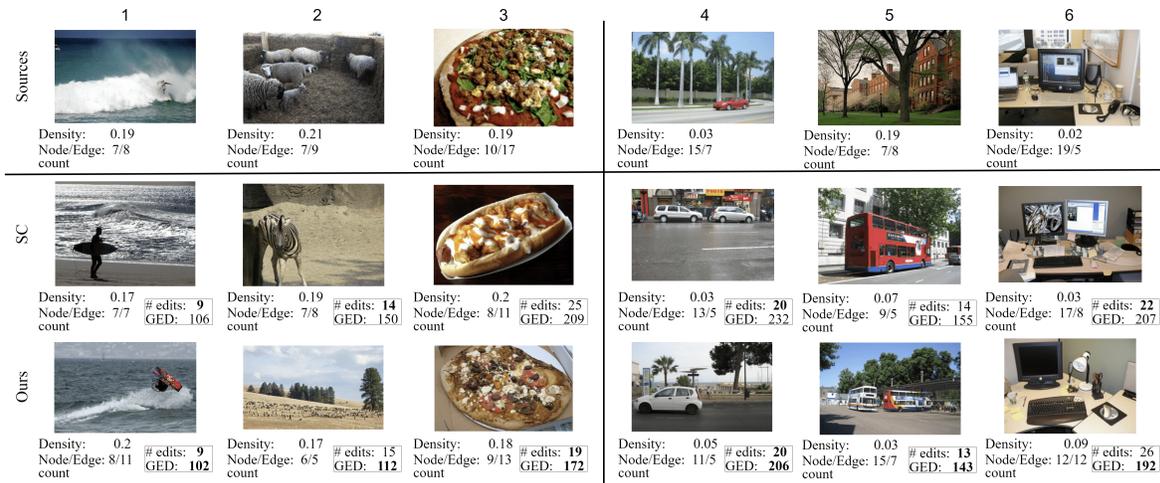
*Figure 4.* Qualitative results (best metrics in **bold**): VG-DENSE (left 3 columns) and VG-RANDOM (right 3 columns).

and 'sausage'). Results on VG-RANDOM (Fig. 4 (right)) follow the same logic. In columns 4-5, our method retrieves the focal points of the images since it regards relations between trees and other objects. Taking into account the sparsity of the underlying graphs, however, in some cases the importance of concepts trumps the underlying structure, as in the 6th column. This fact is reflected in the elevated number of edits of our method for VG-RANDOM, yet it is not true for GED, showcasing once again the importance of semantic context. More details in App. E.2.

**Why GCN?** Ranking metrics of GNN models are provided in Tab. 7. Three GNN variants (GAT, GIN, GCN) are trained using $p = N/2 = 70$k scene graph pairs. The GCN-based variant consistently approaches GED the closest, with a binary P@4 of 49% and P@1 of 24.80% for VG-DENSE and slightly worse results on VG-RANDOM. GCN systematically scores higher in comparison to theoretically more competent GNN alternatives, such as GIN. We attribute this finding to the importance of local neighborhood information for our small yet semantically dense graphs. Specifically, the VG graphs considered in our experiments rarely exceed 3-4 hops, as briefly demonstrated in App. E.2. GIN does not incorporate node features during aggregation resulting in a limited notion of semantic similarity. This ablation study affirms using GCN for the GNN-based similarity component of our approach. GNNs can also outperform other prominent deterministic methods, like graph kernels (Grauman & Darrell, 2007). The reported findings grant us the security that our counterfactual explanations are **trustworthy**, even when applied to complex scene graphs.

### 4.3. Extendability of graph-based counterfactuals

The flexibility of our approach is proven under two scenarios: a) its application on unannotated images, b) its expan-

sion into other modalities. For direct comparison to SC we provide global CEs by averaging overall graph triple edits.

**Unannotated datasets** We replicate Dervakos et al. (2023)'s experiment on explaining the classification of web-crawled creative-commons images into 'driver' and 'pedestrian' classes. Here, images were manually classified by the authors; thus, we explain a non-neural classifier. By employing the SotA scene graph generator (SGG) of Cong et al. (2023) we extract global edits from generated graphs for the transition from 'pedestrian' to 'driver' (Fig. 5(left)). Their relevance is verified by our common sense: people wear helmets when driving -addition of (helmet, on, head) and (man, on, bike)- and cover the bike seat with their body -deletion of (seat, on, bike)-. To validate our method's consistency across other annotation techniques, we replace the SGG with a pipeline of captioning (BLIP (Li et al., 2022)) and graph parsing (Unified VSE (Wu et al., 2019)). We confirm that resulting edits (Fig. 5 (right)) semantically resemble the ones in Fig. 5 (left). Overall, similarly to Fig. 1, more accurate local edits are achieved through the consideration of the multiplicity of objects and relations. Generic triple edits result from errors in the automatic annotation pipeline, emphasizing the importance of meticulous explanation dataset curation. We provide further details regarding the 'pedestrian' vs 'driver' classification experiment in App. F. Additionally, we experiment with more unannotated datasets of scene images, such as Action Genome (Ji et al., 2020), which is presented in App. F.

**Audio classification** Despite focusing on images, we briefly demonstrate our method's model-agnostic nature by applying it for audio classification, following SC. We provide CEs using the Smarty4covid dataset (Zarkogianni et al., 2023) for the IEEE COVID-19 sensor informatics competi-
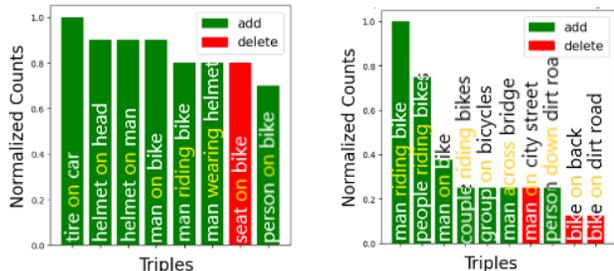
*Figure 5.* Graph edits (triples inserted/ deleted) to implement the 'pedestrian' → 'driver' transition. The yellow color distinguishes edge from node labels within a triple.

tion winner[2], which predicts COVID-19 from cough audio. Our findings align with SC, revealing the high frequency of concept edits among respiratory symptoms and uncovering the same gender bias. No new findings are produced for this primarily concept-based dataset with trivial interconnections, once again placing the focus on the nature and density of the annotations. However, we confirm that our method is at least as good as SC in these cases nonetheless. More results regarding the audio classification experiment are discussed in App. G.

### 4.4. Efficiency of graph-based counterfactuals

**Time Performance for Counterfactual Retrieval**    From a theoretical standpoint, our method's efficiency is expected. The heaviest part of GNN computations occurs during model training, where each backward call is correlated with the square of the number of nodes in addition to the number of edges. Inference, on the other hand, is nearly instantaneous and is linearly correlated with the sum of the number of nodes and edges in a graph (Blakely et al., 2019).

Additionally, we experimentally confirm that our method allows for **efficient** CE retrieval: In Tab. 8, we report execution times for CE computation on the complete sets of graphs using GED (Fankhauser et al., 2011) versus our GNN-powered approach. We further report retrieval and inference time of our method. Even by adding times for all GCN-N/2 operations, we significantly relieve the computational burden of calculating the ground truth GED for all graph pairs, especially for larger graphs.

**Performance-complexity trade-off**    In Fig. 6, we examine how retrieval precision varies using different numbers of training pairs $p$ on CUB (Fig. 6b) and the two VG variants (Fig. 6a). P@k does not exhibit significant increase in any case after the $\sim N/2$ pairs mark (70k for VG and 50k for CUB). On the contrary, it could remain identical (Fig. 6a left) or even decrease (Fig. 6b). The same precision pattern

*Table 8.* Time (sec) for counterfactual calculation. Training time is reported due to the transductivity of the GNN method.

| | GED ↓ | GCN-N/2 (train)↓ | GCN-N/2 (retr.)↓ | GCN-N/2 (infer.)↓ |
|---|---|---|---|---|
| CUB | 46220 | 32691 | 0.03 | 0.06 |
| VG-DENSE | 13982 | 12059 | 0.03 | 0.06 |
| VG-RANDOM | 18787 | 16271 | 0.03 | 0.10 |

per $p$ is experimentally validated on the GQA dataset (Appendix D.3). The consistency of behavior exhibited over $\sim N/2$ pairs concludes our claim that $N/2$ **training pairs are adequate** for appropriate graph embedding using GCN.
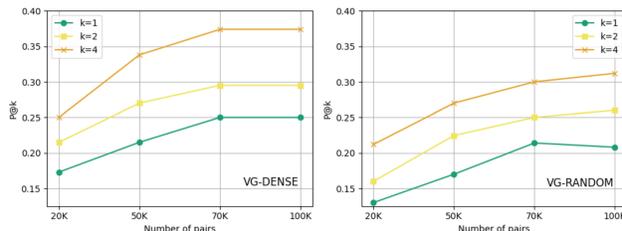


(a) VG variants
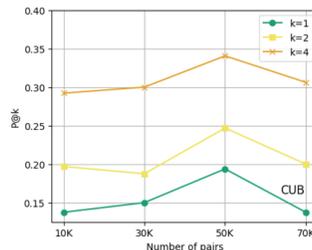


(b) CUB

*Figure 6.* P@k of GCN variant for different training pairs $p$ on the two main datasets explored.

## 5. Conclusion

In this paper, we proposed a new model-agnostic approach for counterfactual computation based on the expressive power of semantic graphs. To this end, we suggested counterfactual retrieval by GED calculation, employing a GNN-based similarity model to accelerate the otherwise NP-hard retrieval process between all input graph pairs. Comparison with previous CE models proved that our explanations correspond to minimal edits and are more human interpretable, especially when interactions between concepts are dense, while still ensuring actionability. We further confirmed the applicability of our framework on datasets without annotations. There is ample room for future work, including exploring potential limitations, like robustness and the impact of low quality annotations, as well as further improving the efficiency by employing unsupervised GNN methods.

## Acknowledgments

## Impact Statement

This paper presents work whose goal is to advance the field of Explainability in Machine Learning. There are no societal consequences of our work concerning the utilization of data such as image datasets or annotations. However, end users should exercise caution when relying on explanation methods, not specifically ours but generally, especially in sensitive domains. This caution stems from the possible presence of low-quality data in the real-world environment, a factor beyond the focus of our research. Users of this system must carefully select their data, as is prudent in any AI application.

