Counterfactual Explanations for Equivariant Graph Neural Networks: CE-EGNN for Molecular Property Regression

Salah-Eddine Rizki[1], Fatia Lekbour[1] and Guillaume Renton[1]

[1]ETIS, UMR 8051, CYU, ENSEA, CNRS, Cergy, France

Abstract

One of the aims of Explainable Artificial Intelligence is to employ deep learning models that might capture the underlying rules governing data on par with explainable methods to understand these rules. This requires both components to be perfectly adapted to the studied problem. In this paper, we focus on 3D molecular data represented as graphs with regression tasks. If the literature is rich concerning explainability and Graph Neural Networks for classification tasks, it is not discussed for regression. We thus present a simple way to perform counterfactual explanations with regression tasks and some preliminary results obtained on the QM9 dataset.

Keywords

Explainability, Graph Neural Networks, Counterfactual

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

1. Introduction

The recent advances in Machine Learning have led to many breakthroughs in many areas, such as computer vision, natural language processing or chemoinformatics, frequently outperforming humans on these tasks.

In this paper, we are particularly interested in molecular data represented as graphs. Graphs are ubiquitous modalities that allow to represent structural information within data. This paper focuses on drug discovery, and in particular graph generation tasks that can effectively support *inverse design* of molecular systems: from an exponential chemical space, generate synthesized molecules that satisfy multiple structural and functional constraints.

In order to improve graph generation, we believe that having a better understanding of the molecular properties is required. The need for explainability is to understand why these models take a certain decision in AI systems. It could also help to provide a better understanding of the desired properties, from an AI for science point of view.

Explainability is generally divided into two categories, factual and counterfactual explanations. On one hand, factual explanations on graphs are looking for the smallest subgraphs that yield the same prediction as the original graph (the *sufficient* explanations). On the other hand, counterfactual explanations with graphs are looking for the smallest changes that allow to significantly modify the prediction (the *necessary* explanations).

Counterfactual explanations have been mostly studied with classification, where a significant change of the prediction can be clearly defined, in particular by changing the predicted class.

However, it is less obvious for regression tasks, which yet could benefit from counterfactual explanations as much as classification tasks. Indeed, defining what a significant change with regression is not as easy than modifying the predicted class, leaving a whole set of tasks without clear explanations.

Published in the Proceedings of the Workshops of the EDBT/ICDT 2026 Joint Conference (March 24-27, 2026), Tampere, Finland

salah-eddine.rizki@ensea.fr (S. Rizki); fatia.lekbour@ensea.fr (F. Lekbour); guillaume.renton@ensea.fr (G. Renton)

Few recent works have addressed regression settings, for example through model-agnostic formulations for regressors and specialised methods for regression trees [1, 2]. However, these approaches typically assume vector inputs or tree-based models and do not readily extend to structured graph data, nor to 3D molecular data.

In this paper, we thus explore a simple way to generate counterfactual explanations. After this short introduction, section 2 quickly presents the state-of-the-art with counterfactual graph explanation. In section 3, we present our methodology, including the GNN and the explanation method. Section 4 focuses on the results and section 5 concludes this paper.

2. Related Works

2.1. Graph Neural Networks

Graph Neural Networks are a particular kind of neural networks dedicated to graphs, based on an update-aggregate strategy. Among the most popular, Graph Convolutional Network (GCN) [3] employs the Laplacian matrix that is extended with the identity matrix. Graph Attention Network (GAT) [4] on its side adapts attention concept for graphs, by considering an attention between each pair of connected nodes in the graph. Graph Isomorphism Network (GIN) [5] proposes a GNN that is as powerful as Weisfeiler-Lehman test to discriminate non-isomorphic graphs by adding a learnable parameter on the ego node. Finally, Equivariant Graph Neural Networks (EGNN) [6] focus on 3D data and propose a graph neural network that is equivariant to translation, rotation and reflection.

2.2. Explainability of GNNs

The literature in graph counterfactual explanations can be divided into multiple categories [7], including search-based methods, heuristic-based methods and learning-based methods. The learning-based method can also be divided into three subcategories: perturbation methods, reinforcement learning methods and generative methods. In this work, we focus on the learning-based methods.

