Published: 09 April 2024
Online AM: 02 September 2023
Accepted: 21 August 2023
Revised: 28 July 2023
Received: 21 October 2022

RESEARCH-ARTICLE

A Survey on Graph Counterfactual Explanations: Definitions, Methods, Evaluation, and Research Challenges

MARIO ALFONSO PRADO-ROMERO, Gran Sasso Science Institute, L'Aquila, AQ, Italy

BARDH PRENKAJ, Sapienza University of Rome, Rome, RM, Italy

GIOVANNI STILO, University of L'Aquila, L'Aquila, AQ, Italy

FOSCA GIANNOTTI, School Normal Superior of Pisa, Pisa, RM, Italy

A Survey on Graph Counterfactual Explanations: Definitions, Methods, Evaluation, and Research Challenges

MARIO ALFONSO PRADO-ROMERO, Gran Sasso Science Institute, Italy
BARDH PRENKAJ, Sapienza University of Rome, Italy
GIOVANNI STILO, University of L'Aquila, Italy
FOSCA GIANNOTTI, Scuola Normale Superiore, Italy

Graph Neural Networks (GNNs) perform well in community detection and molecule classification. Counterfactual Explanations (CE) provide counter-examples to overcome the transparency limitations of black-box models. Due to the growing attention in graph learning, we focus on the concepts of CE for GNNs. We analysed the SoA to provide a taxonomy, a uniform notation, and the benchmarking datasets and evaluation metrics. We discuss fourteen methods, their evaluation protocols, twenty-two datasets, and nineteen metrics. We integrated the majority of methods into the GRETEL library to conduct an empirical evaluation to understand their strengths and pitfalls. We highlight open challenges and future work.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Neural networks**; **Artificial intelligence;**

Additional Key Words and Phrases: Explainability, explainable AI, counterfactual explainability, post-hoc explanation, graphs, graph neural networks, graph learning, molecular recourse, black box problem, fairness in AI, machine learning

ACM Reference format:
Mario Alfonso Prado-Romero, Bardh Prenkaj, Giovanni Stilo, and Fosca Giannotti. 2024. A Survey on Graph Counterfactual Explanations: Definitions, Methods, Evaluation, and Research Challenges. *ACM Comput. Surv.* 56, 7, Article 171 (April 2024), 37 pages.
https://doi.org/10.1145/3618105

1 INTRODUCTION

Explaining predictions is crucial for enabling trusted decision-making in sensitive domains for both users and service providers [25]. However, the prevalent use of deep neural networks in gener-

This work is partially supported by the European Union - NextGenerationEU - National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) - Project: SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics - Prot. IR0000013 - Avviso n. 3264 del 28/12/2021, XAI: Science and technology for the eXplanation of AI decision - ERC Advanced Grant 2018 G.A. 834756 and by the HPC & Big Data Laboratory of DISIM, University of L'Aquila (https://www.disim.univaq.it/).
Authors' addresses: M. A. Prado-Romero, Gran Sasso Science Institute, L'Aquila 67100, Italy; e-mail: marioalfonso.prado@gssi.it; B. Prenkaj, Sapienza University of Rome, Rome 00198, Italy; e-mail: prenkaj@di.uniroma1.it; G. Stilo, University of L'Aquila, L'Aquila 67100, Italy; e-mail: giovanni.stilo@univaq.it; F. Giannotti, Scuola Normale Superiore, Pisa 56126, Italy; e-mail: fosca.giannotti@sns.it.

0360-0300/2024/04-ART171
https://doi.org/10.1145/3618105

ating predictions has given rise to the issue of the "*black box*" problem [57]. These complex models hinder the comprehension of the decision-making process employed for forecasting outcomes. Due to their reliance on nonlinear activations for learning feature representations, deep neural networks remain opaque to users, resulting in limited adoption in critical domains like health and finance. In contrast, there are "*white box*" or transparent models, which allow for easy inspection of their inner processes [37]. Although white-box models are preferred for decision-making [71], black-box models outperform them in scenarios involving high-dimensional data, such as protein relation prediction [43], student dropout prediction [61, 62], and trend forecasting [72].

Recently, **Graph Neural Networks** (**GNNs**) [66] have emerged as a promising solution to various graph mining tasks, such as vertex classification [54, 76, 89] and link prediction [80, 87]. GNNs take as input a graph composed of vertices and edges. Examples of graphs are Facebook's user friendship network and PPIs that collect proteins and their relations. In these scenarios, the vertices are users (proteins), and the edges are the friendships (chemical relations). For instance, in the case of Facebook's user friendship network, GNNs[1] can be used to predict if two users will become friends in the future, by leveraging information from their neighbourhoods in the graph. GNNs operate by transforming the input graph, using learned latent features for both vertex and edge attributes. The connectivity patterns induced by the edges enforce the relationships in the latent vector space of the learned features (see Section A.1 of the supplementary material). However, as black-box models, GNNs are not suitable as **decision support systems** (**DSS**) in critical domains.

The literature has received many contributions in interpreting black-box models via feature-based analysis [24] and counterfactual explanations [23, 39]. Models that perform feature analysis to produce explanations are denoted *feature-based* explainers. However, explanations that reveal a prediction's *why* need to be more comprehensive to understand how to change the outcome of a specific model. Thus, explainers - namely *counterfactual explainers* - that provide examples of what input features to change to obtain a different prediction are necessary to describe cause-effect relationships between the data and the outcome [11]. Counterfactual explainability could help, for example, obtain the following suggestions:

User banning DSS — Suppose a user within a fictitious social network posts content attempting to sell drugs illicitly. Such conduct directly violates the network's terms of use, resulting in potential legal consequences and the suspension of the user's profile. To address this, our DSS takes action by blocking the user's account and offers a counterfactual explanation, stating that "*had the user refrained from posting about drug sales, her account would not have been banned*". The provision of such explanations aids auditors in classifying banned users based on the seriousness of their violations, thereby contributing to a safer online environment for other users.

Drug repurposing DSS — Assume that a drug laboratory aims at curing bacterial infections (e.g., dental abscess) with cephalexin.[2] When treating such diseases with this drug, our DSS gives a negative result with the explanation, "*if we modified cephalexin's molecular structure as in Figure 1, then we would be able to treat the diseases*". This explanation is useful because it would give sprout to the aminopenicillins class of antibiotics, particularly the amoxicillin[3] drug known for its faster treatments and lower side effect risks.

Counterfactual explanations enhance the analysis of black-box models, offering transparency and feedback to non-experts. This fosters trust in critical domains like health and finance and

[1]Facebook has a friend suggestion functionality that takes information from already-established friendships between two users and suggests other similar profiles to make plausible new friendship connections. Because Facebook"s prediction algorithms remain proprietary, we can assume that GNNs would benefit from this particular task due to their intrinsic characteristics of extrapolating information from the neighbourhoods of the graph vertices.

[2]https://pubchem.ncbi.nlm.nih.gov/compound/cephalexin

[3]https://pubchem.ncbi.nlm.nih.gov/compound/amoxicillin

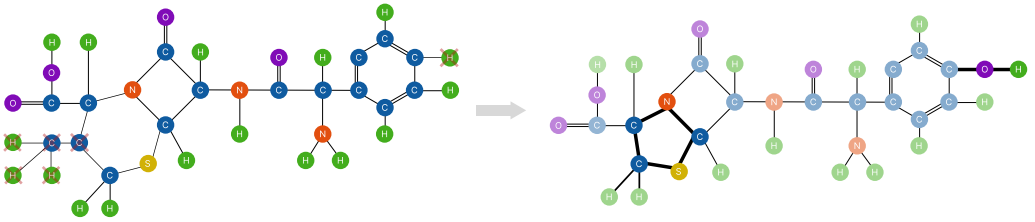


Fig. 1. Drug repurposing mechanism. Passing from cephalexin (left) to amoxicillin (right) by removing atoms and reforming hexagonal connections into pentagons. Here, the counterfactual explanation consists of the highlighted edges and vertices.

helps identify biases. For example, in the user banning DSS, race-related bias in account suspension can be detected [9]. Addressing biases based on ethical principles rather than purely performance-driven concepts becomes feasible through counterfactual explainability. However, eliminating bias from data remains a separate and challenging endeavour.

From a market's perspective, developing a new drug is expensive, costing between $314 million and $2.8 billion as of 2018, with a low success rate of only 12% during clinical development [81]. Clinical trial complexity, larger trial sizes, changes in protocol design, and testing on comparator drugs contribute to this price hike. Clear counterfactual methods can help decipher the causes of outcomes, particularly in scenarios such as drug repurposing, potentially reducing costs and increasing approval rates. Therefore, pharmaceutical companies are interested in adopting innovative explainable graph learning approaches to comprehend trial scenarios without costly tests.

1.1 Contributions of this Survey

Graph Counterfactual Explainability (**GCE**) is still in its early stages since significant attention towards graph learning has risen sharply in the last decade.

This survey addresses GCE, going beyond mere feature-based (factual) explainability in graphs. We provide an in-depth discussion and a systematic literature review of GCE. Specifically, we organise existing counterfactual methods according to a uniform formal notation to facilitate easy comparison of their strengths and pitfalls (See Section 2.1). Additionally, we propose a formalisation of GCE for multi-class prediction problems, deriving the global minimal counterfactual example on a specific graph black-box prediction model (See Section 2.2). We summarise the strengths and weaknesses of counterfactual explainer methods present in the literature, classifying them according to several dimensions, making it easy for readers to identify alternative methodologies that better suit their scenario (See Section 3). In Section 4, we discuss the benefits of the evaluation protocol of GCE methods, including the benchmark datasets and measures used in the literature. We integrated the majority of the methods into the GRETEL library, thus conducting an empirical evaluation among them (see Section 5). In Section 6, we give insights on how to adapt privacy-preserving counterfactual explainers and briefly discuss oracle fairness. Open challenges and promising directions in GCE are summarised in Section 7, and concluding remarks are presented in Section 8. Supplementary material containing background concepts and notation (Section A), detailed dataset descriptions (Section B), evaluation framework comparisons and GRETEL overview (Section C), and the employed hyperparameters adopted in the empirical evaluation (Section D).

Nevertheless, we analysed surveys on graph factual explainability [2, 88] and counterfactual explainability on other data structures for completeness [3, 23, 73] since they are the two closest areas. A recent survey [26] has emerged on Counterfactual Learning that can be useful to have a broader overview of *Trustworthy AI* applications. Still, ours delves much deeper into the formal

Table 1. Qualitative Comparison of the Surveys Present in the Literature According to the Dimensions Reflecting the Scope and Workflow of GCE Approaches

			Surveys on Explainability						
			Considering graphs data				Considering other data types		
			Counterfactual		Factual		Counterfactual		
			[this]	Guo et al. [26]	Amara et al. [2]	Yuan et al. [88]	Artlet and Hammer [3]	Guidotti [23]	Verma et al.[73]
Domain		Synthetic networks	✓	✓	✓	✓	·	·	·
		Social networks	✓	✓	✓	✓	·	·	·
		Molecular networks	✓	✓	·	✓	·	·	·
		Omics Networks	✓	✓	·	·	·	·	·
Prediction Task		Vertex classification	✓	✓	·	✓	·	·	·
		Link (edge) prediction	✓	✓	·	·	·	·	·
		Graph classification	✓	✓	·	✓	·	·	·
		Graph-pair affinity	✓	·	·	·	·	·	·
Definition			✓	✓	·	·	✓	✓	✓
Explainer Adaptation			✓	✓	·	✓	✓	·	·
Evaluation	Datasets	Synthetic	✓	✓	✓	✓	·	·	·
		Real	✓	✓	✓	✓	·	✓	✓
	Reproducibility		✓	·	~	~	·	~	·
	Extensibility		✓	·	·	·	·	·	·
	Supported Methods		✓	·	✓	✓	·	✓	·
	Evaluation Protocol		✓	·	·	·	·	·	·
	Metrics	Runtime	✓	·	✓	·	·	✓	✓
		Oracle Calls	✓	·	·	·	·	·	·
		Oracle Accuracy	✓	·	·	·	·	·	·
		Correctness	✓	·	~	·	·	·	✓
		Sparsity	✓	✓	·	✓	·	·	✓
		Fidelity	✓	✓	✓	✓	·	·	·
		Robustness	✓	✓	·	✓	·	✓	·
		Explainer Accuracy	✓	✓	✓	✓	·	·	·
		Prediction Distance	✓	·	·	·	·	·	·
		Causality	✓	·	·	·	·	·	✓
		Diversity	✓	·	·	·	·	✓	✓
		Actionability	✓	·	·	·	·	✓	·
		GED	✓	✓	·	·	·	·	·
		Explanation Size	✓	·	·	·	·	~	~
		Tanimoto Similarity	✓	·	·	·	·	·	·
		MEG Similarity	✓	·	·	·	·	·	·
		Recourse Cost	✓	·	·	·	·	·	·
		Coverage	✓	·	·	·	·	·	·
		Compactness	✓	·	·	·	·	·	·
Privacy of Explainers			~	·	·	·	·	·	·
Fairness of Oracles			~	✓	·	·	·	·	·

· depicts a missing aspect, ✓a covered aspect, ~ a partially covered aspect.

notation, the methods' analysis, and the benchmarking datasets and metrics, as it is possible to notice in Table 1. In addition to it, we discuss in depth the evaluation protocols of each work. We present an empirical evaluation of the methods and suggest possible strategies for privacy-preserving counterfactual explainers.

We analysed the existing surveys according to seven dimensions (see Table 1), reflecting the scope, the domains and the typical workflow of GCE's research area. *Domain* depicts the considered graph structure. *Prediction Task* represents the goal of the explainable methods. *Definition* depicts formal and uniform definitions adopted for counterfactual explainability. *Explainer Adaptation* illustrates the extension of feature-based explainers for counterfactual generation purposes. *Evaluation* illustrates the evaluation of methods via some established evaluation protocol and designated metrics. *Privacy of Explainers, and Fairness of Oracles* considers the importance of privacy maintenance at the explainers' level and the fairness of the underlying prediction models. We argue that these dimensions are a better framework for categorising and comparing existing surveys.

Domain — We identified three[4] main categories of graphs: i.e., social, molecular, and -omics networks. Other surveys concentrate on social [2, 26, 88] and molecular networks [26, 88] since they

[4]Notice that synthetic datasets have been adopted in GCE, and we include them in the table. Alas, they do not constitute a category of real-world datasets.

represent traditional structures in deep graph learning. Besides extensively discussing the first three graph types, we also shed light on the structure of -omics networks [42], useful for analysing the interconnection of biological processes.

Prediction Task — In this context, we have identified four types of tasks, namely vertex classification, graph classification, link prediction, and graph-pair affinity. These tasks rely on different concepts and have unique characteristics. Vertex classification aims at predicting the role of each vertex in a graph by analysing the role of its neighbouring nodes. For example, the prediction task in Zachary's karate club is to classify whether a member will switch to another club. Link prediction, on the other hand, predicts relationships between vertices. For instance, in an image whose entities are vertices, the task is to predict which of them share an edge. Graph classification predicts the property of an entire graph, such as whether a molecule - represented as a graph - binds to a receptor involved in a specific disease. Finally, graph-pair affinity involves the prediction of the similarity between pairs of instances, such as a drug-target binding for drug discovery or drug repurposing.

Definition — Although graph neural networks [66, 70] and counterfactual explainability [3, 23, 26, 73] have been extensively defined, this survey is the first contribution in the literature which provides a uniform GCE definition. First, we agglomerate the literature's graph counterfactual definitions according to a single notation, and then, we provide a formalisation of the GCE problem. Thus, future researchers can use it to align their work with the SoA.

Explainer Adaptation — Some methods in GCE use factual explainability as a starting point and change the features of the factual to produce the counterfactual [26, 88]. Here, we analyse the counterfactual technique as an adaptation similar to what has been done in factual explanation approaches or as an extension of image-based explainability.

Evaluation — A crucial aspect of evaluating methods in the literature is a uniform benchmarking system that uses standardised metrics and datasets. While other surveys [2, 26, 73, 88] mention the characteristics and potential issues of synthetic and real datasets, only Guidotti [23] and Yuan et al. [88] provide in-depth explanations of the reproducibility[5] of their evaluation framework by sharing links to repositories that contain the code base. These frameworks have several explanation methods from the literature that are readily available. However, they [2, 23, 88] are challenging to adapt and reuse for experiments in different scenarios than those already included. This survey focuses in empirically assessing the performance of SoA explainers using GRETEL, a modular framework for evaluating GCE. GRETEL has an advantage when it comes to reproducibility (see Section C.2). Additionally, we provide the reader with the evaluation protocol used in each article to shed light on various counterfactuality scenarios, thus, the first survey to give such detail. We included several metrics proposed in the literature to evaluate the explainers' goodness, such as oracle accuracy, sparsity, and accuracy. The surveys in [2, 25, 73] also cover the runtime of the explanation technique besides the three previous ones. Besides using only **graph edit distance** (**GED**) as in [26], we include two classes of metrics - i.e., minimality evaluation (*GED*, *Explanation Size*, *Tanimoto Similarity*, and *MEG Similarity*) and global metrics (*Recourse Cost*, *Coverage*, and *Compactness*) useful for future investigations.

Privacy of Explainers and Fairness of Oracles — An important issue regarding trustworthiness is the interplay between privacy, explainability, and fairness. We observed that there is very little attention posed in the GCE domain. Specifically, all the other surveys fail to address the privacy infringement that explainers could introduce when presenting counterfactual examples to end-users, given the vast amounts of sensitive information generated by social networks. On the other

[5]Notice that Amara et al. [2] briefly discuss their explainability but do not show what hyperparameters to use for reproducibility purposes.

hand, counterfactual explanations on graphs can be employed as a mechanism to understand and cope with the oracle's bias. For this reason, Section 6 proposes how to construct privacy-preserving explanation strategies. We discuss how explainability can be harnessed to tackle the unfairness of the underlying model by using generated counterfactuals as an intervention mechanism to cope with such situations.