## References

Abid, A., Yuksekgonul, M., and Zou, J. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning*, pp. 66–88. PMLR, 2022.

Augustin, M., Boreiko, V., Croce, F., and Hein, M. Diffusion visual counterfactual explanations. *arXiv preprint arXiv:2210.11841*, 2022.

Bai, Y., Ding, H., Qiao, Y., Marinovic, A., Gu, K., Chen, T., Sun, Y., and Wang, W. Unsupervised inductive graph-level representation learning via graph-graph proximity. *arXiv preprint arXiv:1904.01098*, 2019.

Bajaj, M., Chu, L., Xue, Z. Y., Pei, J., Wang, L., Lam, P. C.-H., and Zhang, Y. Robust counterfactual explanations on graph neural networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 5644–5655. Curran Associates, Inc., 2021. URL https://proceedings. neurips.cc/paper/2021/file/ 2c8c3a57383c63caef6724343eb62257-Paper. pdf.

Blakely, D., Lanchantin, J., and Qi, Y. Time and space complexity of graph convolutional networks. *GitHub pages*, 2019.

Browne, K. and Swift, B. Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks. *arXiv preprint arXiv:2012.10076*, 2020.

Cong, Y., Yang, M. Y., and Rosenhahn, B. Reltr: Relation transformer for scene graph generation, 2023.

Deccan Chronicle. Cyclists without helmet, 2016. URL https://www.deccanchronicle. com/nation/current-affairs/030316/ motorists-scramble-for-driving-licence. html. [Online; accessed February 1, 2024].

Dervakos, E., Thomas, K., Filandrianos, G., and Stamou, G. Choose your data wisely: A framework for semantic counterfactuals. In Elkind, E. (ed.), *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pp. 382–390. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/43. URL https: //doi.org/10.24963/ijcai.2023/43. Main Track.

Dimitriou, A., Chaidos, N., Lymperaiou, M., and Stamou, G. Graph edits for counterfactual explanations: A comparative study, 2024.

Fankhauser, S., Riesen, K., and Bunke, H. Speeding up graph edit distance computation through fast bipartite matching. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pp. 102–111. Springer, 2011.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Filandrianos, G., Thomas, K., Dervakos, E., and Stamou, G. Conceptual edits as counterfactual explanations. In *AAAI Spring Symposium: MAKE*, 2022.

Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., and Lee, S. Counterfactual visual explanations. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2376–2384, 2019.

Grauman, K. and Darrell, T. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8(4), 2007.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Hou, W. and Ji, Z. Gpt-4v exhibits human-like performance in biomedical image classification. *bioRxiv*, pp. 2023–12, 2024.

Hudson, D. A. and Manning, C. D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Ji, J., Krishna, R., Fei-Fei, L., and Niebles, J. C. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10236–10247, 2020.

Jonker, R. and Volgenant, A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

Li, Y., Gu, C., Dullien, T., Vinyals, O., and Kohli, P. Graph matching networks for learning the similarity of graph structured objects. In *International conference on machine learning*, pp. 3835–3845. PMLR, 2019.

Lucic, A., Ter Hoeve, M. A., Tolomei, G., De Rijke, M., and Silvestri, F. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 4499–4511. PMLR, 2022.

Miller, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Nikolentzos, G., Meladianos, P., and Vazirgiannis, M. Matching node embeddings for graph similarity. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

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

Prado-Romero, M. A., Prenkaj, B., Stilo, G., and Giannotti, F. A survey on graph counterfactual explanations: definitions, methods, evaluation, and research challenges. *ACM Computing Surveys*, 56(7):1–37, 2024.

Ranjan, R., Grover, S., Medya, S., Chakaravarthy, V., Sabharwal, Y., and Ranu, S. Greed: A neural framework for learning graph distance functions. In *Advances in Neural Information Processing Systems*, 2022.

Ribeiro, M. T., Singh, S., and Guestrin, C. "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*, pp. 1135–1144, 2016.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Sanfeliu, A. and Fu, K.-S. A distance measure between attributed relational graphs for pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (3):353–362, 1983.

Siglidis, G., Nikolentzos, G., Limnios, S., Giatsidis, C., Skianis, K., and Vazirgiannis, M. Grakel: A graph kernel library in python. *J. Mach. Learn. Res.*, 21(54):1–5, 2020.

Slack, D., Hilgard, A., Lakkaraju, H., and Singh, S. Counterfactual explanations can be manipulated. *Advances in neural information processing systems*, 34:62–75, 2021.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Vandenhende, S., Mahajan, D., Radenovic, F., and Ghadiyaram, D. Making heads or tails: Towards semantically consistent visual counterfactuals. *arXiv preprint arXiv:2203.12892*, 2022.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

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

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., and Zhang, Z. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Williams, C. On a connection between kernel pca and metric multidimensional scaling. *Advances in neural information processing systems*, 13, 2000.

Wu, C., Lei, J., Zheng, Q., Zhao, W., Lin, W., Zhang, X., Zhou, X., Zhao, Z., Zhang, Y., Wang, Y., et al. Can gpt-4v (ision) serve medical applications? case studies on gpt-4v for multimodal medical diagnosis. *arXiv preprint arXiv:2310.09909*, 2023.

Wu, H., Mao, J., Zhang, Y., Jiang, Y., Li, L., Sun, W., and Ma, W.-Y. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6602–6611, 2019. URL https://api.semanticscholar.org/CorpusID:119284150.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Zarkogianni, K., Dervakos, E., Filandrianos, G., Ganitidis, T., Gkatzou, V., Sakagianni, A., Raghavendra, R., Nikias, C., Stamou, G., and Nikita, K. S. The smarty4covid dataset and knowledge base: a framework enabling interpretable analysis of audio signals. *arXiv preprint arXiv:2307.05096*, 2023.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

# A. Human evaluation details

## A.1. Participants and Consent

We distributed an information sheet describing the goals and stages of our human surveys to software engineering students online. We clarified that their participation would be voluntary and without any form of compensation. We additionally distributed the following form to obtain annotators' consent in the form of a checklist. We used the same form both for the machine teaching as well as the counterfactual preference experiment. The 33 people who ultimately participated were young adults of ages 19-25 both male and female, without any knowledge of bird species.

I confirm that I have read and understand the information sheet for the above research. I have had the opportunity to consider the information, ask questions and have had these answered satisfactorily.

I understand that my participation is voluntary without compensation and that I am free to withdraw at any point, without giving any reason.

I understand who will have access to personal data provided, how the data will be stored and what will happen to the data at the end of the project.

I understand that I will not be identifiable from any publications or organisations.

I agree to take part.

*Figure 7.* Screenshot of the consent form for human evaluation. Our annotators fill out this form before they proceed with annotations.

Our human survey was completely anonymous and we did not record any type of personal data from our annotators.

## A.2. 1st experiment: comparative human survey

In Fig. 8, we present a screenshot of the platform we provided to our evaluators for the comparative user survey. Users are asked to select a sample to annotate, as shown in the panel of Fig. 9. We ensured that our evaluators can clearly view the images and their details by providing 'zoom-in'/'zoom-out' tools, as well as the ability to navigate within the image with the 'pan' and 'move' options.

An annotator can click on any sample to be annotated, thus moving to a screen such as the one of Figure 9. The source image is presented on the left, and the two alternative options (ours versus a counterfactual image of CVE (Vandenhende et al., 2022) or SC (Dervakos et al., 2023)) are placed in the middle and the rightmost column. These options are shuffled in each sample, so that no bias towards each choice is created. Only one of the options ("Image 1", "Image 2" or "Can't tell") can be selected for each sample.

In this first human experiment, our annotators can evaluate as many samples as they wish; however, they cannot update an existing annotation. All 33 annotators participated in this experiment.

## A.3. 2nd experiment: machine-teaching human survey

We once again employ the same platform as for the previous human experiment. However, this time each annotator can only evaluate **one** single sample; we enforce this restriction to clearly evaluate the contribution of the learning phase, excluding situations that an annotator could have become more 'competent' after passing many times through the learning phase.

The experimental workflow is adopted from (Vandenhende et al., 2022), therefore we include all the three stages (pre-learning, learning and testing).

**Pre-learning stage**    In the pre-learning stage, users are presented with unlabeled images from the test set to get familiarized with the nature of the images they will be tasked to classify later on. Fig. 10 is provided as an example of the pre-learning screen. The annotators become aware that the classification to the anonymized classes A and B cannot be performed without

*Figure 8.* Screenshot of the platform provided for human evaluation.



*Figure 9.* Annotation panel with instructions and image navigation tools provided to the evaluators for CUB.

passing through the learning stage, therefore selecting "I don't know" is the expected option. In Fig. 10, we can explicitly see the three options for image classification, namely "Class A", "Class B" or "I don't know". Only one can be selected at a time, as in (Vandenhende et al., 2022).

**Learning stage** The learning stage comprises the heart of this human experiment. As mentioned in the main paper, we perform two variants of it to measure the degree of reliance on concepts, according to human perception. A user can either participate in the "visually-informed" or the "blind" experiment, but not both. This is necessary so that we exclude the possibility of evaluating the same data sample in each of the experiments and thus eliminate the possibility of having some knowledge transfer across the two variants of this experiment. Annotators are divided into equal subgroups (17 in the "visually-informed" variant and 16 in the "blind" one).