Within generative methods, D4explainer [8] uses graph diffusion models to generate a counterfactual graph that resembles the input graph. GNNviz [9] uses adversarial attacks to manipulate edges and to generate counterfactual explanations. Finally, CLEAR [10] employs Variational Graph AutoEncoder to encode the input graph into an embedding space, and decode it into a counterfactual explanation.

Among the reinforcement learning methods, on one side, MEG [11] focuses on molecules and aims to generate counterfactual graphs that are valid molecules. On the other side, MACDA [12] produces counterfactual explanations for drug-target affinity prediction. In both methods, the reward function is used to obtain a molecule close to the original one.

Finally, perturbation methods represent the counterfactual explanation as an optimization problem. CF-GNNExplainer [13] is the counterfactual part of GNNExplainer [14], and it aims at perturbing the graph structure through the adjacency matrix, by getting the smallest amount of edge removal that will change the prediction. CF^2 [15] is looking for an explanation that is both counterfactual and factual, i.e. necessary and sufficient, by solving a constrained multi-objective optimization problem. NSEG [16] goes one step further and aims at optimizing a lower bound of the most necessary and sufficient explanations, according to the Probability of Necessity and Sufficiency (PNS).

3. Proposed approach

In this paper, we are particularly interested in 3D molecules with regression tasks. For this reason, we need to employ a Graph Neural Network that considers the spatial information of the atom, and we decided to use EGNN as a base predictor. We also employed CF-GNNExplainer as an explainer, which we updated to consider regression tasks. The GNN, the explainer and the modifications are presented below.

3.1. Problem Setting

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{R})$ denote a graph, where:

- $\mathcal{V}$ is the set of nodes,
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges,
- $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ are node features,
- $\mathbf{R} \in \mathbb{R}^{|\mathcal{V}| \times k}$ are optional node coordinates in a metric space (e.g., $k = 3$ for Euclidean space).

Edge attributes are currently out of the scope, but future work will consider this information.

We consider a *graph-level regression task*, where a trained predictor f_θ maps an input graph to a scalar output:

$$f_\theta : \mathcal{G} \to y \in \mathbb{R} \tag{1}$$

Given an input graph $\mathcal{G}$ with prediction $y_{\text{orig}} = f_\theta(\mathcal{G})$, our objective is to generate a *counterfactual explanation* in the form of a minimally modified graph $\mathcal{G}_{cf}$ such that the predictor output satisfies a desired target condition, i.e. a significant change in the predicted value.

We propose to employ *targeted regression counterfactuals*, defined by:

$$y_{\text{target}} = (1 + \alpha_{\text{shift}})\, y_{\text{orig}}, \tag{2}$$

where $\alpha_{\text{shift}} \in \mathbb{R}$ controls a relative change in the prediction, allowing to control the desired amount of changes.

Finally, we consider a counterfactual explanation as valid if it falls within the range of the targeted value, i.e. if:

$$\left| f_\theta(\mathcal{G}_{cf}) - y_{\text{target}} \right| \leq \tau, \tag{3}$$

for a tolerance parameter $\tau > 0$, while the modification from $\mathcal{G}$ to $\mathcal{G}_{cf}$ is kept minimal.

3.2. Base Predictor: Equivariant Graph Neural Network

The base predictor f_θ is an *Equivariant Graph Neural Network (EGNN)*[6], which explicitly incorporates relational and geometric information while preserving equivariance to transformations of the coordinate space.

Each EGNN layer performs message passing using node features and pairwise geometric relations. For each directed edge $(j \to i)$, a message embedding is computed as:

$$\mathbf{m}_{ij} = \phi_m \left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \|\mathbf{r}_i - \mathbf{r}_j\|, \mathbf{e}_{ij} \right), \tag{4}$$

where $\mathbf{h}_i^{(l)}$ denotes the node embedding at layer l, $\mathbf{r}_i \in \mathbf{R}$ denotes the coordinates of node i, and $\mathbf{e}_{ij}$ represents optional edge attributes.