Table 1 highlights several limitations of factual surveys on graphs. Firstly, they rely solely on synthetic, social, and molecular networks. Secondly, they fail to explain the unique characteristics of link prediction tasks and how they differ from vertex and graph classification tasks. Additionally, these surveys provide a list of explanation techniques without analysing their primary limitations. While some explainers are suitable for all prediction tasks, others may only be effective in certain scenarios. Contrarily, Guo et al. [26] incorporate these aspects. Nevertheless, they fail to address evaluation metrics such as minimality evaluation and global measures, which are useful to provide a list of counterfactual examples to explain an entire dataset (or set of instances). They also fail to provide a benchmark of the current SoA: they only list the available frameworks in the literature and do not compare them according to a reproducible, replicable, generalisable, and robustness point-of-view (see Section C.1).

2 DEFINING COUNTERFACTUAL EXPLANATIONS IN GRAPHS

Here, we discuss the GCE definitions used in the literature. Notice that not all the works have provided a formal graph explanation. Most of them present a loss function, typically based on the distance between the prediction of the input instance and its counterfactual. Hence, we provide a uniform definition as a cornerstone in this research area, discussing the (dis)similarities of what is proposed in the literature (see Sect 2.1). Then, we provide our definitions of multi-class and global minimal counterfactuals (see Section 2.2). We refer the reader to Section A.1 for details on the notations and background knowledge useful to follow this article.

2.1 Unifying GCE Definitions in the Literature

The literature in GCE has tried to provide a formalisation for the problem definition and, eventually, the way a counterfactual example is defined. Although the definitions do not follow a rigorous formalisation, the literature states that a counterfactual example satisfies the following equation:

$$\Phi(x) \neq \Phi(x'), \tag{1}$$

where x is the input instance (i.e., graph, vertex, or edge), $x' \in X'$ is the counterfactual example, X' is a set of possible counterfactuals, and $\Phi(\cdot)$ is the prediction of the oracle Φ. For example, in the case of graph classification $x = G$ and its counterfactual $x' = G'$ where $G = (V, E)$ is a graph with vertices V and edges E. Nevertheless, we believe that the following equation is a more inclusive way to define a counterfactual for both classification and regression tasks.

$$\mathcal{D}_{pred}(\Phi(x), \Phi(x')) \geq t, \tag{2}$$

where $\mathcal{D}_{pred}(\Phi(x), \Phi(x'))$ is the distance function between the prediction of the original instance and the counterfactual one, and t is a threshold. The multi-class and the multi-label classification tasks can be addressed by a specialised $\mathcal{D}_{pred}$.

In this survey, we generalise the generation of a counterfactual via the minimisation of a loss function because the majority of the works follows an optimisation process. Hence, the generation of a counterfactual in GCE can be expressed as a combination of a minimisation objective and a regularisation term, constrained by the γ term:

$$\arg\min_{\gamma} \alpha \cdot \mathcal{L}_{pred} + \beta \cdot \mathcal{L}_{inst}, \tag{3}$$

where $\mathcal{L}_{pred}$ is an optimisation function over the prediction outcome of the oracle Φ, $\mathcal{L}_{inst}$ is a regularisation term over the original and counterfactual instances, $\alpha, \beta \in [0, 1]$ account for the contribution of each term in the overall optimisation function.

Subsequently, we discuss the (dis)advantages and (dis)similarities of each GCE definition in the literature w.r.t to Equation (3) by specialising them according to the following notation.

We use $\mathcal{S}_{pred}(\Phi(x), \Phi(x'))$ to indicate a similarity function between the outcome of the original instance $\Phi(x)$ and the counterfactual one $\Phi(x')$. Additionally, we denote with $\mathcal{S}_{inst}(x, x')$ the similarity function between the original instance x and the counterfactual one x' and $\mathcal{D}_{inst}(x, x')$ is its distance counterpart.

According to Lucic et al. [39], $\mathcal{L}_{pred}$ is the **negative log-likelihood** (**NLL**), $\alpha = -\mathbb{1}[\Phi(x) = \Phi(x')]$ which produces -1 if $\Phi(x) = \Phi(x')$ and 0 otherwise, $\mathcal{L}_{inst}$ is the distance between x and x', β remains a free hyperparameter, and $\gamma = x' \in X'$. Hence, we can rewrite this optimisation function as follows:

$$\arg \min_{x' \in X'} \mathbb{1}[\Phi(x) = \Phi(x')] \cdot \mathcal{D}_{pred}(\Phi(x), \Phi(x')) + \beta \cdot \mathcal{D}_{inst}(x, x') \tag{4}$$

Notice that the chosen formulation for $\mathcal{L}_{pred}$ admits a non-counterfactual example as a possible generated explanation example, including the original instance that minimises the overall loss. In other words, if $x' = x$, then $\mathcal{D}_{pred}(\Phi(x), \Phi(x')) = \mathcal{D}_{inst}(x, x') = 0$. In this case, the original instance x is also a minimal counterfactual example. Similarly to Equations (12) and (4) supports multi-class classification since the prediction from the oracle $\Phi(x)$ is not enticed to a binary scenario. Nevertheless, Equation (4) cannot support the generation of multiple counterfactual examples x' belonging to different classes w.r.t. the original instance x since it tries to find a global explanation.

Wellawatte et al [79] do not consider $\mathcal{L}_{pred}$ explicitly. Rather, they embed the search for valid counterfactuals in γ, making them robust towards the pitfall of considering the original instance x the minimal counterfactual, as seen in Equation (4). We can rewrite the original formulation as follows:

$$\arg \min_{x' \in X' \mid \Phi(x) \neq \Phi(x')} \mathcal{D}_{inst}(x, x'), \tag{5}$$

where α and $\mathcal{L}_{pred}$ can be omitted, $\beta = 1$, $\mathcal{L}_{inst}$ is the distance between x and x', and $\gamma = x' \in X' \mid \Phi(x) \neq \Phi(x')$. In the original article, the authors rely on the Tanimoto index as a similarity measure between x and x'. To remain compliant with Equation (3), we minimise the negated Tanimoto index instead of maximising it. Equation (5) supports multi-class classification scenarios as in Equation (12), however, it cannot produce a set of counterfactuals for the same input instance x. Furthermore, notice that this optimisation function is the same as Equation (13).

Similarly to [79], Abrate and Bonchi [1] search for a counterfactual graph x' such that the symmetric difference between x and x' is minimised. Only the constraint γ is changed to $\Phi(x) = 1 - \Phi(x')$, which confines finding valid counterfactuals to only binary classification scenarios. Hence, Equation (5) becomes the following:

$$\arg \min_{x' \in X' \mid \Phi(x) = 1 - \Phi(x')} \mathcal{D}_{inst}(x, x'). \tag{6}$$

Numeroso and Bacciu [53] optimise a distance-based loss function for molecules that incorporates their structural information. They exploit both a distance between the outcomes of the original molecule x and the counterfactual one, x', and a similarity between the structure of the two molecules. According to the original article, when performing a classification, Φ emits a probability distribution for a certain set of classes C. Hence, given a molecule in input x and its outcome $\arg\max_{c \in C} \Phi(x)$, the authors produce counterfactuals x' that maximise the outcome of classes

different from c, and the similarity between x and x'. The final form of the proposed optimisation function maximises the negation of the probability of outputting a class equal to c and the structural similarity between x and x' controlled by the weight α and $\beta = 1 - \alpha$, respectively. Maximising the negation of the probability of producing c is fancy wording for maximising the distance between the outcomes on the original instance x and the counterfactual x'. It is important to highlight that the counterfactual can be generated by adding and removing both vertices and edges. Therefore, the counterfactual graph is not necessarily a sub-graph of the original one. Before uniforming what is described here in the same form of Equation (3), it is useful to review the original optimisation function as follows:

$$\arg \max_{x' \in X'} \alpha \cdot \mathcal{D}_{pred}(\Phi(x), \Phi(x')) + (1 - \alpha) \cdot \mathcal{S}_{inst}(x, x'). \tag{7}$$

We can easily translate Equation (7) by considering its dual function (i.e., minimisation) as follows:

$$\arg \min_{x' \in X'} (1 - \alpha) \cdot \mathcal{S}_{pred}(\Phi(x), \Phi(x')) + \alpha \cdot \mathcal{D}_{inst}(x, x'). \tag{8}$$

Notice that the condition γ remains the same. Nevertheless, $\mathcal{L}_{pred}$ changes into its dual form (i.e., from maximising distance of predicted classes $\mathcal{D}_{pred}(\Phi(x), \Phi(x'))$, to minimising their similarity $\mathcal{S}_{pred}(\Phi(x), \Phi(x'))$). A similar reasoning is applied to $\mathcal{L}_{inst}$. Naturally, the weights α and $\beta = 1 - \alpha$ are specular. Notice that this formalisation is a generalisation of Equations (5) and (6) since it introduces the minimisation of $\mathcal{S}_{pred}(\Phi(x), \Phi(x'))$ instead of the constraints $\Phi(x) \neq \Phi(x')$ and $\Phi(x) = 1 - \Phi(x')$, respectively. Similarly, Ma et al. [41] rely on Equation (8) to produce valid counterfactuals. In the original article, the authors include the minimisation of the KL divergence that makes the explainer learn to produce realistic counterfactuals w.r.t. the distribution of the original instances and the ground truth. However, producing realistic counterfactuals can be included as a component of $\mathcal{D}_{inst}(x, x')$.

Nguyen et al. [50] rely on oracles that predict drug-target pair interactions. In this scenario, Φ takes in input pairs of instance $(x_d, x_t) \in X_d \times X_t$ and produces an output, where X_d is the set of drugs, and X_t is the set of targets. Naturally, X'_d and X'_t are the counterfactual sets of X_d and X_t, respectively. Hence, for each instance pair (x_d, x_t), we obtain the minimal counterfactual pair (x'_d, x'_t) by optimising the following function:

$$\arg \min_{x'_d \in X'_d,\, x'_t \in X'_t} \alpha \cdot \mathcal{S}_{pred}(\Phi(x_d, x_t), \Phi(x'_d, x'_t)) + \beta \cdot (\mathcal{D}_{inst}(x_d, x'_d) + \mathcal{D}_{inst}(x_t, x'_t)). \tag{9}$$

Although Huang et al. [28] provide a set of counterfactual explanations for the oracle Φ, we can easily translate their optimisation according to Equation (3). To this end, we rely on the power set $\mathcal{P}(X')$ of all possible counterfactuals, X'. In detail, the authors maximise the coverage of the counterfactual set $X^* \in \mathcal{P}(X')$ w.r.t. the size of the set of instances, $|X|$:

$$\arg \max_{X^* \in \mathcal{P}(X') \mid |X^*| = k} \frac{\left| x \in X \mid \min_{x' \in X^*} \mathcal{D}_{inst}(x, x') \leq \theta \right|}{\left| X \right|}, \tag{10}$$

where θ is an upper-bound for the distance between the original instance x and the counterfactual x', and k is the size of the counterfactual set X^* drawn from $\mathcal{P}(X')$. Notice that the fraction in Equation (10) is in $[0, 1]$, meaning that we can rewrite it as a minimisation function:

$$\arg \min_{X^* \in \mathcal{P}(X') \mid |X^*| = k} 1 - \frac{\left| x \in X \mid \min_{x' \in X^*} \mathcal{D}_{inst}(x, x') \leq \theta \right|}{\left| X \right|}. \tag{11}$$

Notice that Equation (11) is compliant with the notation of Equation (3) where $\gamma = X^* \in \mathcal{P}(X') \mid |X^*| = k$, $\mathcal{L}_{pred}$ is not specified since γ implies only searching for valid counterfactuals, and $\mathcal{L}_{inst}$ is the negated coverage function with $\beta = 1$.

Bajaj et al. [4] are the only ones that provide a formal definition for GCE instead of optimising a loss function. Hence, given a model Φ trained on a set of graphs, for an input graph $G = \{V, E\}$, the authors explain why G is predicted as $\Phi(G)$ by identifying a small subset of edges $S \subseteq E$, such that (1) removing the set of edges in S from G changes the prediction on the remainder $\{V, E - S\}$ of G significantly; and (2) S is stable w.r.t. slight changes on the edges of G and the feature representations of the vertices of G. According to this definition, the authors cannot always change Φ's prediction only relying on edge removals. Generally, it is not always possible to find a set $S \subseteq E$ that maintains the stability w.r.t. changes on G. Thus, generating a counterfactual example is impossible. Additionally, the set $E - S$ contains the factual edges for which $\Phi(G = (V, E - S)) \neq \Phi(G = (V, E))$. Without loss of generality, all methods that generate counterfactual examples by only removing edges suffer from the same phenomenon.

The works in [36, 69] are factual-based methods that can be used to derive counterfactuals. Generally, the counterfactual example can be considered as the remainder of the original graph when the factual explanation (e.g., usually a sub-graph) gets eliminated. Furthermore, by concentrating first in identifying the subset of edges, vertices, and vertex attributes to form a sub-graph for the factual explanation, these works do not guarantee that the counterfactual example is minimal. We do not formalise these methods due to their complete disalignment with Equation (3).

2.2 Minimal Graph Counterfactual Explanations

Considering the variety of the proposed GCE definitions in the literature, it is necessary to define a comprehensive one which, on the one hand, encompasses all of them and, on the other, ensures their correctness and provides enough flexibility to embed future definitions. First, we present a general setting of our GCE definition considering the instance classification problem (see Definition 2.1).

Definition 2.1 (Multi-class minimal counterfactual examples). Let Φ be a prediction model that classifies x into a class $c \in C$ from a set of classes C. Let X' be the set of possible counterfactual examples x' and $\mathcal{S}_{inst}(x, x')$ be a similarity measure that tells how similar x' is to x. Then, we define the set of counterfactual examples w.r.t. Φ as follows:

$$\begin{aligned} s(c', x) &:= \max_{x' \in X', x \neq x'} \{\mathcal{S}_{inst}(x, x') \mid \Phi(x') = c'\} \\ \mathcal{E}_\Phi(x) &= \bigcup_{c' \in C - \{c\}} \{x' \in X' \mid x \neq x', \mathcal{S}_{inst}(x, x') = s(c', x)\}. \end{aligned} \tag{12}$$

According to Equation (12), $\mathcal{E}_\Phi(x)$ has the maximally similar counterfactual examples $x' \in X'$ to the original graph x for each class c' that is different to the prediction on x (i.e., $c' \in C - \{c\}$ where $c = \Phi(x)$). More specifically, we find those counterfactual examples that must be different from the original instance x such that they maximise a similarity function w.r.t. the original instance $\mathcal{S}_{inst}(x, x')$. Accordingly, we can refer to the counterfactual examples of a specific class $c' \in C - \{c\}$ as $\mathcal{E}_{c',\Phi}(x) = \{x' \in X' \mid x \neq x', \mathcal{S}_{inst}(x, x') = s(c', x)\}$.

The definition above has two main advantages w.r.t. the ones provided in the literature: i.e., (1) it supports all the graph-based tasks in a multi-class scenario, and (2) it contains all the minimal counterfactual examples for each class $c \in C$. Furthermore, differently from the majority of the formalisations in Section 2.1, we found that using a similarity function $\mathcal{S}_{inst}(x, x')$ is more beneficial because of its flexibility since it might consider both the attributes and the structure of

the graph (i.e vertices, vertex/edge attributes, and edges). Finally, we constrain the counterfactual example to be different from the original instance (i.e., $x' \neq x$).

Definition 2.2 (Global minimal counterfactual example). Let Φ be a prediction model that classifies x into a class $c \in C$. Let X' is the set that contains all the possible counterfactual examples x'. We define the global minimal counterfactual example $\mathcal{E}^*_\Phi(x)$ of x, as follows:

$$\mathcal{E}^*_\Phi(x) = \arg\max_{x' \in X'} \mathcal{S}_{inst}(x, x'). \tag{13}$$

As anticipated at the beginning of this section, we can extend Definition 2.1 to consider also vertex and edge classification tasks. The only component that changes is the prediction model Φ. Recall that we denote with x an instance now it can be a vertex or an edge belonging to the original graph G. In this way, the prediction model Φ takes in input the instance x and G to produce a class c (i.e., $\Phi(x, G) = c$). Hence, for a particular counterfactual example graph G', $\Phi(x, G') = c'$. It is advisable that the generated counterfactual $G' = (V', E')$ contains the original instance x: i.e., $x \in V'$ for vertex prediction and $x \in E'$ for edge prediction. In this way, it is possible to understand how x's relations (vertices or edges) in its vicinity have changed in G' w.r.t. G.

3 METHODS

Here, we present the GCE methods present in the literature gathered following this process. We collected the works from Google Scholar on a bimonthly basis from September 2021 according to

intitle:*counterfactual* AND
[(intitle:*graphs* OR intitle:*graph* OR intitle:*gnn* OR intitle:*drug* OR intitle:*molecular* OR intitle:*molecules*)
AND (intitle:*explainable* OR intitle:*explanations* OR intitle:*explaining* OR intitle:*explainer*)]

The searches produced a total of 21 results, which we reviewed according to the following inclusion/exclusion criteria: we kept only works on graphs as input data evaluating their quality and their in-scope; we excluded Theses and non-original (reproducibility) articles; we excluded works that treat agents' explainability. After the selection, we obtained an initial set of 10 papers. We examined the papers cited and the papers which cite this initial set of papers according to the previously defined inclusion/exclusion criteria. We also monitored the literature with google scholar alerts. Finally, we collected fourteen papers (fifteen methods) [1, 4, 12, 13, 28, 36, 39, 41, 50, 53, 68, 69, 79, 82] that are at the base of this survey.