In the **visually-informed** variant, annotators are presented with training images from anonymized classes A and B, together with their scene graphs, as shown in Figures 11, 12, 13. Of course, training and test images do not overlap. Annotators are again provided with 'zoom-in'/'zoom-out', 'pan', 'move' tools, etc. to navigate within the images and the accompanying scene graphs.
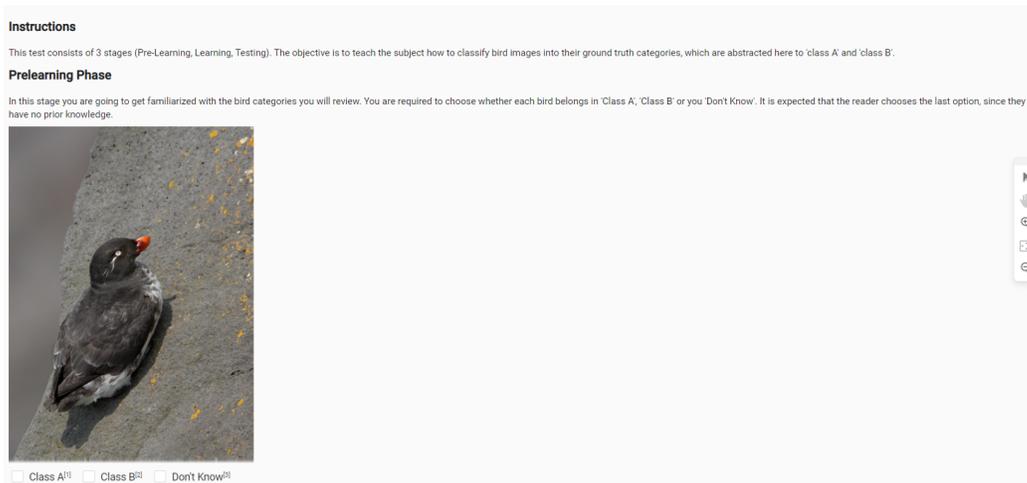
*Figure 10.* Pre-Learning stage instructions for CUB machine teaching experiment. Choices are "Class A", "Class B" or "I don't know".

Training images on the left always belong to class A, while images on the right always belong to class B. Scene graphs on the right also contain the edits needed to perform the $A \rightarrow B$ transition, with green nodes representing concept additions, blue nodes indicating concept substitutions (both source and target concepts of the substitution are shown), and red nodes denoting concept deletions. The rest of the nodes imply that the corresponding concepts remain the same between the two classes.

A user implicitly focuses on the most frequent insertions, substitutions, and deletions performed throughout the training stage to understand the discriminative features between class A and class B. Associating such concepts with the images helps mapping graph edits to visual differences so that the user learns to separate classes visually and conceptually.

In the **blind** variant of the learning stage, only scene graphs are provided, but no training images. Also, the graph edits are presented to the users via colored nodes. This learning variant is a direct analogy to the machine-teaching learning stage implemented by Vandenhende et al. (2022): in their case, pixels corresponding to discriminative regions that act as explanations are provided, while the rest of the bird image is blurred out. Therefore, annotators need to learn solely from the explanation and mentally connect the corresponding concepts to existing visual regions of the testing images. In our case, the derived explanations correspond to graph edits, therefore annotators have to learn the discriminative concepts that are added, substituted, or deleted to perform the $A \rightarrow B$ transition. However, since our learning setting is performed without any visual clue, we regard our blind learning stage as being **more difficult** than the learning stage that (Vandenhende et al., 2022) implement; our annotators have to connect concepts with image regions, thus performing cross-modal grounding in order to learn discriminative features.

Throughout the blind learning stage, we are able to measure the reliance on concepts rather than pixels to learn to classify images of unknown classes. This experiment is important in order to highlight how meaningful and informative conceptual explanations are to humans, so that they can approximate a zero-shot classification setting.

**Testing stage**    In the testing stage, users are provided with the same images as in the pre-learning stage. No scene graphs are provided. Based on the previous stage, annotators should have learned visual and conceptual differences between classes; therefore, they are tasked to assign an appropriate class to each test image, by selecting either "class A" or "class B" for each of them. Contrary to the pre-learning stage, the option "I don't know" is not provided.

After this stage, an accuracy score is extracted per user, based on their correct selections in the testing stage. We then extract an average accuracy per user, which we report in the main paper. Our average accuracy for the visually-informed experiment is 93.88%, indicating that in most cases users are highly capable of recognizing the key concepts that separate the two given bird classes, grounding them with visual information. As for the blind experiment, the average testing accuracy is 89.28%. Being rather close to the visually-informed accuracy percentage, we can safely assume that **concepts are more than adequate** towards teaching discriminative characteristics to humans, even if they lack association with purely visual information. Both visually-informed and blind accuracy scores clearly outperform the accuracy scores reported in CVE,
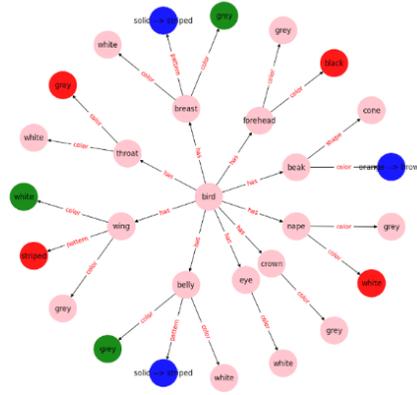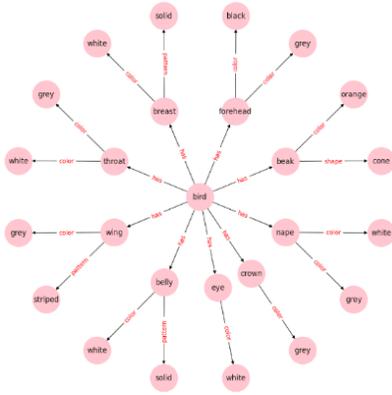
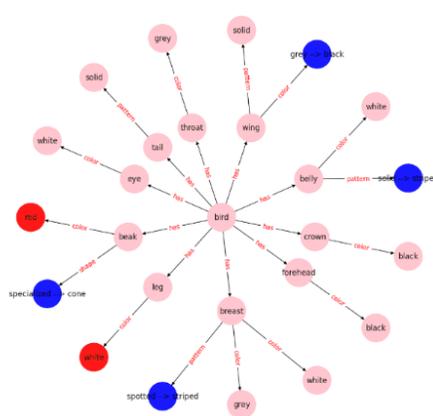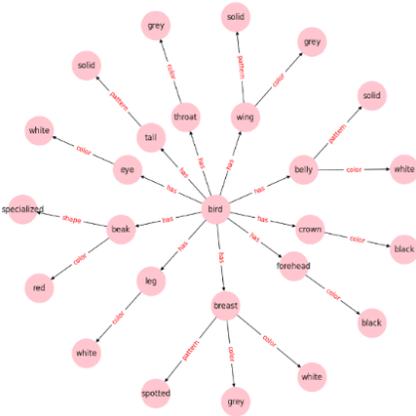*Figure 11.* Example of the visually-informed learning stage.



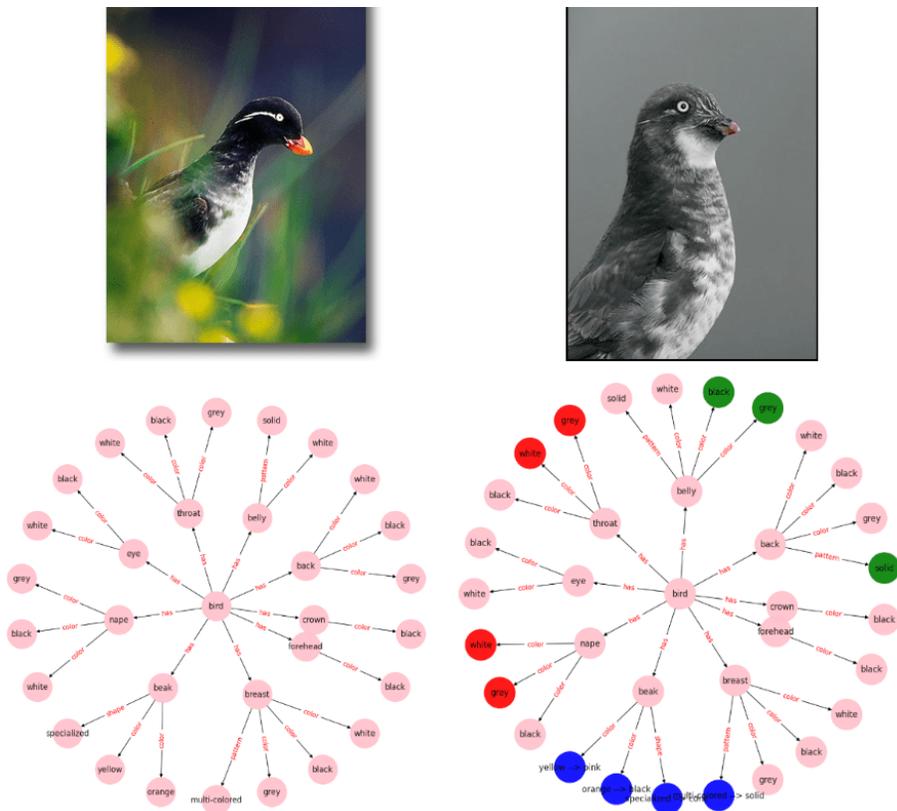*Figure 12.* Example of the visually-informed learning stage.

*Figure 13.* Example of the visually-informed learning stage.

demonstrating that conceptual explanations are more **meaningful** and **informative** to humans compared to pixel-level explanations.

**Accuracy score distribution**    In Fig. 14, we present a more detailed analysis of the accuracy scores achieved by human subjects during the testing phase of the machine teaching experiment. It is apparent that scores peak at 0.9 and 1.0; thus, explanations produced by our method are highly human-interpretable and beneficial to perform classification. Comparison between 'visually-informed' and 'blind' results reveals that the decrease in test accuracy for the experiment without a visual aid is gradual.