Following the edge inference mechanism proposed in [6], a learnable gate modulates each message:

$$a_{ij} = \sigma\left(\phi_a(\mathbf{m}_{ij})\right) \in (0, 1), \tag{5}$$

where $\sigma(\cdot)$ denotes the sigmoid function.

The gated messages are aggregated to update node features:

$$\mathbf{h}_i^{(l+1)} = \mathbf{h}_i^{(l)} + \phi_h \left(\mathbf{h}_i^{(l)}, \sum_{j:(j,i)\in\mathcal{E}} a_{ij} \mathbf{m}_{ij} \right). \tag{6}$$

After stacking multiple EGNN layers, a permutation-invariant readout function aggregates node embeddings into a graph-level representation, which is passed through a regression head to produce the final prediction $y = f_\theta(\mathcal{G})$

All parameters θ of the base predictor are fixed during counterfactual generation. The counterfactual explainer operates by modifying the graph through a learnable masking mechanism, described in the following subsections.

3.3. Counterfactual Edge Masking

We formulate counterfactual explanations by identifying a sparse set of edges whose removal induces a desired change in the predictor output. Rather than modifying an explicit adjacency matrix, we introduce a learnable edge masking mechanism that operates directly on message passing.

Edge mask parameterization. For each undirected edge $\{i, j\} \in \mathcal{E}$, we associate a scalar parameter $p_{ij} \in \mathbb{R}$ which defines a continuous relaxation of a binary edge indicator via:

$$g_{ij} = \sigma(p_{ij}) \in (0, 1), \tag{7}$$

where $\sigma(\cdot)$ denotes the sigmoid function. The same mask value is applied to both directions to preserve symmetry.

Counterfactual message passing. Let $\mathbf{m}_{ij}$ denote the message computed by the base EGNN layer, and a_{ij} the learned edge gate of the predictor. In the counterfactual model, messages are modulated as:

$$\mathbf{m}_{ij}^{cf} = \mathbf{m}_{ij} \cdot a_{ij} \cdot g_{ij}. \tag{8}$$

This formulation preserves the original EGNN architecture while allowing the explainer to suppress the influence of selected edges. Edges with g_{ij} close to one remain active, while edges with g_{ij} close to zero are effectively removed from message passing.

For interpretability, the final counterfactual explanation is defined by a binary edge mask obtained via thresholding:

$$\hat{g}_{ij} = \mathbb{1}[g_{ij} \geq 0.5]. \tag{9}$$

The resulting graph defines the counterfactual explanation reported in all experiments.

All parameters of the base predictor are kept fixed; only the edge mask parameters $\{p_{ij}\}$ are optimized during counterfactual generation.

3.4. Optimization Objective

The goal of counterfactual generation is to induce a desired change in the predictor output while minimizing the number of modified edges. Given an input graph $\mathcal{G}$ with prediction y_{orig}, the target value is defined following Eq. 2. The prediction loss is defined as:

$$\mathcal{L}_{\text{pred}} = \big(f_\theta(\mathcal{G}_{cf}) - y_{\text{target}}\big)^2. \tag{10}$$

To encourage minimal modifications, we penalize edge removals using:

$$\mathcal{L}_{\text{graph}} = \sum_{\{i,j\} \in \mathcal{E}} (1 - g_{ij}), \tag{11}$$

which acts as a differentiable proxy for the number of removed edges. Finally, the total loss is defined as:

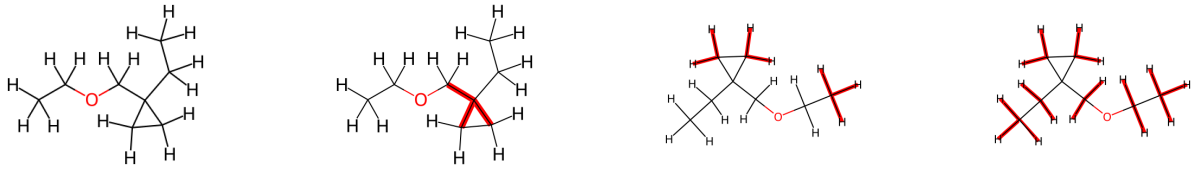
$$\mathcal{L} = (1 - \mathbb{I}_{\text{success}})\, \mathcal{L}_{\text{pred}} + \beta\, \mathcal{L}_{\text{graph}}, \tag{12}$$

where β controls the trade-off between prediction accuracy and sparsity.