In the following, we show the gathered methods organised within the GCE taxonomy that we identified (Section 3.1). In Section 3.2, we summarise all the works in Table 2 accordingly to the ten chosen dimensions that we present. Finally, in Section 3.3, we discuss the methods by describing them and showing how they fit in each dimension.

3.1 Graph Counterfactual Explainability Taxonomy

Upfront, we need to clarify the main differences between graph counterfactuality and that on other data structures. First, GCE methods consider the graph structure when generating counterfactuals. This means that changes to the structure (e.g., edge additions/removals) can be used to generate the explanation. Generally, CE methods are strongly dependent on the type of data they have to explain (e.g., tabular data or images). Thus, Applying an explainer not specifically designed for graphs would not be able to exploit the graph structure properly. For instance, a tabular explainer might modify all the edges. Likewise, an image explainer might modify solely edges between contiguous vertices in the graph's adjacency matrix (due to the fact that vertices do not have a fixed order by definition). Second, graphs are highly complex, with many vertices, edges, and

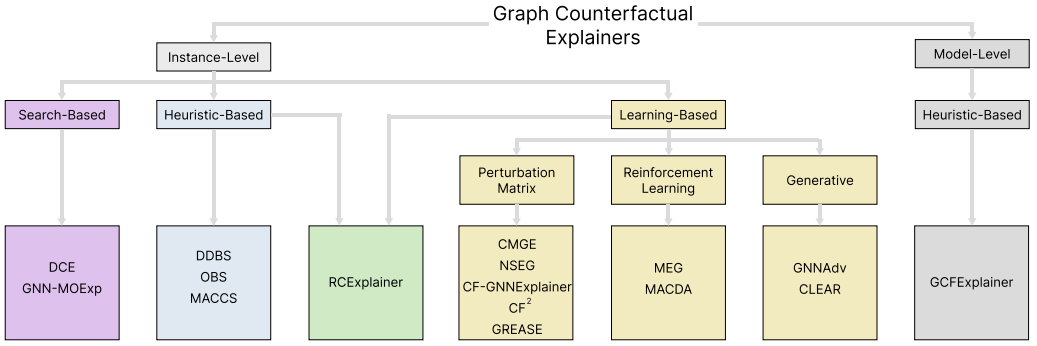


Fig. 2. Taxonomy of GCE methods.

attributes associated with both. This specific complexity poses different challenges for generating counterfactuals compared to other data structures. Additionally, interpreting the explanations is a challenging task requiring expertise due to the wide range of domain applications. Finally, when providing counterfactuals for graphs, one should rely on graph-based explainers (e.g., GNNs, GCNs, GATs) that can extract/embed features from neighbouring vertices (see Section A). Thus, having a specialised explainer is fundamental, as it happened with the specialisation of neural networks when the fully connected network (successfully applied to tabular data) evolved in the convolutional one (specialised for images data) and lastly was evolved in the GNN (specialised for graphs data).

Here, we describe the taxonomy of GCE methods - see Figure 2 shaped by considering [2, 3, 23, 73, 88] which, on the other hand, were focused on counterfactual explainability in general. The first level analyse the scope of the GC explainers by dividing them into instance-level and model-level. The former focuses on providing the reasons that make a black-box model reach a specific decision on a particular input instance. Whereas the latter entails producing an explanation of the overall logic of the black-box model assuming that the provided explanation is complete and valid for any instance. In other words, model-level explainability on graphs engages in providing the boundaries of the decision space of explained outcomes. Therefore, having a group of graph instances, model-level explainers aim at determining their collocation in the decision space.

The following level discusses the high-level approach adopted by the explainers dividing them into search-based, heuristic-based, and perturbation methods. Notice that each of these classes are not mutually exclusive (see RCExplainer [4]) meaning that methods can merge two different strategies into a single one to produce counterfactuals. Search-based methods rely on a specific criterion (e.g., similarity between instances) to search for a counterfactual within the dataset for a given input instance [21, 36]. Heuristic-based methods rely on a rigorous policy of modifying the input graph until they reach a valid counterfactual [1, 79]. Without loss of generality, learning methods study the output variations w.r.t. input changes. All methods follow a similar high-level pipeline consisting of (1) generating masks that indicate features of interest given a specific input graph G; (2) combining the mask with G to obtain a new graph G' such that the features of interest remain unchanged; (3) feeding G' to the prediction model Φ and updating the mask according to the outcome $\Phi(G')$. We have identified three sub-classes of learning-based methods: i.e., perturbation matrix [12, 39, 69, 82], reinforcement learning [50, 53], and generative [41, 68]. Perturbation matrix methods take the desired counterfactual class, the input instance, and the oracle weights and learn a soft-mask that represents which graph features should be present and which removed to get a counterfactual instance. Reinforcement learning methods employ agents that take actions to produce counterfactuals that maximise a user-defined cumulative reward function. Generally, this

Table 2. Comparison of GCE Methods

Method	Model Agnosticism	Model Access	Factual-Based Explanations	Minimal CE	Domain Agnosticism	Training Data Accessibility	Explanation Level	Classification Task	Generation Type	Approach
DDBS [1]	✓	·	·	✓	✓	✓	Instance	G	$E(+,-)$	Heuristic
OBS [1]	✓	·	·	✓	✓	·	Instance	G	$E(+,-)$	Heuristic
RCExplainer [4]	✓	✓	✓	·	✓	✓	Instance	G, V	$E(-)$	Heuristic & Learning
GNN-MOExp [36]	✓	·	✓	·	✓	·	Instance	V	sub-graph	Search
MEG [53]	✓	✓	·	✓	·	~	Instance	G	$E(+,-), V(+,-)$	Learning
GNNAdv [68]	✓	✓	·	✓	✓	·	Instance	G	$E(+,-)$	Learning
CMGE [82]	·	✓	✓	·	·	✓	Instance	G	$E(+,-), V(-)$	Learning
NSEG [12]	✓	✓	✓	·	✓	·	Instance	G, V	$E(-), F(*)$	Learning
CF-GNNExplainer [39]	✓	✓	·	✓	✓	·	Instance	V	$E(-)$	Learning
CLEAR [41]	✓	·	·	✓	✓	✓	Instance	G	$E(+,-), F(*)$	Learning
MACDA [50]	✓	·	·	✓	·	~	Instance	(G_1, G_2)	$E(+,-), V(+,-)$	Learning
CF^2 [69]	✓	·	✓	·	✓	·	Instance	G, V	$E(-), V(-), F(-)$	Learning
MACCS [79]	✓	·	·	✓	·	·	Instance	G	$E(+,-), V(+,-)$	Heuristic
GREASE [13]	✓	·	✓	✓	·	·	Instance	E	$E(-)$	Learning
GCFExplainer [28]	✓	·	·	·	✓	·	Model	G	$E(+,-), V(+,-)$	Heuristic

· depicts a missing aspect; ✓depicts a covered aspect; ~ depicts a partially covered aspect; E, V, and F indicate edge, vertex, and vertex/edge attribute sets, respectively; + and − indicate, respectively, adding and removal operations over the set that precedes them; $*$ indicates perturbation over the set it precedes.

kind of approach fixes the number of possible actions that the agent can take that lead towards a valid counterfactual (e.g., see MEG [53]) in a well-defined domain. Generative methods [41, 68] sample counterfactuals from a learned latent space.

Lastly, since model-level explainers are recent (see GCFExplainer [28]), there is only one class that is currently explored: i.e., heuristic-based. Hence, we invite researchers to explore this class of explainers due to its promising future directions (see Section 7). Note that almost all the methods perform a generation based on perturbing the input instance. Thus, in the taxonomy, we highlight the approach employed by each method, and this must not be confounded with the methods' properties that we discuss in the following section.

3.2 GCE Literature Classification

Here, we provide the reader with a classification of the GCE methods by identifying more dimensions w.r.t. the other surveys [7, 23, 25], as some of these are related to the graph domain. For each dimension, we provide a brief description and then delve into further detail (see Section 3.3). Table 2 reports the methods characterised within the chosen dimensions.

Model Agnosticism - A GCE method relies on a black-box model Φ to generate counterfactual examples by maximising Equations (12) or (13). Notice that the desiderata for a counterfactual explainer is to decouple the process of producing counterfactuals G' over an input instance G from the prediction $\Phi(G')$ s.t. $\Phi(G) \neq \Phi(G')$. In other words, model-agnostic counterfactual explainers can be used to explain the outcome of any prediction model Φ. Contrarily, model-specific explainers are intertwined on a particular class of prediction models (e.g., attention-based methods).

Model Access - GCE methods might require different levels of access to the underlying oracle Φ. According to [73], model access can be embedding- and gradient-wise. Embedding-wise access suggests that the explainer can get the embedding of graph G at a specific layer of Φ. Gradient-wise access consists of obtaining the gradients of G at any layer of Φ. Contrarily to gradient-wise, embedding-wise access does not restrict the explainer to rely only on neural network oracles.

Factual-based explanations - Some GCE methods use a factual explainer to generate a factual example of G producing the most important features w.r.t. the prediction $\Phi(G)$. Then, it is possible to build a counterfactual example G' by changing the most important features that change the prediction $\Phi(G')$. Notice that these explainers do not guarantee to produce a minimal counterfactual because they do not use an optimisation function to minimise the distance between the generated example G' and the input G. Figure 3 illustrates three different explanations. On the left

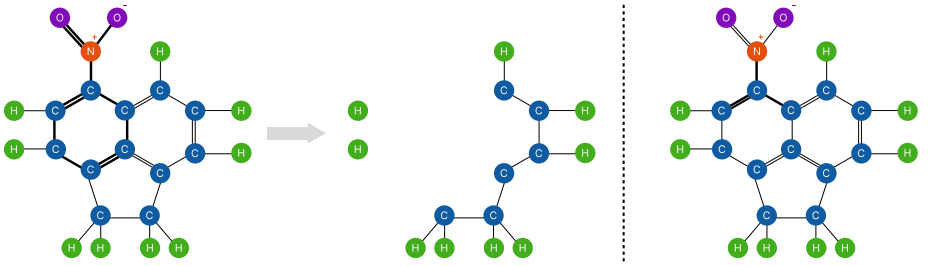


Fig. 3. An example of a 5-Nitroacenaphthene molecular structure. (left) Factual example, represented as bold connections, as a Nitrobenzene molecule. (middle) Derived counterfactual example by eliminating the Nitrobenzene molecule. (right) Minimal counterfactual example highlighting the three connections that, if removed, break the Nitrobenzene structure avoiding the molecule being a carcinogen.

corner, we depict a factual example - highlighted connections - representing the most important characteristics of the Nitrobenzene molecule. The middle molecular structure illustrates a derived counterfactual example from the factual one on the left. It is not guaranteed that the derived counterfactual example entails a valid molecule. Without loss of generality, derived counterfactuals can only be produced if the original graph was perturbed by eliminating vertices/edges (see Section 3). For completeness purposes, we illustrate the non-minimality of factual-based counterfactual explainers (compare the molecule in the middle with the one on the right). Here, we can clearly see that the minimal counterfactual has the three connections highlighted that, if removed, would break the Nitrobenzene structure shown on the left, thus, avoiding it to be a carcinogen.

Minimal Counterfactual Example (MCE) - According to [23], generating counterfactuals with minimal changes from the input is paramount. Providing a MCE is useful when giving the end-user tangible explanations. Recall the loan approval example with counterfactual *if the customer were a millionaire, then her loan would be granted.* Notice that this particular example is a useless counterfactual since it does not provide a meaningful explanation. Minimal counterfactual examples ensure that the change in the outcome is guaranteed when the structure of the input slightly differs. In the loan approval example, a minimal counterfactual explanation would be *if the customer had $1,000 more in her bank account, then her loan would be granted*, and the minimality is represented as the additional $1,000.

Domain Agnosticism - Domain-agnostic counterfactual explainers are transferable from one domain to the other. In particular, this kind of explainer can work with heterogeneous graph data (e.g., passing from molecular graphs to human social networks). Contrarily, domain-specific explainers are specialised in generating counterfactuals on only a subclass of graph data (e.g., protein-protein interactions). These explainers, when given in input a graph from a different domain, fail to generalise or even output a valid counterfactual example. The desiderata for robust counterfactual explainers is to be domain-agnostic and decouple from properties valid only in specific cases.

Training Data Accessibility - GCE methods can access training data to generate meaningful examples or be oblivious and rely only on the prediction $\Phi(G)$ while maximising the similarity between G' and G. When trying to preserve privacy in critical domains, accessing further data is not feasible. In these scenarios, we can estimate how the prediction model works by randomly perturbating G to produce a new graph G', and verify whether $\Phi(G') \neq \Phi(G)$. In this way, it is possible to construct an ad-hoc synthetic dataset of graph instances and understand Φ's functioning while maintaining data privacy. Hence, being oblivious of any additional information besides G is desired for GCE methods.

Explanation Level - It refers to the granularity of the explanation. Most explainers are focused on explaining the decision of Φ for a single instance G. On the other hand, model-level [28] or multiple single-instance explanation methods [59] provide counterfactuals for a set of input instances to have a high-level understanding of Φ's decision boundaries.
Classification Task - There are three classification tasks in graph learning: i.e., vertex, edge, and graph classification. In the literature, graph classification is the most common task, while edge classification has only been explored in [13]. Recently, Liu et al. [36] provide explanations for graph-pair affinity prediction tasks. In other words, given a drug-target (G_1, G_2) instance, the task is to generate counterfactual (G_1', G_2') for both the drug and the target graphs such that $\Phi(G_1, G_2) \neq \Phi(G_1', G_2')$.
Generation Type - The literature focuses on two main strategies of generating counterfactual examples: i.e., search-based and perturbation-based generation. Search-based methods find the most similar instance G' from the original dataset[6] w.r.t. the input graph G s.t. $\Phi(G) \neq \Phi(G')$ without guarantee that the produced explanation is minimal. Contrarily, perturbation-based approaches perturb G to generate G'. Many perturbation-based methods minimise an optimisation function, trying to produce a minimal counterfactual explanation. Perturbations can happen on vertices and edges by either adding or removing them (denoted in the table by + and - respectively) and by changing vertex/edge attributes (denoted by $F(*)$ in the table).
Approach - We report the high-level approach adopted by the method as in the taxonomy Section 3.1 included in the Table 2 for convenience. While in the future description of each method, we did not report it explicitly to avoid redundancy since the method will be thoughtfully described.

3.3 GCE Methods

Here, we provide a detailed description of the methods in GCE and a discussion on how they cope with the dimensions provided in Table 2. In particular, we first describe the method and how it produces counterfactuals. Then, we provide, for each dimension, its advantages and limitations. We report the methods according to their publication year breaking ties according to the alphabetical order of the authors' surnames.
DDBS and **OBS** [1] are two heuristic explanation methods designed for brain networks. The brain can be represented as a graph where the vertices represent well-established regions of interest (ROIs) and the edges are links between two co-activated ROIs. Both methods rely on a bidirectional search heuristic. First, they perturb the edges of the input G until they reach a counterfactual G'. Second, they rollback some perturbations done in the first stage such that the distance between G and G' decreases while maintaining the counterfactuality condition. OBS chooses to perturb the edges randomly, whereas DBS queries the dataset for the most common edges for each class and uses this information for perturbation purposes. (1) *Model agnosticism and model access*: Both methods do not access the oracle since they only need to know whether the produced graph G' is a valid counterfactual. Furthermore, since they treat the oracle as a black-box model, they work with any kind of oracle. (2) *Factual-based explanations and minimal CE*: OBS and DBS do not rely on any existing factual explanation method. Both methods focus on finding minimal counterfactuals since, in their second stage, they try to minimise the distance between G and the produced counterfactual. Notice that, given that these methods are heuristics, it is not guaranteed to converge to a minimal counterfactual example. (3) *Domain agnosticism and training data accessibility*: Both methods are designed for adaptability across various domains. However, their original implementation assumes binary classification and uniform data instances with the same vertices throughout the dataset.

[6]Having access to either the training data or the embeddings/gradients of the model Φ, makes them unfeasible for privacy preservation.

Fixing the first limitation is feasible with minor tweaks, but the second requires significant changes, affecting performance. OBS relies solely on the input graph G, while DDBS needs labelled instances to extract common edges for each class.

(4) *Classification task and generation type*: Both methods rely on oracles for graph classification tasks to generate counterfactuals via edge addition/removal operations. These edge perturbations are carried on for a user-defined number of times (i.e., upper-bound).