**Applicability of machine-teaching experiment**    The machine-teaching experiment is purposely run exclusively on the CUB dataset. To highlight the merits of the learning phase: annotators have no knowledge of bird species, therefore they can highly benefit from learning discriminative bird attributes, and then apply this new knowledge in the testing phase. For example, none of the annotators knows the difference between a Parakeet Auklet and a Least Auklet. Nevertheless, after the learning stage, they are able to recognize the basic discriminative attributes, which will help them classify instances of



*Figure 14.* Distribution of test accuracy for machine teaching human evaluation experiments.

the test phase. On the other hand, Visual Genome contains images of common everyday scenes, rendering a similar experiment rather redundant in such instances. For example, a human already knows key concepts that discriminate a kitchen from a bedroom, therefore the learning stage would be of no value, even if the scene labels are anonymized. We can view this scenario as an analog to data leakage.
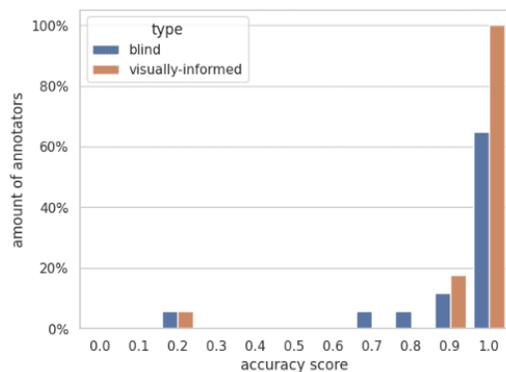
Moreover, there is always a possibility that some concepts can be misleading. In such cases, we expect visual classifiers to present a bias towards such concepts, while this is not the case for humans. For example, a TV can be present in both kitchens and bedrooms. However, in a hypothetical scenario that selected bedroom images have TVs, but kitchen images do not, the graphs that serve as explanations would contain many "add TV" nodes. Therefore, a human expects to classify images containing TVs in one class, and images that do not contain TVs in the other (as a visual classifier would do if trained on such data). But when finally humans are presented with real test images, they will not be misled by the presence or the absence of TVs, but rather rely on their commonsense to perform classification. Thus, not only is the learning stage redundant, but the obvious existing bias "add TV" is not reflected in the final classification; in this case, the counterfactual explanation itself would be of no value to humans.

## B. Graph statistics

*Table 9.* Statistics regarding graphs of different datasets used in the main paper.

|  |  | VG-DENSE | VG-RANDOM | CUB | D/P-SGG | D/P-CAPTION | SMARTY |
|---|---|---|---|---|---|---|---|
| Mean | density | 0.20 | 0.06 | 0.04 | 0.13 | 0.25 | 0.23 |
|  | edges | 9.04 | 8.77 | 27.52 | 9.37 | 1.76 | 4.40 |
|  | nodes | 7.25 | 14.57 | 28.52 | 9.73 | 3.20 | 5.40 |
|  | isolated nodes | 0.47 | 3.37 | 0 | 0.32 | 0.90 | 0 |
| Max | density | 0.47 | 0.67 | 0.11 | 1.0 | 0.5 | 0.33 |
|  | edges | 36 | 27 | 53 | 18 | 4 | 15 |
|  | nodes | 15 | 20 | 54 | 18 | 5 | 16 |
|  | isolated nodes | 3 | 12 | 0 | 3 | 4 | 0 |
| Min | density | 0.14 | 0.01 | 0.02 | 0.05 | 0.05 | 0.06 |
|  | edges | 5 | 5 | 8 | 1 | 1 | 2 |
|  | nodes | 6 | 4 | 9 | 2 | 2 | 3 |
|  | isolated nodes | 0 | 0 | 0 | 0 | 0 | 0 |

In Tab. 9 we present some additional statistics regarding the graphs of the datasets used in our work (max and min details). VG-DENSE and VG-RANDOM contain 500 graphs each, CUB contains 422 graphs, D/P-SGG and D/P-CAPTION denote the web-crawled datasets with 259 graphs each and SMARTY denotes the COVID-19 classification dataset with 548 graphs. Table 10 contains additional statistics about datasets utilized only in the appendix. These are GQA (Hudson & Manning, 2019) with 500 graphs mentioned in App. refsec:quant-additional and Action Genome (AG) (Ji et al., 2020) with 300 graphs mention in App. refsec:unannotated. The size and density of input data should be considered when viewing results in the experimental section.

*Table 10.* Statistics regarding graphs of different datasets in the appendix.

|  |  | AG | GQA |
|---|---|---|---|
| Mean | density | 0.19 | 0.24 |
|  | edges | 13.23 | 8.14 |
|  | nodes | 8.84 | 6.66 |
|  | isolated nodes | 1.17 | 1.37 |
| Max | density | 0.45 | 1.0 |
|  | edges | 51 | 20 |
|  | nodes | 17 | 12 |
|  | isolated nodes | 2 | 15 |
| Min | density | 0.1 | 0.13 |
|  | edges | 4 | 5 |
|  | nodes | 5 | 4 |
|  | isolated nodes | 0 | 0 |

## C. Experimental Settings

In addition to details regarding resources used for the experimental setup mentioned in the main paper, we further report specific training configurations for GNN models. All presented results were achieved using single-layer GNNs of a dimension of 2048, built as explained in Sec. 3 of the main paper. For reproducibility purposes, we report that these models were optimized for a batch size of 32 and trained for 50 epochs, without the use of dropout. The employed optimizer was Adam without weight decay. The respective learning rate varied among GNN variants. To be precise, we used a learning

rate of 0.04 for GCN and 0.02 for GAT and GIN. GAT and GIN also have model-specific hyperparameters - attention heads and the learnable parameter epsilon respectively. Best results were achieved by leveraging 8 attention heads and setting epsilon to non-learnable.

Last but not least, an important hyperparameter of the GNN models is the number of training pairs, denoted as $p$. As explained, optimal models used $p \approx N/2$, which varies among datasets. Specifically, the parameter $p$ is set to 70K for datasets with 500 graphs, 50K for datasets with 422 graphs, and 25K for datasets with 300 graphs. However, we also conducted ablations on the number of training pairs, setting $p$ to values reported in Fig. 5 of the main paper. In those cases, we explored using 16%, 40%, and 80% of the existent graph pairs, in addition to the "golden" 50%.

Regarding graph kernels that were employed for comparison, we report that the Pyramid Match kernel was used with its default settings. The settings include leveraging labels, a histogram level $L$ of 4, and hypercube dimensions $d$ of 6.

The classifier in our CUB experiments (ResNet-50) was chosen in alignment to experiments performed in the works compared and recreated here. As for the choice of the Places365 instead of a pretrained ImageNet classifier, it was conscious. Despite the latte being potentially more widely recognized and researched, it is trained on the ImageNet dataset, which primarily consists of foreground objects. In Visual Genome, the majority of instances depict scenes, providing substantial background. Although some instances focus more on specific objects, they are still situated within a particular environment. In contrast, ImageNet classifiers face challenges with such inputs, as only about 3% of the target classes in the corresponding dataset pertain to broader scenes. Classifiers for the rest of the datasets are explained in detail in the following sections.

The code for all experiments is provided within the zip file of the supplementary material, accompanied by comprehensive instructions.

## D. Quantitative Experiments

### D.1. Graph Kernels

Graph kernels are kernel functions used on graphs that measure similarity in polynomial time, providing an efficient and widely applicable alternative to GED. In the context of this paper, we experimented with several kernels from the GraKeL library (Siglidis et al., 2020), as a baseline measure for counterfactual retrieval. Our goal is to guarantee that our GNN framework outperforms such methods. We present results from the best-performing kernel Pyramid Match.

**Pyramid Match (PM) kernel**   The PM (Grauman & Darrell, 2007; Nikolentzos et al., 2017) graph kernel operates by initially embedding each graph's nodes in a d-dimensional vector space using the absolute eigenvectors of the largest eigenvalues of the adjacency matrix. The sets of graph vertices are compared by mapping the corresponding points in the d-dimensional hypercube to multi-resolution histograms, using a weighted histogram intersection function. The comparison process occurs in several levels, corresponding to different regions of the feature space with increasing size. The algorithm counts new matches at each level - i. e. points in the same region - and weights them according to the size of the level. The cells/regions double in size in each iteration of the algorithm.

This procedure is applicable to graphs with node/edge labels; thus, we cannot leverage GloVe embeddings for initialization. Matches exist only between points with the exact same label. The overall complexity of the algorithm is $O(ndL)$, which compared to other kernel methods is quite computationally expensive.

### D.2. Average GED

In addition to the average number of edits metric and the ranking metrics using the ground truth GED as the golden standard, we present the average GED of the top-1 counterfactual results. This supplementary measure serves to explicitly enhance comprehension of the significance of semantic context. Notably, within the main paper, our qualitative results illustrate scenarios where, despite an equal (or lower) number of edits, the GED can at times be higher. This divergence arises because edits are not uniformly weighted but rather based on their semantic similarity.

For the VG dataset, we present results comparing "Normal" and "Refined" outcomes. In this context, "Refined" denotes presenting averages exclusively when the two methods yield distinct counterfactuals. We adopted this approach due to the observation that 75%

*Table 11.* Average top-1 GED on CUB. **Bold numbers** denote best results.

|      | CUB $\downarrow$ |
| ---- | ------ |
| CVE  | 257.20 |
| SC   | 263.80 |
| Ours | **211.69** |

*Table 12.* Refined average number of node, edge & total edits on VG.

| | VG-DENSE | | | VG-RANDOM | | |
|---|---|---|---|---|---|---|
| | Node↓ | Edge↓ | Total↓ | Node↓ | Edge↓ | Total↓ |
| SC | **4.73** | 7.65 | 12.38 | **11.96** | **7.48** | **19.44** |
| Ours | 5.07 | **6.96** | **12.03** | 12.37 | 7.52 | 19.89 |

of CEs for VG-DENSE and 73% for VG-RANDOM were identical between methods, creating an impression of increased result proximity. To provide a comprehensive view, we furnish more refined average number of edits results in Table 12. Notably, for the CUB dataset, such an analysis is unnecessary; nonetheless, we include the average top-1 GED in Table 11.