4. Results

4.1. Experimental Setup

We conduct experiments on the QM9 dataset [17] using graph-level regression targets. Each input graph consists of node features, an edge list, and three-dimensional coordinates. The dataset is composed of 134K molecules and aims at predicting 19 quantum chemistry properties which are all numerical values.



(a) (b) (c) (d)

Figure 1: (a) Original isopropoxycyclohexane molecule. (b,c,d) Counterfactual explanation for isopropoxycyclohexane under a respectively $+25\%$, -25% and -50% target shift in molecular polarizability. For $+50$ target shift, no graph felt within the tolerance threshold. Highlighted edges indicate suppressed interactions in the message-passing process.

All experiments rely on the same pretrained Equivariant Graph Neural Network (EGNN) described in Section 3.2. The predictor parameters are fixed during counterfactual generation to ensure that explanations are computed for an immutable model.

Counterfactuals are generated by optimizing a sparse edge mask while keeping all model parameters frozen. Across all runs, we fix the $\tau = 0.15$, $\beta = 10^{-4}$, and use Adam (lr$= 10^{-2}$) with gradient clipping at 2.0, while varying α_{shift}.

All experiments are conducted on a single NVIDIA Quadro RTX 8000 GPU.

4.2. Effect of Target Shift Magnitude

We first provide a qualitative analysis by visualizing representative counterfactual explanations. Figure 1 shows the original isopropoxycyclohexane molecule and its counterfactual counterparts for a target shift of $\alpha_{\text{shift}} = +25\%$, -25% and -50% respectively. Counterfactuals are obtained by suppressing a minimal set of message-passing interactions, highlighted in red. These suppressed interactions are sufficient to increase the predicted molecular polarizability by approximately 25%, while preserving most of the molecular structure. The explanations provided by Figures 1c and 1d seem to indicate a correlation between the number of edges removed and the negative shift, while keeping a connected graph. On the other side, the explanations for Figure 1b result in a disconnected graph, hardly exploitable. We believe that this might indicate the need for the model to consider edge insertion as well as edge deletion.

4.3. Large-Scale Evaluation Across Test Graphs

While the previous experiment provides qualitative insight on a single molecule, we further evaluate the proposed counterfactual explainer on a large subset of the QM9 test set to assess robustness and generalization.

We generate targeted counterfactual explanations for 1,000 randomly selected test graphs and consider three molecular properties: polarizability (α), zero-point vibrational energy (zpve), and dipole moment (μ). For each property, we evaluate four target shifts $\alpha_{\text{shift}} \in \{-0.5, -0.25, 0.25, 0.5\}$.

For each configuration, we report: (i) the success rate, (ii) the mean relative error (MRE) among successful counterfactuals, and (iii) the standard deviation of the MRE.

The results in Table 1 reveal consistent patterns across properties and target shifts. For polarizability (α) and dipole moment (μ), moderate negative shifts (e.g., -25%) achieve the highest success rates, while large positive shifts ($+50\%$) are more difficult to obtain, to an extent where large negative shifts (-50%) are easier to obtain than moderate positive shifts (25%). This asymmetry suggests that decreasing these properties is generally easier than increasing them through edge suppression alone.

Interestingly, this behavior differs for zero-point vibrational energy (ZPVE). Although overall success rates are lower for ZPVE, the asymmetry between positive and negative shifts is less pronounced. ZPVE corresponds to the quantum mechanical vibrational energy present at absolute zero temperature and

Table 1
Large-scale counterfactual evaluation on 1,000 QM9 test graphs. We report success rate, mean relative error (MRE), and standard deviation for different properties and target shifts.