RCExplainer [4] generates robust counterfactuals. Firstly, it devises a set of decision regions containing linear decision boundaries obtained by an underlying GNN. RCExplainer exploits an unsupervised approach to find the decision regions for each class. In this way, the linear boundaries of these regions capture the commonalities of the graph instances therein, thus avoiding overfitting due to possible noises/peculiarities of specific instances. Therefore, the produced counterfactuals are robust to noise. Secondly, RCExplainer uses a loss function based on the obtained linear boundaries to train a network that produces a small subset of the edges E^* of the original graph G. RCExplainer ensures that the graph $G^* = (V^*, E^*)$ induced by E^* lies on the same decision region as G (i.e., have the same class). Therefore, $G' = G - G^*$ (i.e., removing the edges E^* from G) should lie outside this decision region. Hence, G' can be considered a counterfactual. (1) *Model agnosticism and model access*: RCExplainer assumes that the oracle is a GCN for it to access its last convolutional layer to obtain a vector representation of the input graph and, thus, represent the oracle's decision regions/boundaries. However, RCExplainer is adaptable to work with other types of oracles that provide dense graph representations. (2) *Factual-based explanations and minimal CE*: RCExplainer finds a subset of edges E^* such that $G^* = (V^*, E^*)$ induced on E^* is a factual explanation of the input G. Consequently, $G' = G - G^*$ is a counterfactual example of G. As noticed with CF-GNNExplainer, only removing edges from the input graph does not necessarily produce minimal counterfactuals. (3) *Domain agnosticism and training data accessibility*: The explanation method is not dependent of specific domain knowledge and can be used in diverse application domains. RCExplainer needs a collection of labelled instances to determine the decision regions governing each of the classes predicted by the oracle. (4) *Classification task and generation type*: RCExplainer explains the oracles' decision on graph classification tasks by removing the edges of G to produce G'. However, as with CF-GNNExplainer, RCExplainer does not guarantee to produce valid counterfactual explanations. For instance, assume that an oracle decides whether a graph has cycles or not. If an input G is a tree (i.e., it does not have cycles), RCExplainer would fail to produce a counterfactual G' that has cycles only by removing the edges of G. Notice that transforming a tree into a graph with cycles is only possible via edge additions.

GNN-MOExp [36] is a multi-objective factual-based explanation method for GNN predictions. Although a factual method, GNN-MOExp enforces counterfactual relevance to its factual explanation subgraphs (e.g., CF^2). It searches for a subgraph in the original instance that optimises both the factual and counterfactual features. GNN-MOExp comes with several limitations that limit the expressiveness of the produced counterfactual: i.e., the factual subgraphs are required to be acyclic; the explanation size is specified a priori; it restricts counterfactuals to be subgraphs of the factual ones. (1) *Model agnosticism and model access*: GNN-MOExp assumes the oracles are black boxes and does not require access to their internal representations. (2) *Factual-based explanations and minimal CE*: GNN-MOExp is a factual explanation method. It encourages the generation of small factual explanations. However, this is not enough to optimise towards minimal GCEs. Furthermore, it is affected by its impossibility to generate GCEs containing cycles. (3) *Domain agnosticism and training data accessibility*: The method does not make any particular assumption about the dataset and does not perform training. (4) *Classification task and generation type*: GNN-MOExp focuses on oracles for vertex classification. For this reason, it searches for the factual explanation subgraphs in

the neighbourhood of the current vertex, ranking them according tothe desired metrics. However, this generation mechanism cannot perform edge/vertex additions.

MEG [53] is a reinforcement learning approach that produces counterfactuals for a given input molecule. The reward function of the procedure incorporates a task-dependent regularisation term that affects the policy of choosing the next action to perturb the input. Since MEG strives to produce valid counterfactuals, its policy is designed to only chose those actions that lead towards the generation of "*new*" valid molecules [90]. (1) *Model agnosticism and model access*: MEG accesses the vector representation of molecules used by the oracle to calculate the similarity between instances (i.e., input and produced molecules). Since MEG's original formulation requires its underlying oracle to be a GNN, this method is not model agnostic. However, one can argue, as done with RCExplainer, that MEG can use any type of oracle that can provide learned representations of molecules. (2) *Factual-based explanations and minimal CE*: MEG perturbs the input molecule directly to produce valid counterfactuals, hence, it is not a factual-based approach. Since MEG employs a policy to choose the most promising action to perform on the input molecule, this policy needs to be calibrated according to a reward function that maximises the similarity between the input instance and the produced counterfactual. Thus, MEG produces a minimal counterfactual example. (3) *Domain agnosticism and training data accessibility*: MEG is designed for the molecular domain and enforces that the graphs are valid molecules. It uses domain knowledge to provide meaningful molecular counterfactuals, thus losing its adaptation flexibility in other domains. Since MEG is a method that is trained/optimised per instance, then it requires access to all the instances one-by-one. However, notice that MEG does not use other instances in the dataset besides the current one to decide the next step to take w.r.t. the reward function that brings to the generation of a new, potentially counterfactual, molecule (hence the ∼ in Table 2). (4) *Classification task and generation type*: MEG provides explanations for oracles focusing on the graph classification task. However, the method can also be used to explain the decisions in regression tasks. Additionally, MEG is one of the few methods that employ additions and removals of both vertices and edges in its generation process.

GNNAdv [68] is the first strategy that generates counterfactual explanations via adversarial attacks. In particular, GNNAdv uses a **Topology Attribution Map** (**TAM**) defined with the help of two variables that summarise edge manipulations. Then, it optimises a sparsity-promoting problem over the perturbation variables and exploits a GNN oracle to optimise the TAM. Then, the TAM is sampled according to a Bernoulli distribution to produce the adjacency matrix of the counterfactual explanation.

(1) *Model agnosticism and model access*: GNNAdv can work with different models as long as they are GNNs. This limitation results from the explainer's need to access the gradients of the oracle to update the TAM. (2) *Factual-based explanations and minimal CE*: GNNAdv relies on a reward function that encourages smaller explanations without guaranteeing minimality. (3) *Domain agnosticism and training data accessibility*: The explainer does not need direct access to training data. It has also been tested in datasets from different domains. (4) *Classification task and generation type*: GNNAdv works on graph classification tasks and generates counterfactuals by only adding/removing edges from the TAM.

CMGE [82] works only for **Electronic Health Records** (**EHRs**). First, the authors transform the medical data into a hierarchical graph structure that encodes the relationship between the different types of records. Then, they train a learnable soft-mask matrix to mask the features of vertices/edges in the graph while keeping the decision unaltered. The remaining features - i.e., those that have not been masked - can be considered as supportive to the decision representing a particular health diagnosis. The authors rely on **Graph Attention Networks** (**GATs**) as the explainer of their approach despite the fact that they are not intended as interpretable models [29].

When generating counterfactual examples, the authors focus on graph classification. (1) *Model agnosticism and model access*: Because CMGE relies on the attention mechanism as an explanation method, the explainer and the oracle cannot be decoupled, thus becoming an oracle-specific explanation method. Additionally, CMGE requires complete access to the model logic to verify the attention weights of each feature. (2) *Factual-based explanations and minimal CE*: Despite the authors claiming to do counterfactual reasoning, CMGE is a perturbation-based factual explanation method. Here, by relying on the attention mechanism, the authors return the top-k most important features, which is not guaranteed to be the smallest possible set to engender counterfactuals (minimality violation). (3) *Domain agnosticism and training data accessibility*: The authors analyse CMGE's performances only on EHRs. However, we argue that GATs can be adopted to other domains as long as the input data can be represented as graphs. Thus, CMGE is partially domain agnostic. Additionally, CMGE is oracle-specific since it relies on the attention mechanism used by the oracle. Notice that attention mechanisms need to access the network's weights to produce the contribution scores for each input feature. Hence, CMGE accesses the training data to produce counterfactuals. (4) *Classification task and generation type*: The authors produce counterfactuals via vertex/edge perturbations for graph classification. Recall that CMGE is a factual-based counterfactual explainability method. Therefore, for vertex perturbations on G, the authors can only remove vertices because their goal is to find a sub-graph G^* s.t. $\Phi(G) = \Phi(G^*)$. Then, by eliminating G^* from G, the authors produce the (possibly) counterfactual $G' = G - G^*$. For edge perturbations, the authors use both adding and removal operations.

NSEG [12] generates necessary and sufficient generations in a similar manner as CF^2. While CF^2 heuristically determines a tradeoff between the probability of necessity and that of sufficiency, NSEG maximises their joint lower bound. For this reason, NSEG leverages a continuous mask with a specific sampling strategy, thus producing a discrete adjacency matrix to optimise this lower bound. (1) *Model agnosticism and model access*: NSEG can work with different explainers. However, it needs to access the gradients of the overall loss w.r.t. to the masks because it uses them to update the masks and obtain the necessary and sufficient explanation. (2) *Factual-based explanations and minimal CE*: NSEG is primarily a factual explanation method that can be used to generate counterfactual explanations, given its emphasis on obtaining explanations with a high probability of necessity. This kind of approach, in general, cannot generate minimal counterfactual explanations. (3) *Domain agnosticism and training data accessibility*: The explainer is not tied to perform in any specific domain and does not requires access to training data. (4) *Classification task and generation type*: NSEG can be used with oracles targeting both graph and vertex classification tasks. The perturbation method used to generate the mask is based on [86], so NSEG can only remove edges. However, for vertex features, it performs feature value permutation values by only considering the values of the features that exist in other vertices in the input graph.

CF-GNNExplainer [39] finds a binary perturbation matrix that sparsifies the adjacency matrix of G. To find the perturbation matrix, the authors rely on [67] to train sparse neural networks. Additionally, the authors' goal is to remove edges by zeroing the adjacency matrix. CF-GNNExplainer produces counterfactual examples whose distance is the smallest w.r.t. G. (1) *Model agnosticism and model access*: CF-GNNExplainer is model-agnostic and can be used with any oracle. In this way, the explainer does not access any information regarding the inner workings of the oracle. Instead, it just relies on the input instance and the oracle's decision. (2) *Factual-based explanations and minimal CE*: CF-GNNExplainer is not based on existing factual explanation techniques. Moreover, it is designed to tackle the minimal GCE problem, returning, from the generated counterfactual examples, the closest one to the input instance. However, notice that it is not always possible to find the minimal counterfactual $\mathcal{E}^*_\Phi(G)$ due to the limited perturbation operations performed on G. (3) *Domain agnosticism and training data accessibility*: CF-GNNExplainer does not

use any domain knowledge. Hence, it can be employed on different application domains. Additionally, it does not require labelled data for training purposes. (4) *Classification task and generation type*: CF-GNNExplainer produces counterfactuals for oracles trained for the vertex classification task. Hence, the explainer gets a vertex x in input and its ego-network. While other methods of this kind perturb the vertex features, CF-GNNExplainer perturbs x's ego-network until Φ changes decision. In other words, CF-GNNExplainer strives to change the relationships between instances (e.g., connected vertices) rather than the instances themselves. CF-GNNExplainer can only remove edges from the original graph which is its main limitation since it is not always possible to obtain a valid counterfactual.

CLEAR [41] is a generative GCE method. It relies on a **variational autoencoder** (**VAE**) where the encoder maps each input graph G into a latent representation Z, and the decoder generates the counterfactual based on Z. The counterfactuals are complete graphs with stochastic weights on the edges[7] where the vertex features and graph structure are similar to G. The generation of the counterfactuals is conditioned on G and a desired class $c \neq \Phi(G)$. It is important to highlight that, during the decoding process of generating the counterfactual G', the order of the vertices of G is not the same as that in G'. Thus, a graph matching step between G and G' is necessary. (1) *Model agnosticism and model access*: CLEAR does not require access to the oracle internal representation and can be used with different kinds of oracles. (2) *Factual-based explanations and minimal CE*: The method is not a factual-based explanation method. CLEAR's loss function enables the generation of counterfactual graphs that are the closest to the original instance. (3) *Domain agnosticism and training data accessibility*: The method is not domain specific and was tested in social networks and molecular datasets. CLEAR requires having access to training data and performing gradient descent on it to learn how to generate counterfactuals on unseen graphs. (4) *Classification task and generation type*: CLEAR is designed to explain oracle predictions on graph classification tasks. To generate the explanations, the method performs edge removals/additions and feature value perturbations.

MACDA [50] produces counterfactuals consisting of drug-target pairs for the drug-target affinity (DTA) prediction task relying on multi-agent reinforcement learning. Given in input a pair of instances (G_1, G_2), where G_1 is a drug and G_2 is a target graph, MACDA generates the counterfactual pair (G_1', G_2') such that G_1' is a counterfactual to G_1 and G_2' to G_2 (i.e., $\Phi(G_1) \neq \Phi(G_1') \ \wedge \ \Phi(G_2) \neq \Phi(G_2')$). Notice that the counterfactual pair is produced simultaneously. (1) *Model agnosticism and model access*: The method assumes the oracle is a black-box and only accesses its output. (2) *Factual-based explanations and minimal CE*: MACDA is designed to generate counterfactual explanations, and the loss function of the agents encourages the method to produce counterfactuals close to the original instance. (3) *Domain agnosticism and training data accessibility*: The framework is designed to work with molecular graphs and specifically for the drug-target affinity prediction task, not easily adaptable to other domains. As in MEG, MACDA needs to generate the next possible actions w.r.t. a specific reward function for each instance. Therefore, it does not need to access training data to generate a counterfactual for the current instance G, but it optimises the actions to take for G specifically. (4) *Classification task and generation type*: The DTA prediction task is a classification task that consists of predicting the strength (binding affinity) of a drug molecule and a target protein. Usually, the binding affinity is categorised as zero, low, medium, and high. As with MEG, MACDA adds/removes both vertices and edges in its generation process.

CF^2 [69] produces factual explanations by balancing factual and counterfactual reasoning. It solves a multi-objective optimisation problem where the generated counterfactuals need to

[7]CLEAR cuts edges according to a Bernoulli distribution over the stochastic adjacency matrix to produce a counterfactual graph.

adhere to specific constraints. According to the proposed factual reasoning, the factual graph is a subgraph of the input instance. Hence, the counterfactual explanation is the input graph without the factual subgraph, as in the case of RCExplainer. CF^2 takes into consideration the simplicity of the counterfactual (i.e., the smaller the explanation size, the better). Despite being a factual method, we include CF^2 because it inherently has a counterfactual property of eliminating the factual subgraph. (1) *Model agnosticism and model access*: CF^2 does not access the oracle's latent representations to produce counterfactuals, and it can be used with any kind of oracle. (2) *Factual-based explanations and minimal CE*: CF^2 is a perturbation-based factual method. However, it requires the explanations to comply with the counterfactual property, meaning that removing the factual subgraph from the original graph must produce a change in the prediction. As mentioned for RCExplainer, subtracting a factual subgraph from the original graph does not guarantee producing a minimal counterfactual explanation. (3) *Domain agnosticism and training data accessibility*: The authors of CF^2 provide experimental results demonstrating the effectiveness of their method on synthetic, citation, and molecular datasets. In this way, CF^2 is adaptable to any domain as long as it encompasses graph data. Additionally, the explainer does not require labelled data to produce a counterfactual example since the explanation produced is a mere derivation of the removal of the factual subgraph from the original instance. (4) *Classification task and generation type*: CF^2 can to explain oracle predictions on graph and vertex classification tasks. It does edges, vertex, and vertex attribute removals on the original graph to generate counterfactuals. A limitation of this approach is that edge and vertex additions are not considered because the factual explanations are a subset of the input graph, while, in general, counterfactuals are not.

MACCS [79] works in the molecular domain. The method takes in input a molecular graph represented as ***SELF*-referenc*I*ng *E*mbedded *S*trings (SELFIES)** [32]. MACCS builds on top of the STONED protocol [51], which rapidly explores the chemical space without relying on pre-trained generative models or reaction rules. The STONED protocol consists of string insertion, deletion, and modification steps that can generate valid perturbed molecules that are close in the chemical space w.r.t. the input molecule. After expanding the chemical space around the original molecular graph, MACCS identifies similar counterfactuals with a changed prediction, selecting a small number of these using clustering and the Tanimoto similarity. By clustering the counterfactual examples and selecting, for each cluster, the closest counterfactual to the original molecule, MACCS returns multiple counterfactuals that are different from each other. (1) *Model agnosticism and model access*: MACCS is model agnostic. Neither the STONED method nor the clustering/similarity processes need access to the oracle's internal information to produce counterfactuals. (2) *Factual-based explanations and minimal CE*: MACCS is not based on factual explanation approaches. The explainer returns multiple counterfactual examples ordered by their proximity w.r.t. the original molecular graph. Furthermore, MACCS has an upper-bound on the modifications the counterfactual molecules can have to ensure they are located nearby in the chemical space. (3) *Domain agnosticism and training data accessibility*: By employing the STONED protocol, MACCS can generate counterfactuals without needing labelled training data. Contrarily, MACCS is domain-specific, being restricted to the molecular domain. This poses an important limitation to the adoption of MACCS on other domains since instances (molecules) are represented as SELFIES strings. Consequently, the mutations (i.e., changes in the molecular structure) are restricted to an alphabet specific to the molecular domain. (4) *Classification task and generation type*: MACCS is designed to explain the decisions of oracles on molecular graph classification. Since STONED can token deletions, replacements, and insertion on the SMILES representation on molecules, MACCS can, thus, modify vertices and edges on the original graph.

GREASE [13] is designed to explain user-item recommendations, which is equivalent to explaining the existence of edges in a bipartite graph composed by user and item vertices. The authors

train a surrogate model to find an optimal perturbation mask without accessing the original black-box model. (1) *Model agnosticism and model access*: Thanks to the surrogate model, no knowledge about the original model is necessary (i.e., no gradient access). (2) *Factual-based explanations and minimal CE*: The method performs factual and counterfactual explanation generation separately. However, using a common definition for both processes limits the counterfactual examples to be a subgraph of the original graph. The number of changes performed to the original instance to produce the counterfactual explanation is considered, thus encouraging minimal explanations. (3) *Domain agnosticism and training data accessibility*: GREASE is designed for explainability in recommender systems and not in general graph data. The method does not require training data. (4) *Classification task and generation type*: The classification task could be equivalent to edge classification but also takes into account the rankings used by the recommender system. The method only performs edge removals during the explanation generation process.