### D.3. Additional Datasets

*Table 13.* Ranking results on GQA for different graph models.

| | P@k ↑ | | | NDCG@k ↑ | | | P@k (binary) ↑ | | | NDCG@k (binary) ↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | k=1 | k=2 | k=4 | k=1 | k=2 | k=4 | k=1 | k=2 | k=4 | k=1 | k=2 | k=4 |
| PM | 0.06 | 0.10 | 0.05 | 0.65 | 0.65 | 0.68 | 0.10 | 0.15 | 0.20 | 0.17 | 0.22 | 0.31 |
| GIN-70K | 0.16 | 0.24 | 0.29 | 0.70 | 0.70 | 0.72 | 0.16 | 0.27 | 0.39 | 0.20 | 0.25 | 0.34 |
| GAT-70K | 0.13 | 0.17 | 0.22 | 0.66 | 0.68 | 0.69 | 0.13 | 0.21 | 0.30 | 0.18 | 0.24 | 0.32 |
| GCN-70K | **0.19** | **0.29** | **0.34** | **0.73** | **0.73** | **0.74** | **0.19** | **0.33** | **0.48** | **0.22** | **0.28** | **0.36** |

**Ranking results for GQA** The analysis performed on Visual Genome (VG) is extended on the GQA dataset (Hudson & Manning, 2019). In fact, GQA comprises a variant of VG focusing on compositional question-answering involving real-world scenes. Since GQA images and accompanying scene graphs are very similar to the ones involved in our VG analysis, the obtained results verify the findings reported for VG without offering other novel insights. In Tab. 13 we present per-model results for 70K training pairs. GCN remains the most powerful architecture compared to the other ones, an observation validating the findings reported for the rest of the datasets.

**Performance-complexity trade-off for GQA** In Fig. 15 we present the performance analysis for different numbers of training pairs $p$ on the GQA dataset, focusing on our best-performing model (GCN). Once again, $N/2 \sim 70K$ pairs are adequate for learning proper representations of scene graphs, validating our initial claim that GED does not have to be computed for more than $N/2$ graph pairs to obtain a satisfactory approximation.



*Figure 15.* Comparison of the GCN performance measured in P@k for different number of training pairs $p$ for GQA.

**Ranking results for Action Genome** are presented in Tab. 14, while **number of edits for Action Genome** is presented in Tab. 15, both for $N/2 = 25K$.

### D.4. Global edits on CUB

By aggregating edits from each image participating in the dataset, we can extract *global edits*: they describe what needs to be changed in total to explain the transition from one class to the other. These edits are more meaningful in the form of graph triples, but we can also provide concept or relationship edits. In Figure 16a, we provide the triple edits to explain the
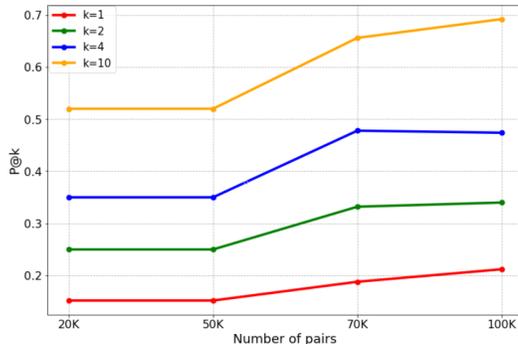
*Table 14.* Ranking results on AG.

| | P@k ↑ | | | NDCG@k ↑ | | | P@k (binary) ↑ | | | NDCG@k (binary) ↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | k=1 | k=2 | k=4 | k=1 | k=2 | k=4 | k=1 | k=2 | k=4 | k=1 | k=2 | k=4 |
| GCN-25K | 0.17 | 0.21 | 0.27 | 0.70 | 0.70 | 0.72 | 0.17 | 0.26 | 0.41 | 0.21 | 0.27 | 0.35 |

*Table 15.* Average number of node, edge & total edits on AG.

| | Node↓ | Edge↓ | Total↓ |
|---|---|---|---|
| GCN-25K | 4.87 | 7.99 | 12.86 |

Parakeet Auklet → Least Auklet counterfactual transition. Similarly, in Figure 16b we present global edits for concepts appearing on CUB images. The results align with human perception.
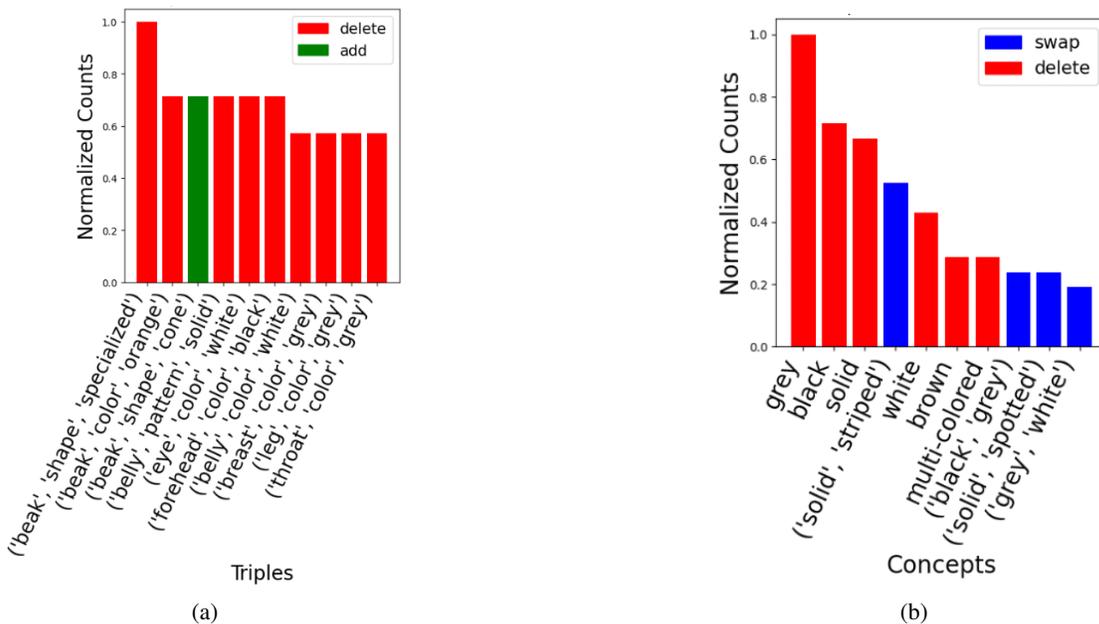


(a)



(b)

*Figure 16.* Triple and concept edits (insertions, deletions, substitutions) to perform Parakeet Auklet → Least Auklet transition.

# E. Qualitative analysis

### E.1. Counterfactual graph geometry on CUB

Our framework is capable of retrieving counterfactual graphs that not only respect node and edge semantics, but also graph geometry. This observation corresponds to more accurate retrieval capabilities that focus on semantic information regarding bird species without being significantly distracted from irrelevant characteristics such as the background. This can be an encouraging characteristic of our counterfactuals towards more robust explanations, even though this aspect is not analyzed in the current paper. First, we present a qualitative example of this claim. In Figure 17, we search for the most similar image to 17a using the method of CVE and ours.

Apparently, both counterfactual images are visually similar, as appearing in Fig. 17b and 17c. However, the representation power of scene graphs becomes evident in this case. In Fig. 18 we present the scene graphs corresponding one-to-one to the images of Fig. 17. The most similar graphs of 18a correspond to the graph of 18b according to CVE and 18c according to our approach. It is evident that our approach can successfully retrieve graphs that **better respect the geometry** of the source
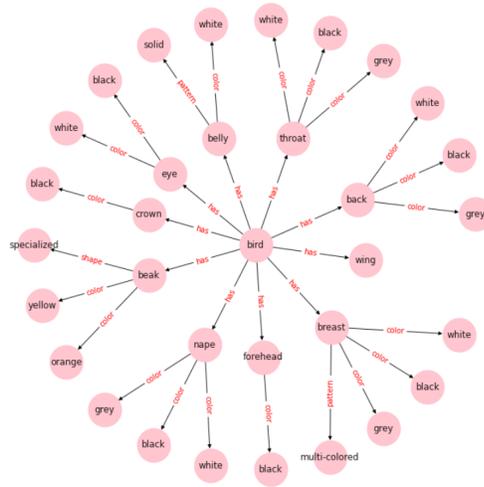
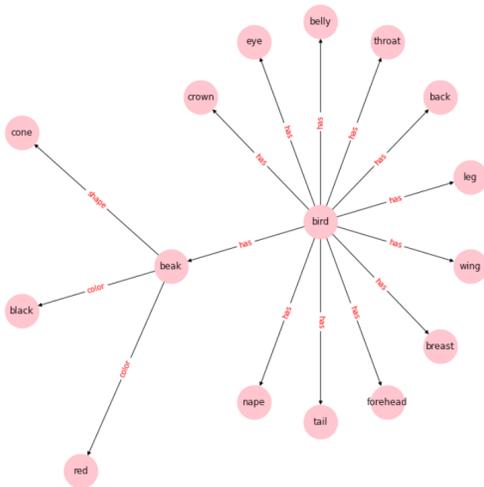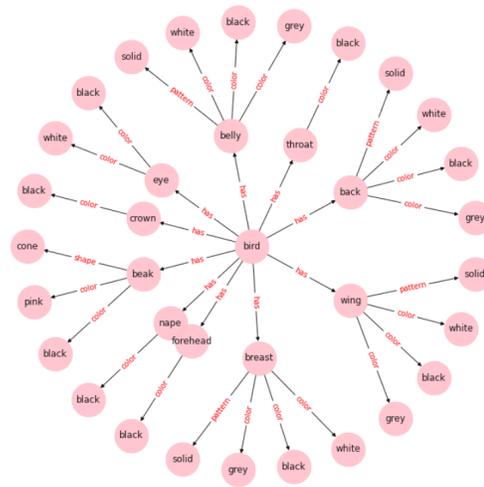(a) Query image      (b) Top-1 retrieved by CVE      (c) Top-1 retrieved (ours)

*Figure 17.* A counterfactual explanation example.



(a) Graph of class Parakeet Auklet corresponding to query image of Fig. 17a.



(b) Counterfactual graph of target class Least Auklet corresponding to Fig. 17b (as retrieved by CVE).

(c) Counterfactual graph of target class Least Auklet for Fig. 17c (as retrieved by our GCN-70K).

*Figure 18.* Example of scene graph structures of counterfactual graphs for Parakeet Auklet → Least Auklet class transition.

image scene graph. Another observation is that our approach manages to retrieve an image without the concepts 'leg' or 'tail' which is more accurate compared to the source. Therefore, structural similarity leads to better semantic consistency.