Property	α_{shift}	Success Rate	MRE (mean)	MRE (std)
α	-0.50	89.7%	0.359	0.358
	-0.25	98.9%	0.182	0.130
	$+0.25$	86.4%	0.118	0.078
	$+0.50$	70.7%	0.139	0.126
zpve	-0.50	42.4%	0.555	0.407
	-0.25	59.7%	0.254	0.113
	$+0.25$	61.7%	0.149	0.070
	$+0.50$	48.0%	0.167	0.136
μ	-0.50	80.9%	0.363	0.360
	-0.25	87.2%	0.187	0.126
	$+0.25$	75.5%	0.124	0.074
	$+0.50$	59.8%	0.149	0.125

primarily depends on local vibrational frequencies determined by bond stiffness and atomic masses. As a result, it is more strongly governed by local bond characteristics than by global electronic redistribution. This may explain why structural edge suppression does not produce the same directional bias observed for α and μ, suggesting that geometric interaction patterns play a comparatively less dominant role for this property.

5. Conclusion

To conclude, in this paper we aimed at employing counterfactual explanations in a molecular regression problem. By shifting the regression value of a predefined percentage, we managed to generate different explanations, highlighting the most important bonds in the molecule. The resulting graph can however not be considered as a molecule yet, and in future work we aim at ensuring that the resulting graph is a valid molecule, in order to be used as an AI for Science tool. Another idea is also to not consider a shift percentage, but the least number of editions that lead to the most important shift. Last but not least, considering other types of editions is mandatory, such as edge insertions or node insertions/deletions/substitutions.

References

[1] T. Spooner, D. Dervovic, J. Long, J. Shepard, J. Chen, D. Magazzeni, Counterfactual explanations for arbitrary regression models, arXiv preprint arXiv:2106.15212 (2021).

[2] S. S. Hada, M. Á. Carreira-Perpiñán, Exploring counterfactual explanations for classification and regression trees, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 489–504.

[3] T. Kipf, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint arXiv:1710.10903 (2017).

[5] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, arXiv preprint arXiv:1810.00826 (2018).

[6] V. G. Satorras, E. Hoogeboom, M. Welling, E (n) equivariant graph neural networks, in: International conference on machine learning, PMLR, 2021, pp. 9323–9332.

[7] M. A. Prado-Romero, B. Prenkaj, G. Stilo, F. Giannotti, A survey on graph counterfactual explanations: definitions, methods, evaluation, and research challenges, ACM Computing Surveys 56 (2024) 1–37.
[8] J. Chen, S. Wu, A. Gupta, R. Ying, D4explainer: In-distribution explanations of graph neural network via discrete denoising diffusion, Advances in Neural Information Processing Systems 36 (2023) 78964–78986.
[9] Y. Sun, A. Valente, S. Liu, D. Wang, Preserve, promote, or attack? gnn explanation via topology perturbation, arXiv preprint arXiv:2103.13944 (2021).
[10] J. Ma, R. Guo, S. Mishra, A. Zhang, J. Li, Clear: Generative counterfactual explanations on graphs, Advances in neural information processing systems 35 (2022) 25895–25907.
[11] D. Numeroso, D. Bacciu, Meg: Generating molecular counterfactual explanations for deep graph networks, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.
[12] T. M. Nguyen, T. P. Quinn, T. Nguyen, T. Tran, Explaining black box drug target prediction through model agnostic counterfactual samples, IEEE/ACM Transactions on Computational Biology and Bioinformatics 20 (2022) 1020–1029.
[13] A. Lucic, M. A. Ter Hoeve, G. Tolomei, M. De Rijke, F. Silvestri, Cf-gnnexplainer: Counterfactual explanations for graph neural networks, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 4499–4511.
[14] Z. Ying, D. Bourgeois, J. You, M. Zitnik, J. Leskovec, Gnnexplainer: Generating explanations for graph neural networks, Advances in neural information processing systems 32 (2019).
[15] J. Tan, S. Geng, Z. Fu, Y. Ge, S. Xu, Y. Li, Y. Zhang, Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning, in: Proceedings of the ACM web conference 2022, 2022, pp. 1018–1027.
[16] R. Cai, Y. Zhu, X. Chen, Y. Fang, M. Wu, J. Qiao, Z. Hao, On the probability of necessity and sufficiency of explaining graph neural networks: A lower bound optimization approach, Neural Networks 184 (2025) 107065.
[17] R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. Von Lilienfeld, Quantum chemistry structures and properties of 134 kilo molecules, Scientific data 1 (2014) 1–7.