GCFExplainer [28] is a model-level GCE method. It takes in input a set of graphs $\mathcal{G}$ belonging to a class c, an oracle Φ, a distance boundary θ, and a budget k. It, then, produces a set of counterfactual graphs $\mathcal{G}'$ with $|\mathcal{G}'| = k$ s.t. it maximises the number of graphs $G \in \mathcal{G}$ that have a counterfactual $G' \in \mathcal{G}'$ within an edit distance lower than θ (see Equation (11)). In other words, GCFExplainer maximises the coverage of the explanation set. First, GCFExplainer organises the search space as a meta-graph where the vertices are all the graphs that can be obtained by performing no more than θ edits on any input G, and edges connect graphs that are at one edit distance. After building this search space, GCFExplainer relies on vertex-reinforced random walks [55] to obtain a set of counterfactuals prioritising those that most instances can reach. Lastly, GCFExplainer uses a greedy algorithm to select a set $\mathcal{G}'$ of size k that maximises $\mathcal{G}$'s coverage. (1) *Model agnosticism and model access*: GCFExplainer does not require access to the oracle internal representation. Furthermore, it can be used with different kinds of oracles as it only needs access to the prediction. (2) *Factual-based explanations and minimal CE*: GCFExplainer is specifically designed to produce counterfactual explanations and it is not based on any existing factual explainer. The method does not search for a minimal counterfactual explanation. However, it permits manually setting the maximum allowed edit distance θ to produce counterfactuals. (3) *Domain agnosticism and training data accessibility*: The method was tested only on molecular datasets. However, there are no domain-specific restrictions that prevent its application to different domains. The explainer does not perform any kind of training, thus, it does not need access to training data. (4) *Classification task and generation type*: GCFExplainer is designed to explain the oracle prediction on graph classification tasks. To generate the explanations, the method performs vertex and edge removals/additions.

4 EVALUATION

A fundamental aspect of successful research is to provide evidence of the effectiveness of the proposed solution. This evaluation is done through quantitative assessments, including standardised tests and various measurements, following a fixed experimental protocol. In the case of GCE methods, a general evaluation protocol involves the following steps. First, the oracle is trained on a selected dataset. Second, the explainer, to be evaluated, takes the trained oracle and a selected instance as inputs and produces an explanation. Third, the produced explanation and the runtime traces are evaluated using several evaluation measures. This process can be repeated for different datasets, explainers, and oracles, aggregating the resulting measures in tables and plots.

In Section 4.1, we describe the datasets adopted in the literature. We refer the reader to Section C for a detailed description of the characteristics of each of them. In Section 4.2, we summarise the evaluation metrics used in GCE and argue that multiple metrics should be used for fair performance evaluations. Lastly, Section 4.3 illustrates the evaluation protocol adopted in each SoA work.

Table 3. The Datasets used in the Literature Alongside their Domains, the Link to their Repository, and the Papers they are Used by

Dataset	Domain	Publicly Available Repository (Data or Code)	Used by
Tree-Cycles [86]	synthetic	https://github.com/RexYing/gnn-model-explainer	[4, 12, 39, 69]
Tree-Grid [86]	synthetic	https://github.com/RexYing/gnn-model-explainer	[4, 12, 39]
Tree-Infinity	synthetic	https://github.com/MarioTheOne/GRETEL	[60]
BA-Shapes [86]	synthetic	https://github.com/RexYing/gnn-model-explainer	[4, 12, 39, 69]
BA-Community [86]	synthetic	https://github.com/RexYing/gnn-model-explainer	[4]
BA-2motifs [40]	synthetic	https://github.com/flyingdoog/PGExplainer	[4, 69]
ADHD [10]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/adhd	[1]
ASD [16, 33]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/autism/asd	[1]
BBBP [45]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=bbbp.zip	[79]
HIV [17, 20, 63]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=hiv.zip	[28, 79]
Ogbg-molhiv [27]	molecular	https://huggingface.co/datasets/OGB/ogbg-molhiv	[41]
Mutagenicity [30]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/Mutagenicity.zip	[4, 28, 69]
NCI1 [75]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/NCI1.zip	[4, 28, 69]
TOX21 [31]	molecular	https://tripod.nih.gov/tox21/challenge/data.jsp	[53]
ESOL [83]	molecular	https://github.com/deepchem/deepchem	[53, 69]
Proteins [8]	molecular	https://chrsmrrs.github.io/datasets/docs/datasets/	[28]
Davis [18]	molecular	http://staff.cs.utu.fi/~aatapa/data/DrugTarget/	[50]
PDBBind [77]	molecular	http://www.pdbbind.org.cn/	[50]
CiteSeer [22]	social	https://linqs.org/datasets/	[36, 69]
IMDB-M [84]	social	https://virginia.app.box.com/s/941v9pwh83lfw5vnwfbgcertlsoivg5j	[41]
CORA [46]	social	https://relational.fit.cvut.cz/dataset/CORA	[36]
Musae-Facebook [64]	social	https://www.kaggle.com/datasets/rozemberczki/musae-facebook-pagepage-network	[36]
LastFM [65]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/lastfm	[13]
Yelp [78]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/yelp2018/	[13]

4.1 Datasets Adopted in the Literature

Due to the absence of standardised and well-established benchmarks in the literature, comparing the approaches presented in Section 3 is difficult. The surveyed studies typically compare themselves with simple baselines rather than SoA solutions by adopting a heterogeneous set of synthetically generated ad-hoc datasets not part of an established benchmark. We provide Table 3 as an index of available datasets to bridge this gap for future researchers. Thus, researchers can adopt them to evaluate and compare their performances with those proposed in the literature.

Synthetic datasets are structured and generated through specific algorithms[8] and constraints, making them easier to control and modify. They allow for easy evaluation of performance since properties are well-known. Additionally, they enable the identification of the minimum counterfactual example[9] for a given input instance and the computing of the minimal distance between the two. This approach can be applied to classification tasks such as distinguishing cyclic and acyclic graphs. The generation of synthetic datasets follows [86], and it involves three steps presented in its binary version for readability: (1) generate a base graph with specific characteristics, including the number of vertices and edge; (2) generate well-known motifs or use handcrafted ones if preferred, ensuring that the base graph does not already contain the motif to be added; (3) connect the chosen motifs to the base graph while controlling for additional similar motifs. The resulting dataset can have two classes, graphs generated solely by the first step labelled 0 and those following all three steps labelled 1. Repeat the second step for the desired number of classes for multi-class scenarios. Researchers can choose the number of instances and their distribution among classes. *Tree-Cycles* [86], *Tree-Grid* [86], *Tree-Infinity* [60], *BA-Shapes* [85], *BA-2motifs* [40], *BA-Community* [86] are the synthetic datasets adopted in the literature. Besides these synthetic datasets, the literature relies on real datasets. The real datasets can be divided into three domains:

[8]Usually, synthetic graph datasets are generated via programmed processes that rely on specific algorithms and constraints.
[9]Recall that the minimality of a counterfactual example does not strictly depend on the explainer. Rather, the oracle classifying the generated examples is important in considering them counterfactual. In cases where the explainer performs well, but the oracle lacks, the literature has proposed to assess the oracle's performance by measuring its accuracy.

Table 4. Evaluation Metrics used in the Literature

		DDBS & OBS [1]	RCExplainer [4]	GNN-MOExp [36]	MEG [53]	GNNAdv [68]	CMGE [82]	NSEG [12]	CF-GNNExplainer [39]	CLEAR [41]	MACDA [50]	CF^2 [69]	MACCS [79]	GREASE [13]	GCFExplainer [28]
	Runtime	·	✓	·	·	·	·	·	·	✓	·	·	·	·	✓
	Oracle Calls	✓	·	·	·	·	·	·	·	·	·	·	·	·	·
	Oracle Accuracy	·	·	·	·	·	·	·	·	·	·	·	·	·	✓
	Correctness	·	·	✓	·	·	·	✓	·	✓	·	✓	·	✓	·
	Sparsity	·	·	·	·	·	·	·	✓	·	·	·	·	·	·
	Fidelity	·	✓	·	·	·	·	✓	✓	·	·	·	·	·	·
	Robustness	·	✓	✓	·	·	·	·	·	·	·	·	·	·	·
	Explainer Accuracy	·	·	·	·	✓	✓	✓	✓	·	·	✓	·	·	·
	Prediction Distance	·	·	·	·	·	·	·	·	·	✓	·	·	·	·
	Causality	·	·	·	·	·	·	·	·	✓	·	·	·	·	·
	Diversity	·	·	·	·	·	·	·	·	·	·	·	·	·	·
	Actionability	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Minimality Evaluation	GED	✓	·	·	✓	·	·	·	·	✓	✓	·	✓	·	·
	Explanation Size	·	·	·	·	·	·	·	✓	·	·	·	·	✓	·
	Tanimoto Similarity	·	·	·	·	·	·	·	·	·	·	·	✓	·	·
	MEG Similarity	·	·	·	✓	·	·	·	·	·	·	·	·	·	·
Global Metrics	Recourse Cost	·	·	·	·	·	·	·	·	·	·	·	·	·	✓
	Coverage	·	·	·	·	·	·	·	·	·	·	·	·	·	✓
	Compactness	·	·	·	·	·	·	·	·	·	·	·	·	·	✓

✓indicates that the measure is used by that paper, · if it is not used.

i.e., -omics ([10, 16, 33]), molecular ([8, 17, 20, 27, 30, 31, 63, 75, 79, 83]), and social networks ([22, 84]). We point the reader to Section B for detailed descriptions of the datasets provided in Table 3 and their characteristics.

4.2 Evaluation Metrics

Table 4 summarises the metrics used in the literature to evaluate the performance of a GCE method. Since multiple factors can affect the quality of explanations, it is better to employ many simple metrics instead of a complex ones, such as fidelity, to fully understand the different facets of the explainer's behaviour. Moreover, for completeness purposes, we group some metrics according to two classes - i.e., Minimality Evaluation and Global Metrics - and report two ideal metrics (Diversity and Actionability) that are not used in the literature. We believe these metrics can be useful to uncover more insight about the performances of GCE.

Runtime measures the time taken by the explainer to produce the counterfactual example. It provides an efficient way of measuring the explainer's efficiency, including the execution time of the oracle. The runtime can discriminate a *good* explainer based on the sluggishness of its oracle. This measure must be performed in isolation on the same hardware and software platform.

Oracle Calls [1] measures the number of times the explainer must ask the oracle to produce a counterfactual. This metric is similar to the runtime, but it is useful to evaluate the explainer's computational complexity performances, especially in a distributed system, avoiding considering latency and throughput, which are critical exogenous factors of the measurement.

Oracle Accuracy evaluates the oracle's reliability in predicting outcomes. The accuracy of the employed oracle significantly influences the quality of explanations, as the explainer aims at elucidating the model's behaviour. Only accurate model predictions will result in meaningful explanations provided to the user. Mathematically, for a given input x and its true label y_x, accuracy is defined as $\chi(x) = \mathbb{1}[\Phi(x) = y_x]$.

Correctness (Validity) [23, 60] indicates whether the explainer is capable of producing a valid counterfactual explanation (i.e., the example has a different classification from the original instance). More formally, given the original instance x, the produced example x', and oracle Φ, the correctness is an indicator function $\Omega(x, x') = \mathbb{1}[\Phi(x) \neq \Phi(x')]$.

Correctness can also be referred as *Probability of Necessity* when adopted for multiple counterfactual explanations [69] for a given instance, as $\frac{\sum_{x' \in X'} \mathbb{1}[\Phi(x') \neq \Phi(x)]}{|X'|}$.

Sparsity [88] measures the similarity between the input instance and its counterfactual according to the input attributes. Recall that an instance x can be a vertex, edge, or graph, with its associated counterfactual x' and an attribute set of length $|x|$.

Assuming that $\mathcal{D}_{inst}(x, x')$ is the distance between x and x', we adapt the original definition of sparsity to be $\frac{\mathcal{D}_{inst}(x, x')}{|x|}$ for graphs.

Fidelity [88] measures how faithful the explanations are to the oracle considering their correctness. Given the input x, its true label y_x, and its counterfactual x', fidelity is defined as $\Psi(x, x') = \chi(x) - \mathbb{1}[\Phi(x') = y_x]$. Notice that $\Psi(x, x')$ can assume three values. A value of **1** entails that both the explainer and oracle are working correctly. It is trivial to verify that $\chi(x)$ needs to produce a 1 (i.e., $\Phi(x) = y_x$) and the indicator a 0 (i.e., $\Phi(x') \neq y_x$). **0** and **−1** describe something wrong with the explainer or the oracle. However, we cannot attribute the incorrect function to the explainer or the oracle. This is a shortcoming of fidelity since it bases the assessment of the correctness on the ground truth label y_x instead of the prediction $\Phi(x)$.

The **Robustness** [4] of an explanation quantifies its resistance to changes induced by adding noise to the input graph. A perturbed graph G_p is produced to measure robustness by adding random noise to the vertex features of G and randomly adding or deleting edges while ensuring that the oracle prediction remains unchanged. The extent of change in the explanations G' and G'_p is then computed. Bajaj et al. [4] consider the top k edges of G' as the ground truth and compare G'_p against them. They evaluate the robustness using AUC-ROC. It is worth noting that their approach assumes that the explainer assigns importance scores to the graph's edges. One can use different functions to assess robustness to calculate the similarity between G' and G'_p. The metric is also referred to as *Instability* [23].

Explainer Accuracy is the proportion of explanations that are "correct". Works focused on explaining node classifiers [38, 69] follow the evaluation idea used in [86] for nodes that are originally labelled as being part of a specific motif. Then, an explanation is considered correct if it only includes edges inside the motif. Accuracy can only be computed for instances where a ground truth explanation is known.

Prediction Distance [50] is the distance between the predicted outcome of x and that of x', i.e., $|\mathcal{D}_{pred}(\Phi(x), \Phi(x'))|$.

Causality [41] measures if the changes from x to x' are consistent with the underlying structural causal model that describes the causal relations among different variables (e.g., node features, degree) present in the data.

Diversity [23] quantifies how diverse a set of counterfactuals is, i.e., $Div(X') = \frac{1}{|X'|} \sum_{x' \in X'} \sum_{x'' \in X'} \mathcal{D}_{inst}(x', x'')$.

Actionability [23] accounts for the counterfactuals $x' \in X'$ that are feasible in practice for a specific instance x, given the set $\mathcal{A}$ of actionable features. Feasible counterfactuals are those that do not consider features (e.g., race, gender) outside of $\mathcal{A}$. Actionability is defined as $Act(x, X') = \frac{|\{x' \in X' \mid \mathbb{1}(\mathcal{A}(x', x))\}|}{|X'|}$ where $\mathbb{1}(\mathcal{A}(x', x))$ gives 1 if x' is actionable w.r.t. the list of actionable features $\mathcal{A}$, and 0, otherwise.

Minimality Evaluation studies the minimality of the counterfactual examples (cf. Def. 2.2). Notice that, to provide a minimal counterfactual example, one needs to define metrics of similarity/distances between the generated counterfactual G' and the input graph G. Here, we report each metric in this category (i.e., *Graph Edit Distance (GED)*, *Explanation Size*, *Tanimoto Similarity*, and *MEG Similarity*), and comment on them accordingly. **Graph Edit Distance (GED)** quantifies the structural distance between the original graph G and its counterfactual G'. The distance is evaluated based on a set of actions $\{p_1, p_2, \ldots, p_n\} \in \mathcal{P}(G, G')$, representing a path to transform G into G' over all possible paths of actions $\mathcal{P}(G, G')$. The path consists of adding or removing vertices or edges, and each action p_i in the path is associated with a cost $\omega(p_i)$. Given G, G', and the set of actions, the GED is computed as

$$\min_{\{p_1, \ldots, p_n\} \in \mathcal{P}(G, G')} \sum_{i=1}^{n} \omega(p_i)$$

Typically, when presented with two counterfactual examples, we prefer one closer to the original instance G, as it provides shorter action paths on G to change the oracle's output. Notice that GED yields a global measure, and a relative metric that considers the instance size, such as sparsity, can be employed to evaluate the explainer's performance over multiple instances. **Explanation Size** [39] depicts the difference between the original graph G and the counterfactual G'. Similarly to GED, it considers counterfactual explanation as a set of edit actions $\{p_1, \ldots, p_n\} \in \mathcal{P}(G, G')$ to perform on G to transform it into G'. The explanation size is equal to the number of edit actions, i.e., $|\{p_1, \ldots, p_n\}| = n$. **Tanimoto Similarity** [5] calculates the similarity between two molecule graphs represented as binary vectors. Hence, given $G = (V, B, E)$ and $G' = (V, B', E')$ s.t. $B, B' \in \{0,1\}^{|V|}$, $\tau(G, G') = \frac{\sum_{i=1}^{n} B_i \cdot B'_i}{\sum_{i=1}^{n} B_i + \sum_{i=1}^{n} B'_i - \sum_{i=1}^{n} B_i \cdot B'_i}$. Lastly, **MEG Similarity** [53] is a convex combination of $\tau(G, G')$ and the cosine similarity of the graphs G and G'.

Global Metrics measure the explainer's performance at the dataset level (global explanations). Thus, the metrics discussed here assume that there is a single explanation set X' for the entire dataset. **Recourse Cost** [28] evaluates global explanations considering the distance between the original instances $x \in X$ and the produced counterfactuals $x' \in X'$, i.e., $Cost(X') = f_{x \in X}(\min_{x' \in X'} \mathcal{D}_{inst}(x, x'))$ where f is a permutation-invariant aggregation function. Note that, if $|X| = |X'| = 1$, the recourse cost can be rewritten as the previous minimality metrics by appropriately defining $\mathcal{D}_{inst}(x, x')$. **Coverage** [28] measures the quality of the counterfactual explanations as the proportion of input instances $x \in X$ that have close counterfactuals in X' under a given distance threshold t, i.e., $Cov(X, X') = \frac{1}{X} \cdot |\{x \in X \mid \min_{x' \in X'} \mathcal{D}_{inst}(x, x') \leq \theta\}|$. **Compactness**, like *Interpretability* [28], accounts for the number of counterfactual instances included in the explanation set X'. The larger the number of counterfactual graphs in the solution, the harder it gets for a human to understand the explanation. Compactness returns a value in the interval $(0, 1]$ where values closer to 1 are preferred. We define the metric as $Comp(X') = \frac{1}{|X'|}$.