### E.2. Graphs of Visual Genome

In Fig. 19 (VG-DENSE) and 20 (VG-RANDOM) we present the corresponding graphs to counterfactual images of Visual Genome produced by our method and the method of SC (Dervakos et al., 2023).
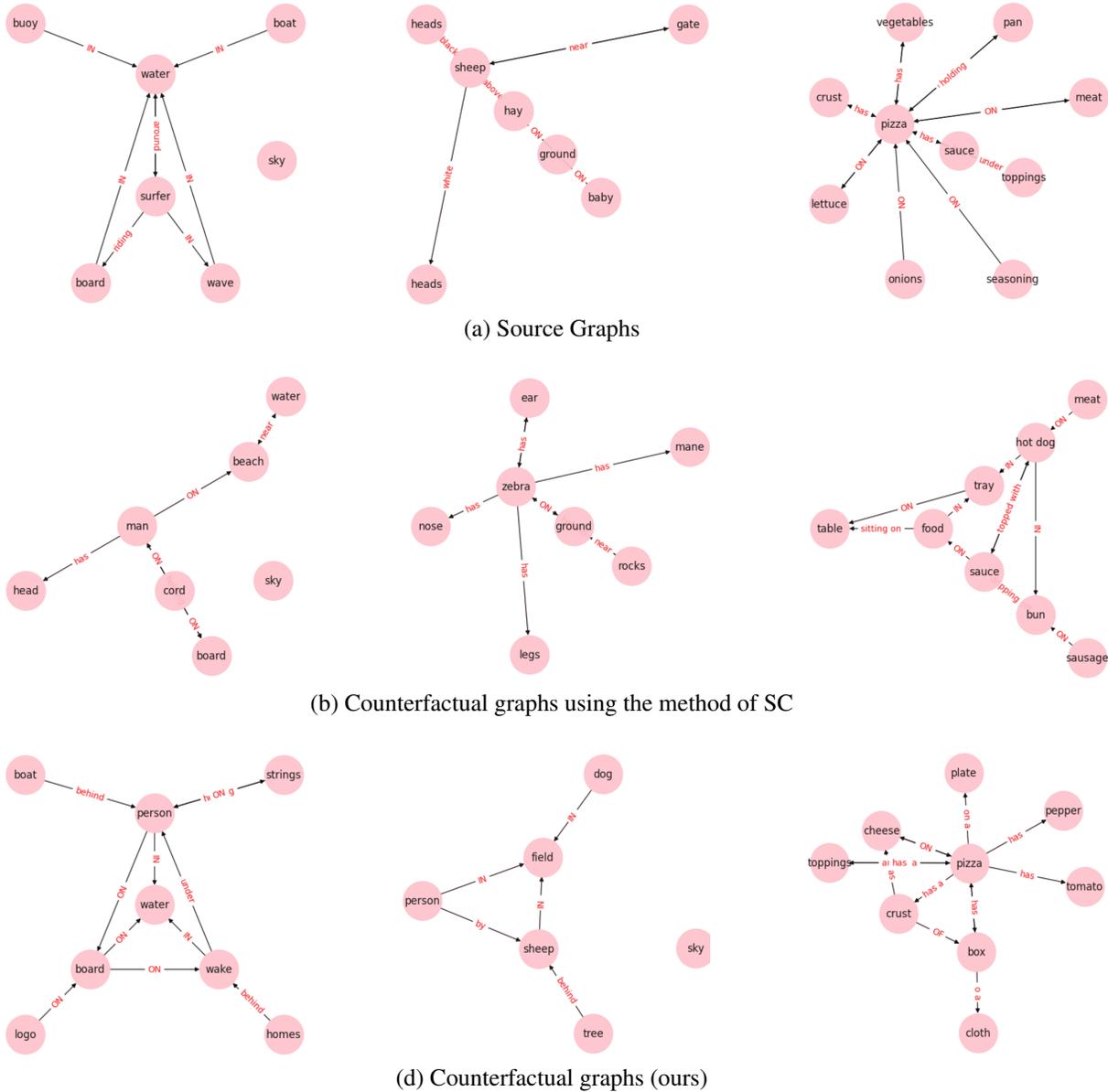


(a) Source Graphs



(b) Counterfactual graphs using the method of SC



(d) Counterfactual graphs (ours)

*Figure 19.* Qualitative Results on graphs for counterfactuals presented in Fig. 4 of the main paper for VG-DENSE.

Inspection of VG-DENSE graphs clearly indicates that our method retrieves counterfactual instances that not only have similar concepts on nodes and edges but are also **structurally closer**. Suggesting counterfactual images with emphasis on object interactions leads to more accurate and meaningful explanations. For instance, in the first column, the relation 'surfer riding board' translates to 'man on board' for our method, whereas for SC (Dervakos et al., 2023) the man is essentially holding the board ('cord on board', 'cord on man').

In the case of VG-RANDOM where graphs have many isolated nodes and fewer edges, the comparison is not as

straightforward. In columns 1 and 2 of Fig. 20, our method retrieves visually more similar instances by combining semantics and structure; thus, managing to preserve the main interacting concept of the image. However, when relations are sparse in the source graph, a greater amount of similar concepts will lead to better counterfactuals.



(a) Source Graph

(b) Counterfactual (SC)
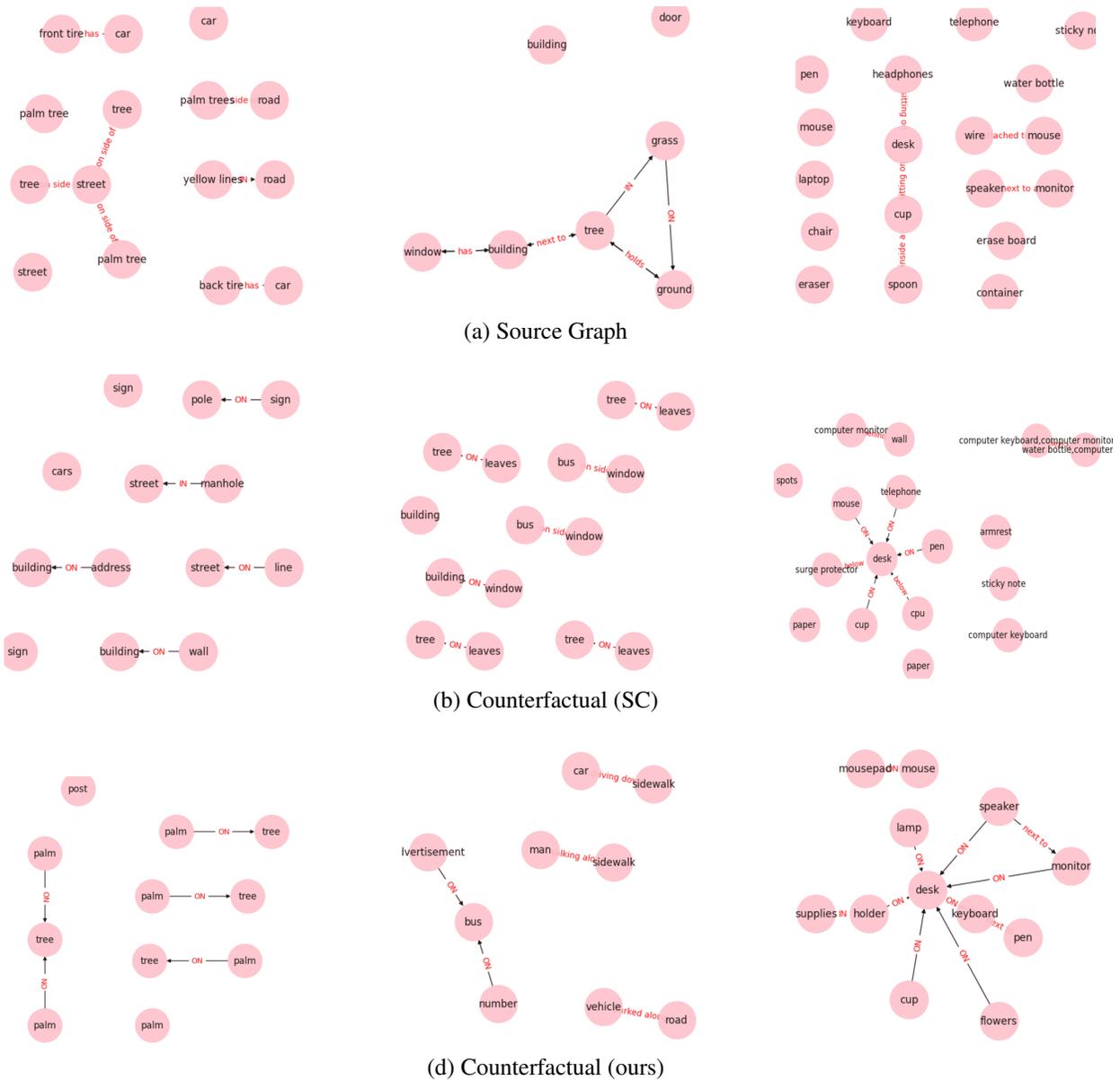
(d) Counterfactual (ours)

Figure 20. Qualitative Results on graphs for counterfactuals presented in Fig. 4 of the main paper for VG-RANDOM.

### E.3. Actionability of edits

We present two non-actionable counterfactual explanations produced by CVE and leverage their method of converting visual CEs into natural language. This approach enables us to precisely define changes between the query and counterfactual instances, which would be challenging with purely visual information. Having emphasized the importance of high-level semantics for human-interpretable CEs, we evaluate the inferred explanations based on linguistic cues rather than pixel-level edits. Both provided examples are deemed successful explanations.
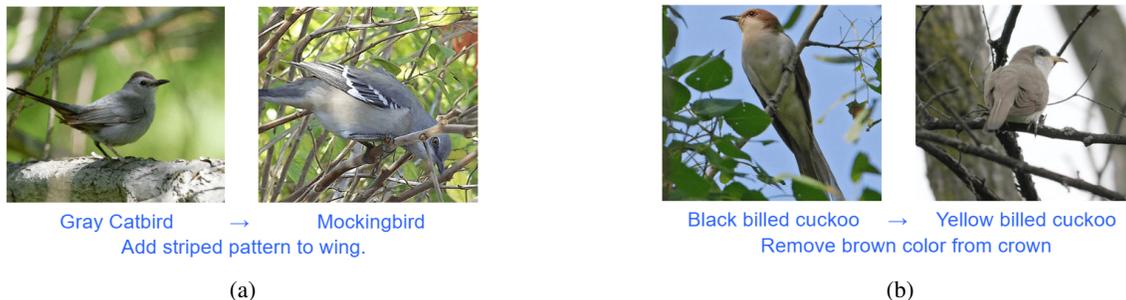
Gray Catbird → Mockingbird
Add striped pattern to wing.

(a)

Black billed cuckoo → Yellow billed cuckoo
Remove brown color from crown

(b)

*Figure 21.* Counterfactual images from CVE and the proposed explanations using natural language.

(Vandenhende et al., 2022) often propose single edits on the query image (left images of Figs. 21a, 21b) and deem them sufficient for the transition from query to target class. However, as explained in Sec. 4.1 of the main paper, this approach disregards the rest of the edits needed to be made between $I_{(A)}$ and $I_{(B)}$ and leads to instances that are in fact out-of-distribution. In the main paper, we gave an example that corresponds to Fig. 21a. In addition to the combination ('has_head_pattern::eyering', 'has_breast_color::grey') that was reported in text, we provide several other attribute combinations that do not exist in any other bird of the target class in Tab. 16.

Furthermore, we present one more example in Fig. 21b. (Vandenhende et al., 2022) claim that removal of the brown color from the crown of the Black billed cuckoo in Fig. 21b (left) is sufficient for it to be classified as a Yellow billed cuckoo. After performing such an edit we obtain a new bird instance that retains the same features as the bird depicted in Fig. 21b (left), except it no longer has a brown crown. By generating all pairs of attributes of this new bird, we discover that none of the attribute pairs listed in Tab. 16 are representative of any bird in the target class (Yellow billed cuckoo).

It is straightforward to understand that more examples can easily be found throughout the dataset. Given the definition of target classes used in this example (most frequently confused by the classifier), counterfactual pairs are generally visually and semantically close. If we chose a different definition of the target class and picked one that is dissimilar to the query class, we can deduce that the list of out-of-distribution attribute combinations would be much longer.

Regarding our method, actionability, in the sense of counterfactuals being representative of the data distribution, is inherent. This guarantee arises from the fact that counterfactuals are actual samples from the target class, specifically the most similar ones to the query, and that we offer complete explanations.To be precise, the proposed counterfactual explanations consist of lists of all graph edits needed to transit from query $I_{(A)}$ to target $I_{(B)}$.

*Table 16.* Out of distribution attribute pairs for target classes.

| Gray Catbird → Mockingbird |
| --- |
| ('has_head_pattern::eyering', 'has_breast_color::grey') |
| ('has_head_pattern::eyering', 'has_belly_color::grey') |
| ('has_breast_color::grey', 'has_nape_color::brown') |
| ('has_breast_color::grey', 'has_shape::swallow-like') |
| ('has_upper_tail_color::white', 'has_wing_shape::pointed-wings') |
| ('has_breast_color::grey', 'has_primary_color::brown') |
| ('has_throat_color::grey', 'has_shape::swallow-like') |
| ('has_belly_color::grey', 'has_shape::swallow-like') |
| ('has_shape::swallow-like', 'has_leg_color::black') |

| Black billed → Yellow billed Cuckoo |
| --- |
| ('has_upperparts_color::buff', 'has_upper_tail_color::white') |
| ('has_back_color::white', 'has_head_pattern::plain') |
| ('has_upper_tail_color::white', 'has_head_pattern::plain') |
| ('has_upper_tail_color::white', 'has_size::very_small_(3_-_5_in)') |
| ('has_upper_tail_color::white', 'has_back_pattern::solid') |
| ('has_upper_tail_color::white', 'has_leg_color::buff') |
| ('has_head_pattern::plain', 'has_nape_color::white') |
| ('has_nape_color::white', 'has_back_pattern::solid') |
| ('has_nape_color::white', 'has_tail_pattern::solid') |
| ('has_size::very_small_(3_-_5_in)', 'has_bill_color::grey') |
| ('has_leg_color::buff', 'has_bill_color::grey') |

### E.4. Additional results

**CUB** In Fig. 22 we provide some additional visual results of counterfactuals comparing our method with SC and CVE. Despite the visual similarity of the retrieved counterfactual images given all three methods, our approach consistently
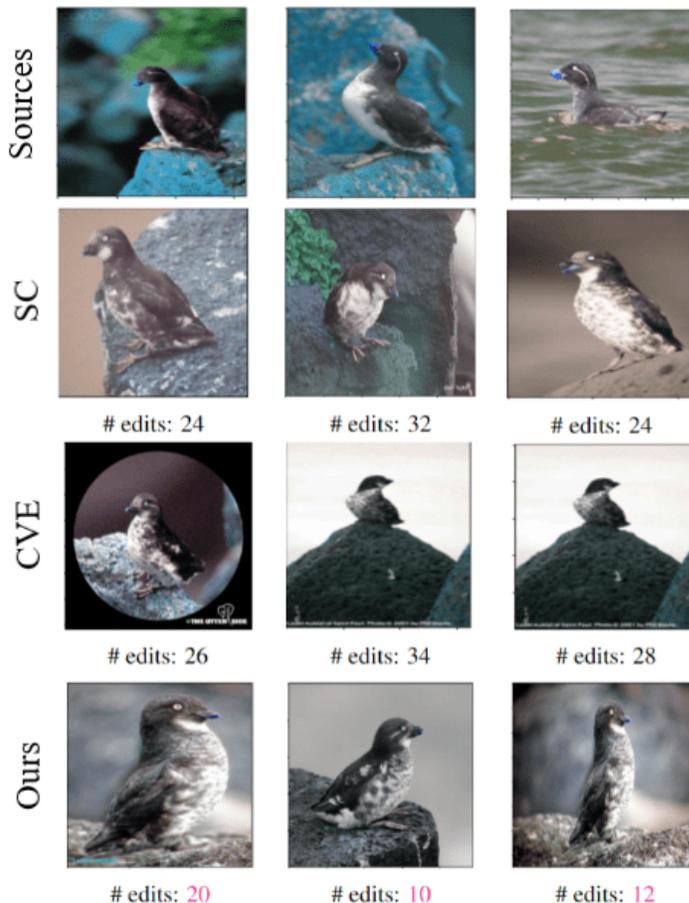
*Figure 22.* Additional qualitative results of counterfactuals of the source class Parakeet Auklet belonging to target class Least Auklet. We also provide number of total edits per method, with colored instances denoting best results.

achieves significantly fewer number of total edits.

## F. Applicability to unannotated datasets

Applicability to unannotated datasets is a valid concern given our approach's dependence on scene graphs. As previously established, graphs of images can be obtained either through manual annotations or automated construction methods. However, not all datasets have such readily available resources, therefore we invest our efforts around proving the applicability of our proposed approach to completely unannotated datasets.

Studying the impact of annotations is an important aspect, since an intrinsic characteristic of semantic explanations is their dependence on the knowledge of the individual that provides them. This inherent explainability attribute impacts systems in the same way it does humans. The knowledge supplied to an explainer will determine the specificity and scope of the explanations. Selecting the appropriate annotation technique is a critical step in receiving the desired breadth and depth of explanations.

In the following experiments, we explore these concerns by extracting counterfactual explanations via our proposed framework on unannotated datasets. Our framework is able to explain *any* classifier in a black-box manner, either being a non-neural classifier (humans in the case of the pedestrian vs driver experiment) or a convolution-based model (Zhou et al., 2017) (in the case of Action Genome).

**Web images: pedestrian vs driver** Dervakos et al. (2023) gather images from Google, Bing, and Yahoo search engines corresponding to 'people', 'motorbikes', and 'bicycles' keywords and their combinations, and then manually split them in 'pedestrian' and 'driver' classes. Finally, 190 'driver' images were obtained (63 images of bicycle drivers and 127 of motorcycle drivers) and 69 'pedestrian' images (31 images of people and parked bicycles, and 38 images of people and parked motorcycles). Those classes are also adopted by us to highlight the importance of relationships (as claimed in Dervakos et al. (2023)), as well as extend this claim to support the usage of graphs over the relationship roll-up of SC. By rolling up the roles and converting them into concepts, we might unintentionally overlook important details for a given task. For example, when examining an image depicting a person on a motorbike in a store, alongside another motorbike on the street, by inspecting the scene graph, it is easy to assume that the scene represents a dealership, with the person testing the motorbike for potential purchase, without actually driving it. However, as (Dervakos et al., 2023) encode this information with the objects: $person, riding.motorbike$, $motorbike, in.store$, and $motorbike, on.road$, they lose the distinction of which motorbike the user is actually riding, potentially leading to erroneous explanations. Nevertheless, leveraging the information within the graph allows us to arrive at more accurate conclusions, especially in fields as critical as Explainable Artificial Intelligence (XAI).

Apart from providing triple edits to explain the 'pedestrian' vs 'driver' classification, we also provide global relationship edits to discover if they are meaningful on their own. Indeed, relationship edits are meaningful in general, especially since the 'riding' relationship is inserted frequently (Figure 23, left plot corresponds to immediately deriving the SGG from the image, while the plot on the left denotes the edits occurring from captioning and then obtaining the graph from the caption). Moreover, the relationship 'on' appears frequently (in the SGG case), again confirming the action of sitting on a bike/motorcycle in order to drive.
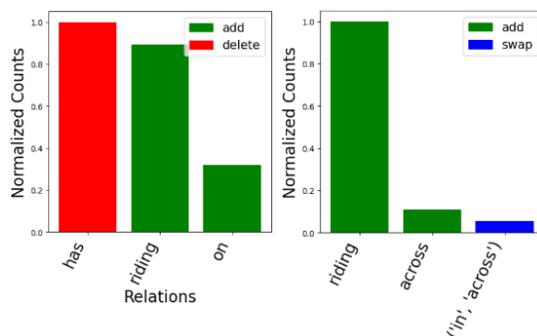


*Figure 23.* Relationships inserted/ deleted/substituted to implement the 'pedestrian' → 'driver' transition.