4.3 Adopted Evaluation Protocols for GCE

Here, we report the evaluation protocol adopted in the surveyed works. To the best of our knowledge, this is the first survey that provides the reader with a detailed description of the way GCE is evaluated in the literature. Knowing how different works evaluate their performance, not only via the metrics adopted but also via the datasets used, compared baselines, and evaluation scenarios, permits future researchers to have a deep understanding of the challenges in this field.

DCE [21], used as a baseline in [1], refers to explanations that belong to the same data distribution as the input instance. It searches for a counterfactual instance G^* in the dataset such that $G^* = \arg\min_{G' \in \mathcal{G},\ \Phi(G) \neq \Phi(G')} \mathcal{D}_{inst}(G, G')$.

DDBS and **OBS** [1] are originally proposed in the domain of -omics networks. Both methods are compared against a simple Data Search (DS) method that looks for a counterfactual instance in the dataset whose distance is minimal w.r.t. the original one. They are tested on the ASD and ADHD datasets. Since these methods are search-based, they exploit a simple white-box classifier that produces a 2-dimensional embedding for the input graph and pass this embedding to a linear classifier to produce the counterfactual. The authors of this work rely on the GED between the counterfactual example and the original instance and the Oracle Calls to evaluate the performance.

RCExplainer [4], as a factual-based explanation method, is compared to GNNExplainer [85], PGExplainer [40], PGM-Explainer [74], and CF-GNNExplainer. The baselines that produce a set of vertices V' as an explanation were modified to induce a subgraph $G' = (V', E')$ on V' whose edges, E', are the counterfactuals produced. Similarly, the baselines that identify a subgraph $G' = (V', E')$ were modified to output E' only. As seen in Table 2, the performance is assessed on graph classification and vertex classification, both relying on a GNN oracle. RCExplainer is

evaluated on BA-2motifs, Mutagenicity, and NCI1 for graph classification. For vertex classification, it is evaluated on BA-Shapes, BA-Community, Tree-Cycles, and Tree-Grid. Fidelity, Robustness, and Runtime are used for comparing RCExplainer with SoA methods.

GNN-MOExp [36] is only compared to SoA factual methods. The metrics used for evaluation are simulatability, probability of necessity, and robustness. Notice that simulatability measures whether a subgraph explanation preserves the classification of the original instance. Therefore, since counterfactuality aims at changing the original class, simulatability is not suitable in GCE. GNN-MOExp is tested in different social graphs for vertex classification, including CiteSeer, CORA, and Musae-Facebook.

MEG [53] is not compared to other SoA methods due to its inherent design for molecular graphs. TOX21 and ESOL are two benchmark datasets used to assess MEG's performances. Additionally, MEG incorporates a molecule sanitisation check to filter out invalid instances. Hence, TOX21 remains with $1,756$ equally distributed samples, while ESOL has $1,129$ compounds. A split of $80\% : 10\% : 10\%$ is used for training, validation, and test sets for both datasets. During the counterfactual generation procedure, MEG finds 10 counterfactuals for each input molecule ranked according to its multi-objective scoring function (reward). The counterfactuals are evaluated according to the MEG similarity function.

GNNAdv [68] provides factual and counterfactual explanation methods. The provided counterfactual explainer is not compared against any other SoA counterfactual method. Furthermore, the GCE method is only tested on MS–COCO [34], a dataset for image classification, considering the co-occurrence of object labels within the images as graphs. Accuracy is the only metric used to evaluate the counterfactual explainer because the authors consider counterfactual explanations as adding noise to the original instance and thus expect the accuracy of the oracle to decrease when counterfactuals are provided instead of the original instances.

CMGE [82] is based on particular knowledge graphs based on EMRs of hospitalised patients. First, the authors extract three main features from EMRs and submit them for human evaluation. The employed dataset, *not publicly available*, is composed of Chinese EMRs of lymphedema patients. The authors also use MIMIC-III-50 [49] to assign multiple **International Classification of Diseases** (**ICD**) codes to EMRs. The proposed method is tested for link prediction and its explanation capabilities are not directly compared against other counterfactual methods.

NSEG [12] is compared against other factual explanation methods and CF^2 [69]. The metrics used in the evaluation are versions of the fidelity metric, namely *Fidelity+* and *Fidelity-* to quantify necessity and sufficiency, respectively, and *charact score*, which is a combination of both. Furthermore, the authors use top-k accuracy and AUCROC, which are commonly used to evaluate factual explainers. NSEG's performance is evaluated on BA-Shapes, Tree-Cycles, Tree-Grid [86], Mutagenicity [30] and MSRC-21 [48].

CF-GNNExplainer [39] can only remove edges from the original instance. The authors compare their performance with Random, Only-1hop, Rm-1hop, and GNNExplainer [86]. Random is a method used for sanity checking and randomly removes edges from the original graph. Only-1hop and Rm-1hop are based on the 1-hop neighbourhood of the vertex - its ego-graph. Only-1hop keeps all the edges in the ego-graph, while Rm-1hop removes them. The employed datasets are Tree-Cycles, Tree-Grids, and BA-Shapes for vertex classification. The metrics adopted to assess the performance of the explanations are Fidelity, Sparsity, and the explanation size. The experimental results show that CF-GNNExplainer can generate counterfactuals for the majority of the vertices in the test set by removing only a small number of edges.

CLEAR [41] is compared to a Random method that performs a fixed number of random perturbations on the input graph and two of its variants, i.e., one that removes edges and one that adds edges. CLEAR also compares against SoA methods, among whom CF-GNNExplainer, MEG, and

GNNExplainer. Here, the authors modify GNNExplainer to remove the identified subgraph (factual instance) from the original graph to produce a possible counterfactual. CLEAR is tested on Community [41], an ad-hoc synthetic dataset, IMDB-M, and Ogbg-molhiv, relying on metrics such as runtime, correctness, proximity, and causality. For the causality aspect reported in the original article, the authors measure the ratio of counterfactuals that satisfy the causal constraints corresponding to a predefined relation of interest.

MACDA [50] is compared to Joint-List and MaMEG. Joint-List chooses the top-10 drug and protein counterfactual instances that have the highest difference in predicted affinity and similarity w.r.t. the original instance. MaMEG is an adaptation of MEG to the drug-target counterfactual generation task. Ad-hoc metrics are used to measure MACDA's minimality of the generated counterfactual, i.e., average drug encoding similarity and average protein encoding similarity. Additionally, MACDA relies on the Prediction Distance. The method uses the Davis and PDBBind datasets.

CF^2 [69] is a factual-based method compared to factual explanation methods, i.e., GNNExplainer, CF-GNNExplainer and GEM [35]. The datasets used are BA-Shapes, Tree-Cycles, Mutagenicity, NCI1, and CiteSeer. BA-Shapes, Tree-Cycles and CiteSeer are employed for vertex classification, while Mutagenicity and NCI1 for graph classification. Furthermore, BA-Shapes and Tree-Cycles have ground-truth motifs for explaining the classification since they are human-designed, while NCI1 and CiteSeer do not contain such motifs. The used metrics are Accuracy and the Probability of Necessity.[10]

MACCS [79] is tied to the molecular domain, and, as such, it is not compared to other SoA methods. It is tested on three real-world datasets: i.e., BBBP, ESOL, and HIV. It relies on a Random Forest as oracle to produce counterfactuals on BBBP, a GRU on ESOL, and a GCN on HIV. The Tanimoto similarity is used to assess the performance on all datasets.

GREASE [13] is designed to explain user-item recommendations, and it is not compared against other SoA explanation methods. It is tested on two real-world datasets, namely LastFM and Yelp. The evaluation metrics used are Probability of Necessity and Explanation Cost.

GCFExplainer [28] is compared against CF^2 and RCExplainer. It is also compared against a bespoke omniscient method that produces counterfactuals directly taken from the dataset for a particular desired class to explain. The datasets used are NCI1, Mutagenicity, AIDS, and Proteins [8, 19]. Since this work produces global counterfactual explanations, the metrics used for evaluation are coverage, recourse cost, and compactness. Additionally, standard metrics such as runtime are also considered. According to the global scenario this work adopts, GCFExplainer demonstrates that producing model-level counterfactuals generalises better than producing multiple counterfactuals at the instance level.

5 EMPIRICAL EVALUATION OF GCE METHODS

A comprehensive benchmark of existing GCE methods is out of the scope of this work. However, here we use GRETEL [58, 60] to assess the performance of several SoA methods in multiple domains.[11] We use one dataset for each scenario: i.e., Tree-Cycles (synthetic), ASD (-omics), and BBBP (molecular). Table 5 reports the datasets' general statistics. Meanwhile, Table 6 depicts the performance of the methods on the test set (i.e., 10% of $|\mathcal{G}|$). For each method, we report averages on 10-fold cross-validation. We use runtime, GED, the number of oracle calls, correctness, sparsity,

[10]The original article also presents precision, recall, and F1 scores for the evaluation. However, we report only counterfactual-related evaluation protocols.

[11]Notice that the repository is constantly updated and the number of explainers is periodically increased. Current reporting is done on the version of April 2023.

Table 5. The Dataset Characteristics

	$\lvert\mathcal{G}\rvert$	$\mu(\lvert V\rvert)$	$\sigma(\lvert V\rvert)$	$\mu(\lvert E\rvert)$	$\sigma(\lvert E\rvert)$	$\lvert C_0\rvert$	$\lvert C_1\rvert$	Class distr.	|Test set|
Tree-Cycles	500	32	0	31.54	0.62	263	237	0.526 : 0.474	50
ASD	101	116	0	655.62	7.29	52	49	0.515 : 0.485	10
BBBP	2,039	24.06	10.58	25.95	11.71	479	1560	0.235 : 0.765	203

$\lvert\mathcal{G}\rvert$ is the number of instances; $\mu(\lvert V\rvert)$ and $\sigma(\lvert V\rvert)$ represent the mean and std of the number of vertices per instance; $\mu(\lvert E\rvert)$ and $\sigma(\lvert E\rvert)$ represent the mean and std of the number of edges per instance; $\lvert C_i\rvert$ is the number of instances in class $i \in \{0, 1\}$. |Test set| represents the number of instances evaluated in each fold.

Table 6. Evaluation of the SoA for 10-fold Cross Validation

Dataset	Method	Runtime ↓	GED ↓	Oracle Calls ↓	Correctness ↑	Sparsity ↓	Fidelity ↑	Oracle Accuracy ↑
Tree-Cycles	RAND@5	**0.01 ± 0.003**	92.18 ± 5.44	0.00 ± 0.00	0.55 ± 0.50	1.45 ± 0.09	0.55 ± 0.50	1.00 ± 0.00
	RAND@10	0.02 ± 0.006	123.74 ± 7.43	0.00 ± 0.00	0.51 ± 0.50	1.94 ± 0.12	0.51 ± 0.50	1.00 ± 0.00
	RAND@15	0.01 ± 0.004	147.93 ± 8.26	0.00 ± 0.00	0.58 ± 0.50	2.33 ± 0.13	0.58 ± 0.50	1.00 ± 0.00
	DCE	0.13 ± 0.00	50.36 ± 0.00	501.00 ± 0.00	**1.00 ± 0.00**	0.79 ± 0.00	**1.00 ± 0.00**	1.00 ± 0.00
	OBS	0.07 ± 0.01	57.31 ± 0.03	149.45 ± 21.13	0.96 ± 0.01	0.90 ± 0.00	0.96 ± 0.01	1.00 ± 0.00
	DDBS	8.87 ± 0.10	71.79 ± 0.24	1342.62 ± 11.95	0.59 ± 0.01	1.13 ± 0.00	0.59 ± 0.01	1.00 ± 0.00
	MACCS	–	–	–	–	–	–	–
	CLEAR	2.47 ± 0.08	79.76 ± 3.60	0.00 ± 0.00	0.53 ± 0.10	1.26 ± 0.06	0.53 ± 0.10	1.00 ± 0.00
	CF^2	0.41 ± 0.01	**31.54 ± 0.12**	0.00 ± 0.00	0.47 ± 0.10	**0.50 ± 0.00**	0.47 ± 0.10	1.00 ± 0.00
	MEG	272.11 ± 5.66	159.70 ± 1.34	0.00 ± 0.00	0.53 ± 0.00	2.51 ± 0.02	0.53 ± 0.00	1.00 ± 0.00
ASD	RAND@5	1.45 ± 0.46	618.06 ± 8.27	0.00 ± 0.00	0.00 ± 0.00	0.80 ± 0.01	0.00 ± 0.00	0.79 ± 0.08
	RAND@10	2.76 ± 1.25	1152.93 ± 20.19	0.00 ± 0.00	0.00 ± 0.00	1.49 ± 0.02	0.00 ± 0.00	0.79 ± 0.08
	RAND@15	1.33 ± 0.39	1600.78 ± 18.22	0.00 ± 0.00	0.00 ± 0.00	2.08 ± 0.03	0.00 ± 0.00	0.79 ± 0.08
	DCE	**0.09 ± 0.02**	1011.69 ± 0.00	102.00 ± 0.00	**1.00 ± 0.00**	1.31 ± 0.00	**0.54 ± 0.00**	0.79 ± 0.08
	OBS	3.24 ± 1.13	**9.89 ± 0.11**	347.73 ± 15.11	**1.00 ± 0.00**	**0.01 ± 0.00**	**0.54 ± 0.00**	0.79 ± 0.08
	DDBS	83.46 ± 34.04	11.79 ± 0.29	362.05 ± 14.56	**1.00 ± 0.00**	0.02 ± 0.00	**0.54 ± 0.00**	0.79 ± 0.08
	MACCS	–	–	–	–	–	–	–
	CLEAR	0.45 ± 0.04	1739.60 ± 131.16	0.00 ± 0.00	0.47 ± 0.13	2.25 ± 0.17	0.25 ± 0.18	0.79 ± 0.08
	CF^2	0.69 ± 0.01	655.49 ± 2.87	0.00 ± 0.00	0.46 ± 0.09	0.85 ± 0.00	0.37 ± 0.15	0.79 ± 0.08
	MEG	×	×	×	×	×	×	×
BBBP	RAND@5	**0.01 ± 0.03**	30.98 ± 33.27	0.00 ± 0.00	0.85 ± 0.35	0.52 ± 0.23	0.62 ± 0.69	0.86 ± 0.02
	RAND@10	**0.01 ± 0.03**	52.98 ± 58.96	0.00 ± 0.00	0.86 ± 0.35	0.93 ± 0.41	0.65 ± 0.66	0.86 ± 0.02
	RAND@15	0.02 ± 0.12	82.97 ± 137.37	0.00 ± 0.00	0.85 ± 0.36	1.32 ± 0.70	0.61 ± 0.69	0.86 ± 0.02
	DCE	37.51 ± 5.21	27.92 ± 0.12	2040.00 ± 0.00	**1.00 ± 0.00**	0.59 ± 0.00	**0.72 ± 0.00**	0.86 ± 0.02
	OBS	2.92 ± 0.07	0.00 ± 0.00	314.61 ± 0.00	0.00 ± 0.00	**0.00 ± 0.00**	0.61 ± 0.00	0.86 ± 0.02
	DDBS	×	×	×	×	×	×	×
	MACCS	31.35 ± 0.97	**11.23 ± 0.08**	1221.33 ± 0.22	0.40 ± 0.00	0.19 ± 0.00	0.23 ± 0.00	0.86 ± 0.02
	CLEAR@1	213.21 ± 5.64	27056.29 ± 9.69	0.00 ± 0.00	0.87 ± 0.02	91.89 ± 0.18	0.64 ± 0.03	0.86 ± 0.02
	CLEAR@5	214.80 ± 6.97	26711.57 ± 112.67	0.00 ± 0.00	0.85 ± 0.02	90.71 ± 0.37	0.62 ± 0.03	0.86 ± 0.02
	CLEAR@15	251.93 ± 36.01	25986.66 ± 170.41	0.00 ± 0.00	0.85 ± 0.01	88.20 ± 0.66	0.62 ± 0.06	0.86 ± 0.02
	CF^2	84.41 ± 49.06	25.72 ± 0.63	0.00 ± 0.00	0.85 ± 0.02	0.09 ± 0.00	0.63 ± 0.03	0.86 ± 0.02
	MEG	90.66 ± 29.51	269.35 ± 0.39	0.00 ± 0.00	0.51 ± 0.04	0.91 ± 0.00	0.32 ± 0.04	0.86 ± 0.02

All oracles have been pre-trained. × depicts no convergence in two days; while – means the method cannot be adapted to the domain at-hand.

fidelity, and the oracle's accuracy as metrics. Notice that all methods share the same folds.[12] In this way, we guarantee a fair comparison of all methods over the same view of the data instances. Moreover, methods that are trained (i.e., CLEAR, CF^2, MEG) or oracle-oblivious (i.e., RAND@k) do not need to access the oracle at inference (test) time. We were unable to adapt MACCS to Tree-Cycles and ASD as its optimisation function incorporates considering the produced counterfactuals as valid molecules, hence the dashes in the table. In Section D.1, we provide reasons to exclude from this evaluation some of the methods surveyed in Section 3. In Section D.2, we describe the explainer hyperparameters. All configuration files needed to reproduce the conducted empirical evaluation are available on GRETEL's GitHub repository.[13]

Oracles have been pre-trained on the entire dataset, as happens in production. Hence, because all explainers are tested on the same folds, thus the oracle accuracy is the same across the board.