Similarly, we extract global edits for concepts discriminating the pedestrian/driver categories. These edits are presented in Figure 24. By observing these plots (SGG - left, captioning and graph from text - right) we conclude that these edits are not really meaningful according to human perception: inserting wheels does not explain the 'pedestrian' → 'driver' transition, since in both classes bike/motorbike wheels may appear as part of these vehicles. The same observation is valid for the rest of the concepts appearing on these plots, resulting in noisy conceptual edits. To this end, we verify that *explanations are human-dependable*, i.e. a human is the final evaluator of any explanation, and while a method is able to provide semantically meaningful explanations (in this case relationship edits), it is possible that at the same time the same method provides meaningless explanations (in this case concept edits). Nevertheless, if the derived explanations are not conceptual, a human cannot verify their validity; therefore, we can safely claim that *human interpretability of explanations is highly tied to semantics*.

**Action Genome** We test our method in a real-world image dataset extracted from Action Genome (Ji et al., 2020), a video database depicting human-object relations and actions. It is completely unannotated and also like VG has no predetermined classes for its instances. AG results are not presented in the main paper because they offer no new insights compared to other extendability experiments. However, a brief qualitative analysis was deemed interesting enough to present in the appendix. We select a subset of 300 individual frames and generate scene graphs following well-established SGG methods[3]. After applying our CE method using predictions made by (Zhou et al., 2017), we obtain results comparable to previous experiments. Specifically, the binary retrieval metrics ranged from 0.17 - 0.41 for P@k and 0.21 - 0.35 for NDCG@k,
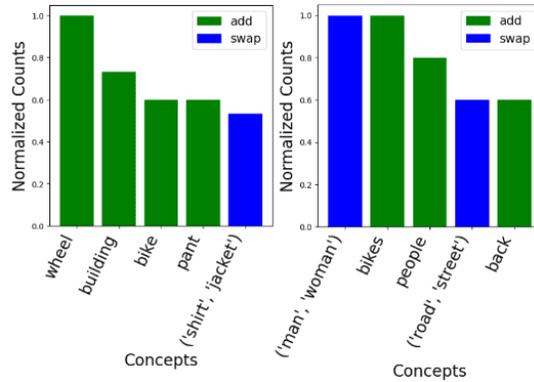
---

[3]SGG on Action Genome

*Figure 24.* Concepts inserted/ deleted/substituted to implement the 'pedestrian' → 'driver' transition.

while overall average edits were 12.86. This experiment validates the relative ease of obtaining graphs from images and demonstrates the applicability of our method to AI-generated graphs of varying quality.

In Fig. 25, we offer some qualitative results on the AG dataset. Instances in this custom AG subset are individual video frames that depict mostly indoor spaces with or without people at a variety of angles and settings. Due to the lack of control in this case, we have identified specific categories that are more meaningful to human perception, such as 'kitchen', 'hall', and 'living room'.

By observing these examples, we can initially note that automatically generated graphs provide a satisfactory representation of the images. However, there are missing details and known biases resulting from imbalanced triple and relation distributions in VG, where the SGG models are trained. We analyze the counterfactuals while acknowledging the potentially lower quality of the input graphs. Since this part of the experiments aims to demonstrate the applicability of our method to unannotated datasets, in-depth analysis is not performed. Nonetheless, we can observe that the retrieved graphs exhibit structural similarities and share common concepts, which is also visually apparent. For instance, images featuring kitchens often involve the removal of cabinets located above counters, while tables are prevalent in hallway depictions.

## G. Applicability on other modalities

We will provide some details on the modality-agnostic nature of our approach, and specifically results on audio classification.

The process of SMARTY graph generation differs compared to our previous experiments in a few key ways. In this new approach, each user or patient was directly connected to their symptoms and characteristics, which were defined to be audible to a certain extent. Symptom analysis involved treating certain symptoms as sub-symptoms when necessary, based on the hierarchical structure presented in Dervakos et al. (2023)'s SMARTY hierarchy, as opposed to using WordNet for computing node edit costs. Regarding edges in the graph, a simpler strategy was adopted due to the limited number of edge types. Specifically, the approach considered edge swaps between different edge types, as well as the addition and deletion of edges, as costly operations.

To initialize the GNN similarity component, custom BioBert (Lee et al., 2020) embeddings were utilized because the language used in the medical field is specific and distinct from general language, unlike previous approaches that relied on simple Glove embeddings. These changes were made to enhance the accuracy and relevance of the SMARTY graph generation.

In Table 17 comprehensive global edit lists can be found. It is important to note that in Table 17, triple edits refer to edge edits and the concepts adjacent to them. For the sake of readability, we have omitted the head and predicate of the triples, where all heads are the 'User' concept and all predicates represent symptoms or sub-symptoms. The second half of 17 on the other hand, focuses on node edits, regardless of edges. Evidently, there is agreement with Table 17, but there are also additional noteworthy findings. One of these findings relates to the reported gender bias mentioned in Dervakos et al. (2023), and another suggests a correlation between COVID-19 positivity and younger users.

*Table 17.* Global triple and concept edits for COVID-19 Negative → Positive.

| Concept Edits | Normalized Counts |
|---|---|
| 'Sneezing' | 1.0 |
| 'RunnyNose' | 0.78 |
| 'DryThroat' | 0.35 |
| 'Fever' | 0.34 |
| 'Dizziness' | 0.31 |
| 'Fatigue' | 0.22 |
| 'Respiratory' | 0.22 |
| 'DryCough' | 0.21 |
| 'TasteLoss' | 0.21 |
| 'Cough' | 0.16 |

| Triple Edits | Normalized Counts |
|---|---|
| 'Sneezing' | 1.0 |
| 'RunnyNose' | 0.73 |
| ('Male', 'Female') | 0.68 |
| 'DryThroat' | 0.36 |
| 'Fever' | 0.35 |
| 'Dizziness' | 0.31 |
| ('Fourties', 'Twenties') | 0.29 |
| 'DryCough' | 0.23 |
| 'Fatigue' | 0.23 |
| 'Respiratory' | 0.23 |

## H. Limitations

Our work is subject to certain limitations. First of all, our experiments involving the CUB and VG datasets are highly dependent on the existing annotations, thus influencing the quality of the derived conceptual explanations. Specifically, the generated semantics through SGG are influenced by the training datasets, namely VG. This limitation was addressed through the comparison of our method's consistency among two vastly different graph generation methods. Despite the positive results validated by the similar produced global edits, there is much room for exploration in this domain. We plan to engage in this venture in our future research. Moreover, pre-trained image classifiers, such as ResNet50 and Places365 may produce imperfect labels for the images under consideration, which may influence the resulting counterfactual explanations. CEs are also characterized by known limitations, such as robustness (Slack et al., 2021). While we have not addressed this particular limitation in our work, we plan to explore it in our future work. Despite these limitations, we have ensured actionability guarantees with the aim of improving the quality of the provided counterfactuals.
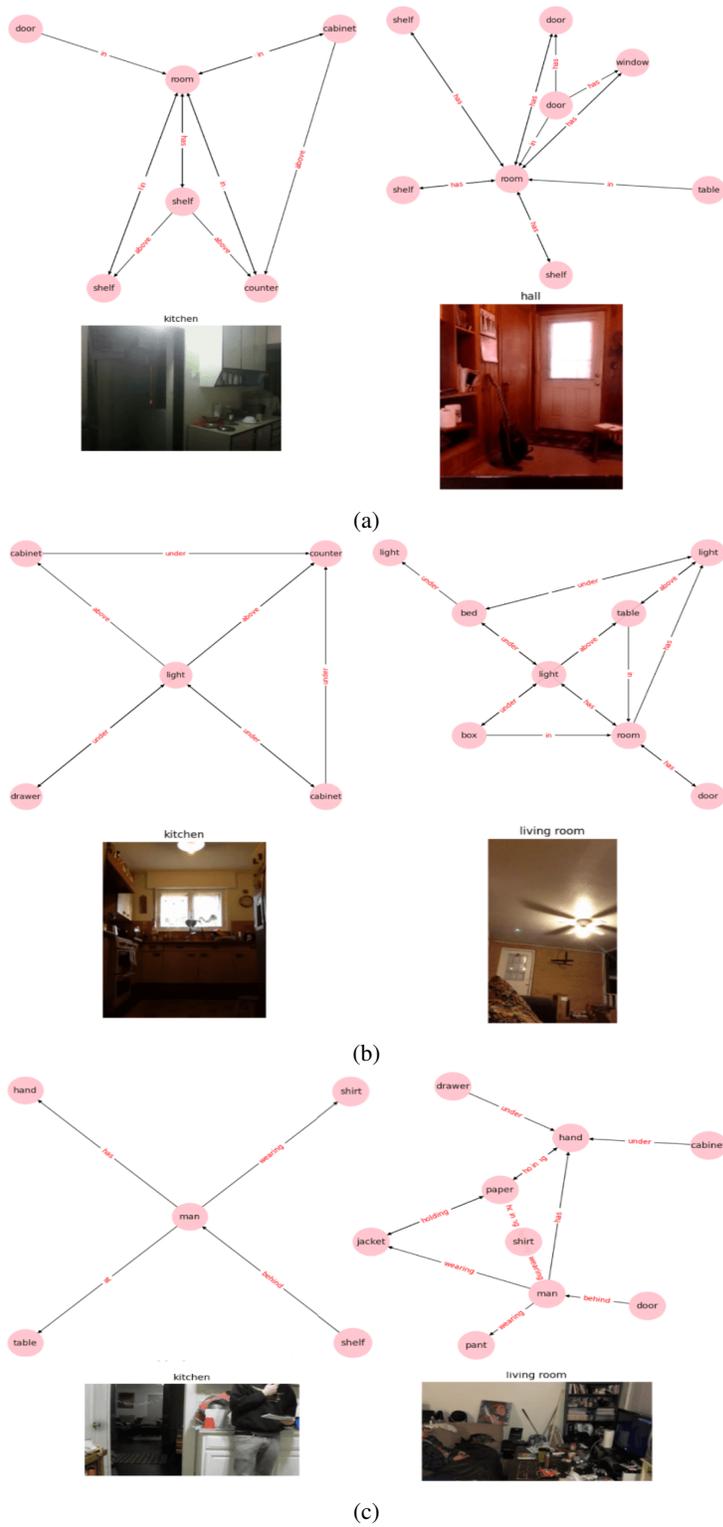
(a)



(b)



(c)

*Figure 25.* Counterfactual examples from AG dataset for query images belonging to the class "kitchen". Here, CEs are classified as "hall" or "living room".