[12]The same number of folds and the same exact splits of the data.

[13]https://github.com/MarioTheOne/GRETEL

For Tree-Cycles we rely on an omniscient oracle that exploits a graph visit that verifies whether an already-visited vertex can be revisited, thus presenting a cycle. This oracle never fails to identify (a)cyclic graphs, thus guaranteeing a perfect accuracy of 1.00. For ASD, we rely on a white-box classifier used in [1]. This oracle is a rule-based classifier that looks at the co-activation of specific regions of interest in a brain graph.

For BBBP, we rely on a GCN with four graph convolutional layers interleaved with ReLU activation functions. The convolution is then aggregated via average pooling over the node features. This aggregation is finally passed to two dense layers with [256, 1] neurons and a final sigmoid.

We also compare the SoA with a bespoke random explainer: RAND@k builds a counterfactual graph $G' = (V, E')$ of $G = (V, E)$ by randomly choosing k% of the edges in $\binom{V}{2}$. First, RAND@k copies E to E'. Then it skims through the sampled k% edges. To this end, if a sampled edge $(v_i, v_j) \in E$, then $E' = E' - \{(v_i, v_j)\}$; contrarily, if $(v_i, v_j) \notin E$, then $E' = E' \cup \{(v_i, v_j)\}$. We use k $\in \{0.05, 0.10, 0.15\}$.

In Tree-Cycles, DCE has the highest correctness across the board. However, since DCE is search-based, it suffers from, potentially, a higher GED w.r.t. the other compared methods. As expected, RAND@k has correctness that is just above the chance level, which, itself, is a bar that most of the SoA does not reach (compare RAND@15 with CF^2, CLEAR, MEG, and DDBS). In detail, CF^2 has the lowest GED overall, but it fails to produce valid counterfactuals $\sim$ 53% of the time. Contrarily, CLEAR, and DDBS, although reporting a higher correctness than CF^2, produce counterfactuals that are nearly twice as distant to the input graph. This phenomenon can also be noticed in sparsity, which is a scaled GED (see Section 4.2). It is interesting to note that the correctness and fidelity are the same for all methods in Tree-Cycles. Recall that fidelity measures how faithful the counterfactual explanations are to the oracle considering their correctness. Because the oracle employed in this scenario is guaranteed to predict the correct class of the input, the fidelity is always faithful to the oracle's prediction. Thus, the $\chi(G)$ component of fidelity is always going to output 1, while $\mathbb{1}[\Phi(G') = y_G]$ depends on whether the explainer returns a valid counterfactual or not. In this scenario, since we know that the oracle is always right, we can attribute the misclassifications to the inability of the explainer to produce valid counterfactuals. We adapted MEG's action-choosing policy (agent environment) in this scenario by flipping the adjacency matrix of the input graph G (i.e., an existing edge is removed; a non-existing edge is added). Therefore, for each cell v_i, v_j in G's adjacency matrix A, MEG can take an action to produce a counterfactual $G' = (V, E')$ s.t. $A'[v_i, v_j] = 1 - A[v_i, v_j]$. Hence, for each input G, there are $\binom{|V|}{2}$ possible actions that lead to a potential valid counterfactual. Now, notice that MEG has the highest runtime across the board. This happens because the number of possible actions per instance are $\binom{32}{2} = 496$. Additionally, because MEG's graph perturbation policy mentioned before is constrained to a single edge addition/removal to produce a counterfactual, its GED is the largest across the board. Notice also that MEG's oracle calls depend on the way the environment is implemented. In this scenario, it does not require any information from the oracle to reward certain actions more than others (see the discussion on BBBP). Finally, CLEAR and CF^2 access each instance once per epoch during training. Hence, CLEAR has 450×600 oracle cals, while CF^2 has only 450×100.

In ASD, it is interesting to notice that RAND@k performs poorly. Without loss of generality, we expect RAND@k to perform poorly in scenarios where the decision boundary between classes is defined at the vertex/edge feature space instead of being expressed in terms of connectivity patterns. Because RAND@k operates only on the adjacency matrix of the input graphs, its correctness, and fidelity suggest that it is unable to produce any valid counterfactual. Besides, its GED is among the worst across the board. All SoA methods surpass the random explainer, which leads us to believe that feature-based SoA methods need to be further investigated to tackle hard cases

such as ASD. However, CLEAR and CF^2 do not have satisfactory correctness (~ 0.47 and ~ 0.46, respectively) as it does not exceed the chance level of producing a valid counterfactual. Additionally, CLEAR has the highest GED across the board. We believe this is due to the graph-matching procedure that CLEAR has after it samples a counterfactual G' from its latent space. This makes the sampled graph G' lose the order of the vertices w.r.t. the original instance. It is interesting to notice for DCE, OBS, and DDBS that their correctness and fidelity are the same. Furthermore, since we know that these methods always produce a valid counterfactual - i.e., correctness is equal to 1 - we can state that oracle is not capable of differentiating between factual and counterfactual instances (see oracle's accuracy). Note that we adapt MEG to this scenario as described in Tree-Cycles. Therefore, due to the elevated number of possible actions per instance $\binom{116}{2} = 6,670$, MEG is not able to generate a counterfactual after several days of execution. OBS and DDBS remain the best-performing methods in terms of correctness and GED. CF^2 has 91×100 oracle accesses during training, and CLEAR 91×600.

In BBBP, notice that RAND@k has good performances in terms of correctness. This means that the employed oracle is sensitive to the connectivity patterns of the molecules rather than the feature set of each vertex (e.g., atom, valence, ionisation) in the molecule graph. Contrarily to Tree-Cycles, perturbing more than 5% of the edges does not have any benefits in this scenario. Notice that DCE has the highest correctness and a low GED, which suggests that the instances in the test set are similar to one another due to DCE's inherent counterfactual searching mechanism. Although MACCS and MEG are designed to work with molecules, they perform poorly, as expected, in terms of correctness with ~0.4 and ~x, respectively, highlighting the necessity to have not only a data type-oriented explainers but also a domain-specific one. Meanwhile, their GED is better than DCE, which suggests that the counterfactual that DCE searches in the test set is not the least distant from the input molecule (DCE's GED is 27.92, while MACCS's is 11.23). CF^2 has one of the lowest GED, second to MACCS, even if it is not designed to work with molecular graphs specifically. Surprisingly, CF^2 reports correctness of 0.85 (random) that exceeds domain-specific methods (i.e., MACCS and MEG). Notice that OBS fails to produce valid counterfactuals (correctness and sparsity equal to 0). Because its GED is 0, we can conclude that it returns the original instance. DDBS fails to produce a valid explanation for a single instance within 4 hours[14] of searching through the test set. CLEAR has $1,836 \times 5$ oracle calls during training, while CF^2 has $1,836 \times 100$. Finally, notice that CLEAR cannot compete with the SoA in BBBP although its correctness score is the second highest due to an extremely elevated GED and sparsity. Due to the size of the dataset, we only use 5 epochs to train CLEAR, which might have affected its overall performance.

In conclusion, Table 6 paints a raucous picture of SoA methods. Most GCE methods fail to produce valid counterfactuals most of the time, even with an underlying omniscient and non-biased oracle (see Tree-Cycles). Besides, in the majority of scenarios, a random perturbation explainer outperforms the SoA, which entails the literature's need to perform an exhaustive evaluation with baselines and not only with other SoA explainers (notice BBBP). Generally, according to these empirical results, there is no silver bullet explanation method. The application domain and the particularities of the dataset influence explanation capabilities. Hence, it is important to use multiple explanation methods as a way to increase fairness and trustworthiness.

6 PRIVACY OF EXPLAINERS AND FAIRNESS OF ORACLES IN GCE

The European Union's vision for AI is to encourage excellence and trustworthiness, boost research and productivity, reinforce safety, and protect citizen rights [14]. To this end, the EU proposed a risk-based legal framework to regulate AI and address risks specifically created by

[14]We ran experiments on an AMD Ryzen 7 4800HS, 2.90 GHz, 32 GB RAM.

AI applications [15]. The framework aims at ensuring AI systems' safety, privacy, fairness, and trustworthiness to create safer and more innovation-friendly digital environments.

The Commission has placed extensive regulations for AI's trustworthiness and the risk mitigation of its wide usage. The proposal points towards black-box models that are inherently non-explainable. In this context, interpretable models are seen as more compliant with the EU regulation for AI than their black-box counterparts. Despite that, on the one hand, good counterfactual explanations give domain laymen insight into the prediction of a specific input. On the other hand, GCE methods can pose a significant privacy risk, even when the explainer Ξ does not have direct access to the original data. When Ξ generates a counterfactual G' for the input G it can breach the privacy of sensitive information. For example, in a social network, Ξ may include vertices that belong to sensitive groups, such as juveniles, without considering established privacy policies within the network.

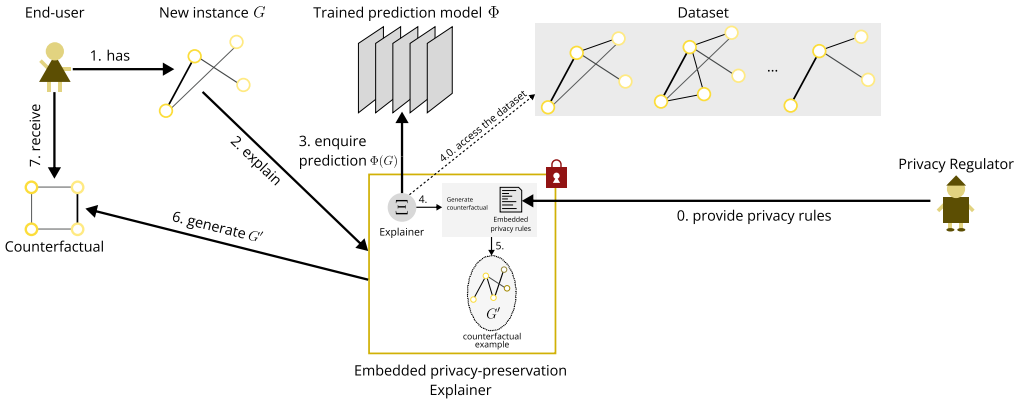
6.1 Privacy of Explainers

The works surveyed here and the aspects they treat to provide counterfactuals to the end-user do not acknowledge any violation of privacy-related policies. For this reason, we propose two high-level solutions to tackle this aspect: i.e., embedded and post-hoc privacy preservation explainers. They differ in how the explainer Ξ interacts with the original data and the end-user. Figures 4(a) and (b) depict embedded and post-hoc privacy preservation explainers, respectively. Figure 4(a) shows an explainer with an embedded mechanism that constrains the way counterfactuals are generated. As shown in step 4, Ξ must either have full or partial access to the dataset. If partial access is granted, Ξ would only have access to specific vertex/edge attributes while being barred from extracting sensitive information such as user profiles and other personal data. This access level is critical to generating a counterfactual example that complies with the privacy rules defined in step 0. For example, in a social network where user vertices have attributes like age, gender, friendships, and profile pictures, Ξ must adhere to privacy policies restricting the extraction of information about profile pictures and gender. Contrarily, Figure 4(b) shows how Ξ delegates the task of assessing privacy compliance to an external module: i.e., software exoskeleton, which mediates actions in the digital world according to the regulation. The privacy compliance module receives the privacy rules - step 0 - and the explainer directly communicates with it without requiring them to be embedded within. Instead, Ξ generates a counterfactual G' arbitrarily and then uses a binary test to assess if G' complies with the privacy rules. If the test indicates a violation, Ξ repeats this procedure in the next iteration until a compliant G' is generated (steps 5.1-5.3). Once generated, G' is presented to the end-user (steps 5.4-7).

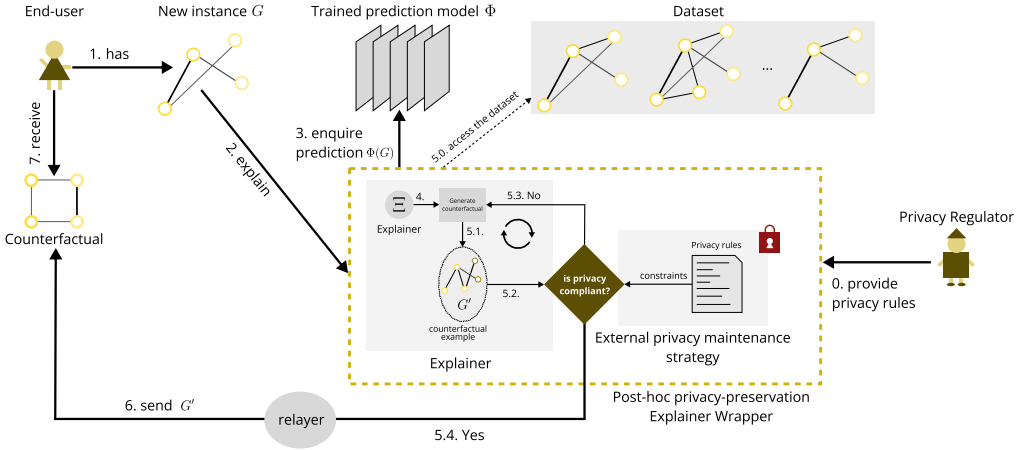
Note that generating a privacy-compliant counterfactual may require considerable effort in post-hoc privacy-preservation explainer strategies if the external privacy maintenance module is limited to providing a binary compliance test. Therefore, a more complex test that identifies the source of the compliance failure is necessary to improve the explainer's ability to regulate the generation procedure and mitigate privacy violations in the next iteration. In this case, we propose to use a wrapper to entice the interaction of the explainer with the privacy rules without exposing the rules to the outside. All interactions with the outer part go through the wrapper, not the components. Finally, embedded and post-hoc privacy-preservation explainers can access the dataset to generate more plausible explanations. However, if dataset access is mandatory, it should be handled via an encrypted and trusted channel.

6.2 Fairness of Oracles

Generally, fairness refers to whether the black-box model accurately represents and predicts incoming instances based on specific features without exhibiting discriminatory behaviour. For example,



(a) Workflow of generating counterfactual examples G' of a graph instance G with an embedded privacy-preservation strategy. Notice that step 0 can be done a priori to fetch specific privacy preservation rules indicated in the original data.



(b) Workflow of generating counterfactual examples G' of a graph instance G with a post-hoc privacy-preservation strategy. The explainer delegates the privacy compliance assessment to an external module.

Fig. 4. Workflow of privacy-preserving counterfactual explainers.

consider a toy dataset of banned users on a social network, with information about each individual's race. If the prediction model, Φ, outputs a 1 (i.e., ban) only based on the user's race [52], irrespective of their connection with other users who post derogatory content or any previous terms of service violations, it is an instance of biased prediction. In such cases, counterfactual examples produced under a biased black-box predictor can be an auditing mechanism to detect unfair disparities. For example, "*if the person had not been of a certain race, then they would not have been banned from the social network*" is an unfair/biased counterfactual example highlighting Φ's underlying unfairness. Thus, GCE can be a monitoring strategy to detect risky and unfair predictors. This auditing characteristic is even more crucial when the discriminated group corresponds to the minority group in the dataset, resulting in class imbalance. A predictor may perform well according to specific evaluation metrics, but it may still be unfair. Therefore, counterfactual examples play an essential role in questioning the fairness of the prediction model in any scenario.

Moreover, while the literature has explored measuring bias and unfairness of the prediction model [6, 47, 56], measuring them for counterfactual explainers is an aspect that is yet to be covered. Lastly, the fairness of the oracles was widely investigated in the literature [26]. However,

the literature on counterfactual explainability has yet to investigate the fairness of the produced counterfactual examples. Nonetheless, the fairness of the prediction method provides insight into whether the explainer might generate biased explanations. Thus, the explainer's bias is an aspect that needs to be considered when evaluating the *effectiveness* of the generated explanation.

7 OPEN CHALLENGES AND FUTURE WORKS

Here, we provide the reader with insight into the remaining open challenges yet to be treated in the area of graph counterfactual explainability so that researchers can concentrate on providing contributions therein.

According to the taxonomy presented in Section 3, there is a clear lack of methods that address model-level and global-level graph counterfactual explanation. Thus, we expect that in the near future, we will see a growing interest in this sub-domain with the proposition of methods based on learning and search.

Moreover, attentive readers have certainly noticed the complete absence of counterfactual methods designed for edge prediction tasks. This might be primarily imputed to the difficulty of the task and the dubious effectiveness that a counterfactual explanation could have in this case. Nonetheless, we expect a future exploration of this research field.

Even though, in this survey, we extend the definition of a counterfactual explanation to the multi-class setting, more work must be conducted to clarify it in the scenario of multi-label prediction. To the best of our knowledge, the research area also lacks any method dedicated to this task.

Similar to [44], researchers could investigate how to measure the discrimination (i.e., unfairness/bias) of the explainers. In this way, explainability can be harnessed to exalt the unfairness of the underlying prediction model or to determine whether the explainer itself suffers from bias. Even though we present some ideas on privacy-preserving possibilities, the field must be explored further. Therefore, new-generation explainers should be capable of complying with the regulation's guidelines, which are being continuously adopted in more countries.

Besides plain graph data, counterfactual explainability can be used to provide explanations on more complex structures such as temporal graphs, and manifolds. For temporal graphs, explanations should contain a temporal component that better provides counterfactual examples for a specific time interval. Contrarily, for manifolds,[15] counterfactual examples might be considered as a *surface* in the same vector space of the input instance.

We invite the reader to focus on the core of this survey, which emphasises explainability with GNNs for graph counterfactuals. In light of recent regulations on AI trustworthiness,[16] it becomes essential to consider a broader perspective on Trustworthy AI. For a more in-depth understanding of fairness, explainability, and bias within the trustworthiness umbrella term, we refer the reader to [26].

Finally, as also highlighted in Section 5, this field of research might benefit from a systematic organisation of public competitions (never done until now) regarding the evaluation of counterfactual explainers as it happens with competitions in data mining (e.g., KDDCup15) that encourage a uniform evaluation benchmark with specific and well-formatted datasets.

8 CONCLUSION

We presented a thorough survey on the methods and best-practices for graph counterfactual explainability accompanied by rigorous formalisations of a minimal counterfactual explanation, the

[15]Usually manifolds are represented as meshes of triangles in a three-dimensional space.
[16]https://artificialintelligenceact.eu/the-act/

evaluation protocols, including datasets and measurements, empirical evaluation, and means to construct privacy-compliant explainers.

In detail, we provided the reader with an organisation of the literature according to a uniform formal notation, thus, simplifying potential comparisons w.r.t to the method's advantages and disadvantages. We emphasise that this is the first work to propose a formalisation of GCE under a multi-class prediction problem. Additionally, we provided a definition that encompasses the global minimal - i.e., the least distant counterfactual w.r.t. the original graph among all classes - for a chosen black-box prediction model.

We proposed a classification of the existing methods according to ten dimensions which aid the reader identify those methods that better suit their scenario of explainability. Besides this classification, we summarised the strengths and weaknesses of the surveyed methods. We also delve into shedding light on the benefits of a standardised evaluation protocol where we enlist synthetic and real datasets used in the literature, the adopted evaluation measures and the evaluation strategies of the surveyed methods.

Additionally, we argue that a fully-extensible and reproducible GCE evaluation framework is of paramount importance. Therefore, we illustrate an empirical evaluation, made with GRETEL which is an evaluation framework that concentrates on providing a highly modular architecture that permits the reader to plug-and-play with their ad-hoc explainer models, synthetic dataset generation, and evaluation metrics.

Finally, we discussed privacy and fairness in GCE which are necessary to comply with regulations being stabilised in more countries worldwide. As a concluding remark, we leave the reader with future directions and open challenges to be tackled in the future of this research area.

REFERENCES

[1] C. Abrate and F. Bonchi. 2021. Counterfactual graphs for explainable classification of brain networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2495–2504.

[2] K. Amara, Z. Ying, Z. Zhang, Z. Han, Y. Zhao, Y. Shan, U. Brandes, S. Schemm, and C. Zhang. 2022. GraphFramEx: Towards systematic evaluation of explainability methods for graph neural networks. In *Proceedings of the Learning on Graphs Conference, LoG 2022 (Proceedings of Machine Learning Research, Vol. 198)*. Bastian Rieck and Razvan Pascanu (Eds.), PMLR, 44.

[3] A. Artelt and B. Hammer. 2019. *On the computation of counterfactual explanations. A survey.*

[4] M. Bajaj, L. Chu, Z. Y. Xue, J. Pei, L. Wang, P. C. H Lam, and Y. Zhang. 2021. Robust counterfactual explanations on graph neural networks. *Advances in Neural Information Processing Systems* 34, (2021), 5644–5655.

[5] D. Bajusz, A. Rácz, and K. Héberger. 2015. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics* 7, 1 (2015), 1–13.

[6] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. Fairness and machine learning: Limitations and Opportunities. fairmlbook.org.

[7] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115. DOI: https://doi.org/https://doi.org/10.1016/j.inffus.2019.12.012

[8] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H. Kriegel. 2005. Protein function prediction via graph kernels. *Bioinformatics* 21, suppl_1 (2005), i47–i56.

[9] A. Bringas Colmenarejo, L. Nannini, A. Rieger, K.M. Scott, X. Zhao, G.K. Patro, G. Kasneci, and K. Kinder-Kurlanda. 2022. Fairness in agreement with European values: An interdisciplinary perspective on ai regulation. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*.

[10] Jesse Brown, Jeffrey Rudie, Anita Bandrowski, John Van Horn, and Susan Bookheimer. 2012. The UCLA multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis. *Frontiers in Neuroinformatics* 6 (2012). DOI: https://doi.org/10.3389/fninf.2012.00028

[11] R. Byrne. 2019. Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning.. In *Proceedings of the IJCAI*. 6276–6282.

[12] R. Cai, Y. Zhu, X. Chen, Y. Fang, M. Wu, J. Qiao, and Z. Hao. 2023. *On the Probability of Necessity and Sufficiency of Explaining Graph Neural Networks: A Lower Bound Optimization Approach.*
[13] Z. Chen, F. Silvestri, J. Wang, Y. Zhang, Z. Huang, H. Ahn, and G. Tolomei. 2022. *GREASE: Generate Factual and Counterfactual Explanations for GNN-based Recommendations.*
[14] European Commission. 2020. *On Artificial Intelligence.—A European Approach to Excellence and Trust.* Retrieved 20 April 2023 from https://ec.europa.eu/info/sites/default/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf
[15] European Commission. 2021. *Regulatory Framework Proposal on Artificial Intelligence.* Retrieved 21 April 2023 from https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai
[16] C. Craddock, Y. Benhajali, C. Chu, F. Chouinard, A. Evans, A. Jakab, B. Singh Khundrakpam, J. D. Lewis, Q. Li, and M. Milham. 2013. The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics* 7, 27 (2013), 5.
[17] Hiv Dataset. 2019. *DTP NCI Bulk Data for Download - Nci DTP Data - nci wiki.* Retrieved 16 January 2023 from https://wiki.nci.nih.gov/display/NCIDTPdata/
[18] M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber, and P. P. Zarrinkar. 2011. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology* 29, 11 (2011), 1046–1051.
[19] P. D. Dobson and A. J. Doig. 2003. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology* 330, 4 (2003), 771–783.
[20] Elahi. 2019. *Cheminformatics - Dataset for Molecular Machine Learning for Drug Discovery.* Retrieved 16 January 2023 from https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=hiv.zip
[21] L. Faber, A. K. Moghaddam, and R. Wattenhofer. 2020. Contrastive graph neural network explanation. In *Proceedings of the 37th Graph Repr. Learning and Beyond Workshop at ICML 2020.* Int. Conf. on Machine Learning, 28.
[22] C.L. Giles, K.D. Bollacker, and S. Lawrence. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the 3rd ACM Conference on Digital Libraries.*
[23] R. Guidotti. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery* 36, 6 (2022), 1–55.
[24] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini. 2019. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems* 34, 6 (2019), 14–23.
[25] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Computing Surveys* 51, 5 (2018), 1–42.
[26] Zh. Guo, T. Xiao, C. Aggarwal, H. Liu, and S. Wang. 2023. *Counterfactual Learning on Graphs: A Survey.*
[27] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems* 33 (2020), 22118–22133. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2020/file/fb60d411a5c5b72b2e7d3527cfc84fd0-Paper.pdf
[28] Z. Huang, M. Kosan, S. Medya, S. Ranu, and A. Singh. 2023. Global counterfactual explainer for graph neural networks. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining* . Association for Computing Machinery, New York, NY, 141–149.
[29] S. Jain and B.C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Tech., Vol. 1.* 3543–3556.
[30] J. Kazius, R. McGuire, and R. Bursi. 2005. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry* 48, 1 (2005), 312–320.
[31] K. Kersting, N. M. Kriege, C. Morris, P. Mutzel, and M. Neumann. 2016. Benchmark Data Sets for Graph Kernels. Retrieved 02 February 2023 from http://graphkernels.cs.tu-dortmund.de
[32] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik. 2020. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology* 1, 4 (2020), 045024.
[33] T. Lanciano, F. Bonchi, and A. Gionis. 2020. Explainable classification of brain networks via contrast subgraphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 3308–3318.
[34] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13.* Springer, 740–755.
[35] W. Lin, H. Lan, and B. Li. 2021. Generative causal explanations for graph neural networks. In *Proceedings of the International Conference on Machine Learning.* PMLR, 6666–6679.
[36] Y. Liu, C. Chen, Y. Liu, X. Zhang, and S. Xie. 2021. Multi-objective explanations of GNN predictions. In *Proceedings of the 2021 IEEE International Conference on Data Mining.* IEEE, 409–418.
[37] Octavio Loyola-González. 2019. Black-Box vs. White-Box: Understanding their advantages and weaknesses from a practical point of view. *IEEE Access* 7 (2019), 154096–154113. DOI: https://doi.org/10.1109/ACCESS.2019.2949286

[38] Ana Lucic, Hinda Haned, and Maarten de Rijke. 2020. Why does my model fail? Contrastive local explanations for retail forecasting. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 90–98.

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

[40] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems* 33 (2020), 19620–19631. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2020/file/e37b08dd3015330dcbb5d6663667b8b8-Paper.pdf

[41] Jing Ma, Ruocheng Guo, Saumitra Mishra, Aidong Zhang, and Jundong Li. 2022. CLEAR: Generative counterfactual explanations on graphs. In *Proceedings of the Advances in Neural Information Processing Systems*. S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35, Curran Associates, Inc., 25895–25907. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/file/a69d7f3a1340d55c720e572742439eaf-Paper-Conference.pdf

[42] L. Madeddu and G. Stilo. 2022. Deep learning methods for network biology. In *Proceedings of the Deep Learning In Biology And Medicine*. World Scientific, 197–246.

[43] L. Madeddu, G. Stilo, and P. Velardi. 2020. A feature-learning-based method for the disease-gene prediction problem. *International Journal of Data Mining and Bioinformatics* 24, 1 (2020), 16–37.

[44] M. Marchiori Manerba and R. Guidotti. 2022. Investigating debiasing effects on classification and explainability. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*. Association for Computing Machinery, 468–478.

[45] I. F. Martins, A. L. Teixeira, L. Pinheiro, and A. O. Falcao. 2012. A Bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of Chemical Information and Modeling* 52, 6 (2012), 1686–1697.

[46] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the Construction of Internet Portals with Machine Learning. *Inf. Retr.* 3, 2 (2000), 127–163. DOI: https://doi.org/10.1023/A:1009953814988

[47] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys* 54, 6 (2021), 1–35.

[48] C. Morris, N. M Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann. 2020. *TUDataset: A collection of benchmark datasets for learning with graphs*.

[49] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the NAACL-HLT*. 1101–1111.

[50] Tri Minh Nguyen, Thomas P. Quinn, Thin Nguyen, and Truyen Tran. 2023. Explaining black box drug target prediction through model agnostic counterfactual samples. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 20, 2 (2023), 1020–1029. DOI: https://doi.org/10.1109/TCBB.2022.3190266

[51] Akshat Kumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alan Aspuru-Guzik. 2021. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *Chemical Science* 12, 20 (2021), 7079–7090.

[52] K. M. Nowotny, Z. Bailey, and L. Brinkley-Rubinstein. 2021. The contribution of prisons and jails to US racial disparities during COVID-19. *American Journal of Public Health* 111, 2 (2021), 197.

[53] D. Numeroso and D. Bacciu. 2021. Meg: Generating molecular counterfactual explanations for deep graph networks. In *Proceedings of the 2021 International Joint Conference on Neural Networks* . IEEE, 1–8.

[54] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. Schardl, and C. Leiserson. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5363–5370.

[55] R. Pemantle. 1992. Vertex-reinforced random walk. *Probability Theory and Related Fields* 92, 1 (1992), 117–136.

[56] D. Pessach and E. Shmueli. 2022. A review on fairness in machine learning. *ACM Computing Surveys* 55, 3 (2022), 1–44.

[57] Jeremy Petch, Shuang Di, and Walter Nelson. 2022. Opening the black box: the promise and limitations of explainable machine learning in cardiology. *Canadian Journal of Cardiology* 38, 2 (2022), 204–213.

[58] M.A. Prado-Romero, B. Prenkaj, and G. Stilo. 2023. Developing and evaluating graph counterfactual explanation with GRETEL. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*. 1180–1183.

[59] M.A. Prado-Romero, B. Prenkaj, G. Stilo, A. Celi, E.L. Estevanell-Valladares, and D.A. Valdés Pérez. 2022. Ensemble approaches for graph counterfactual explanations. In *Proceedings of the 3rd Italian Workshop on Explainable Artificial Intelligence co-located with 21th Int. Conf. of the Italian Assoc. for Artificial Intelligence*. CEUR-WS.org, 88–97.

[60] M.A. Prado-Romero and G. Stilo. 2022. GRETEL: Graph counterfactual explanation evaluation framework. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, New York, NY, 4389–4393.

[61] Bardh Prenkaj, Damiano Distante, Stefano Faralli, and Paola Velardi. 2021. Hidden space deep sequential risk prediction on student trajectories. *Future Generation Computer Systems* 125, (2021), 532–543. DOI: https://doi.org/https://doi.org/10.1016/j.future.2021.07.002

[62] B. Prenkaj, P. Velardi, D. Distante, and S. Faralli. 2020. A reproducibility study of deep and surface machine learning methods for human-related trajectory prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 2169–2172.

[63] K. Riesen and H. Bunke. 2008. IAM graph database repository for graph based pattern recognition and machine learning. In *Proceedings of the Structural, Syntactic, and Statistical Pattern Recognition*. Springer, Berlin, 287–297.

[64] B. Rozemberczki, C. Allen, and R. Sarkar. 2021. *Multi-scale Attributed Node Embedding*.

[65] Benedek Rozemberczki and Rik Sarkar. 2020. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 1325–1334.

[66] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2008. The graph neural network model. *IEEE Trans. on Neural Networks* 20, 1 (2008), 61–80.

[67] S. Srinivas, A. Subramanya, and R. V. Babu. 2017. Training sparse neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 455–462.

[68] Y. Sun, A. Valente, S. Liu, and D. Wang. 2021. *Preserve, Promote, or Attack? GNN Explanation via Topology Perturbation*.

[69] J. Tan, S. Geng, Z. Fu, Y. Ge, S. Xu, Y. Li, and Y. Zhang. 2022. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *Proceedings of the ACM Web Conference 2022*. 1018–1027.

[70] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. 2018. Graph attention net. In *Proceedings of the 6th Int. Conf. on Learning Repr., ICLR*.

[71] I. Verenich, M. Dumas, M. La Rosa, and H. Nguyen. 2019. Predicting process performance: A white-box approach based on process models. *Journal of Software: Evolution and Process* 31, 6 (2019), e2170.

[72] Hanuman Verma, Saurav Mandal, and Akshansh Gupta. 2022. Temporal deep learning architecture for prediction of COVID-19 cases in India. *Expert Systems with Applications* 195, (2022), 116611. DOI: https://doi.org/https://doi.org/10.1016/j.eswa.2022.116611

[73] S. Verma, V. Boonsanong, M. Hoang, K. E. Hines, J. P. Dickerson, and C. Shah. 2022. *Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review*.

[74] Minh Vu and My T. Thai. 2020. PGM-Explainer: Probabilistic graphical model explanations for graph neural networks. In *Advances in Neural Information Processing Systems* 33 (2020), 12225–12235. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2020/file/8fb134f258b1f7865a6ab2d935a897c9-Paper.pdf

[75] N. Wale, I.A. Watson, and G. Karypis. 2008. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems* 14, 3 (2008), 347–375.

[76] B. Wang, J. Jia, and N.Z. Gong. 2021. Semi-supervised node classification on graphs: Markov random fields vs. graph neural networks.. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[77] R. Wang, X. Fang, Y. Lu, and S. Wang. 2004. The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *Journal of Medicinal Chemistry* 47, 12 (2004), 2977–2980.

[78] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.

[79] G. P. Wellawatte, A. Seshadri, and A. D. White. 2022. Model agnostic generation of counterfactual explanations for molecules. *Chemical Science* 13, 13 (2022), 3697–3705.

[80] M. Welling and T.N. Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *Proceedings of the J. Int. Conf. on Learning Repr.*.

[81] O. J. Wouters, M. McKee, and J. Luyten. 2020. Estimated research and development investment needed to bring a new medicine to market, 2009-2018. *Jama* 323, 9 (2020), 844–853.

[82] H. Wu, W. Chen, S. Xu, and B. Xu. 2021. Counterfactual supporting facts extraction for explainable medical record based diagnosis with graph network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Techs*. 1942–1955.

[83] Z. Wu, B. Ramsundar, E.N. Feinberg, J. Gomes, C. Geniesse, A.S. Pappu, K. Leswing, and V. Pande. 2018. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science* 9, 2 (2018), 513–530.

[84] P. Yanardag and S. Vishwanathan. 2015. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1365–1374.

[85] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 974–983.

[86] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. 2019. GNNExplainer: Generating Explanations for Graph Neural Networks. In *Advances in Neural Information Processing Systems* 32 (2019), 9240–9251. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2019/file/d80b7040b773199015de6d3b4293c8ff-Paper.pdf

[87] J. You, J.M. Gomes-Selman, R. Ying, and J. Leskovec. 2021. Identity-aware graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[88] H. Yuan, H. Yu, S. Gui, and S. Ji. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 5 (2022), 5782–5799.

[89] T. Zhao, X. Zhang, and S. Wang. 2021. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 833–841.

[90] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley. 2019. Optimization of molecules via deep reinforcement learning. *Scientific Reports* 9, 1 (2019), 10752.

Received 21 October 2022; revised 28 July 2023; accepted 21 August 2023