# Confidence-Aware Explanations for 3D Molecular Graphs via Energy-Based Masking

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Graph Neural Networks (GNNs) have become a powerful tool for modeling molecular data. To improve their reliability and interpretability, various explanation methods aim to identify key molecular substructures, typically a subset of edges, influencing the decision-making process. Early work on 2D GNNs represented molecules as graphs with atoms as nodes and bonds as edges, ignoring 3D geometric configurations. While existing explanation methods perform well for 2D GNNs, there is a growing demand for 3D explanation techniques suited to 3D GNNs, which often surpass 2D GNNs in performance. Current explanation methods struggle with 3D GNNs due to the construction of edges based on distance cutoffs, leading to an exponential increase in the number of edges a molecular graph possesses. We identify key sources of explanation errors and decompose them into two components, derived from an upper bound between optimized masks and the true explanatory subgraph. This gap is especially significant in 3D GNNs because of the dense edge structures. To improve explanation fidelity, our method assigns two energy values to each atom, representing its contribution to predictions: one for importance and one for non-importance. The explanation model becomes more confident when the distinction between importance and non-importance is clearer. Analogous to physics, lower energy values indicate greater stability, enhancing confidence in the associated scenario. By optimizing these energy values to distinguish the two cases, we minimize both components of the error bound and identify a stable subgraph with high explanation fidelity. Experiments with various 3D backbone models on widely used datasets are conducted to validate our method's effectiveness in providing accurate and reliable explanations for 3D molecular graphs.

## 1 Introduction

In recent years, molecular learning has emerged as a crucial area of study, driving advances in drug discovery, protein engineering, and materials science (Fout et al., 2017; Butler et al., 2018; Wu et al., 2018; Gupta et al., 2021; Zhang et al., 2025). Traditionally, molecules have been represented as 2D planar graphs, where atoms serve as nodes and chemical bonds are depicted as edges without considering the geometric configurations. The limitations of 2D representations in capturing molecular properties have led to a growing focus on 3D graph representations (Schütt et al., 2017; Wang et al., 2022; Liu et al., 2022; Satorras et al., 2021; Gasteiger et al., 2021) that represent entities with spatial coordinates, enabling them to capture complex spatial dependencies that are critical for tasks involving 3D molecular structures. This shift is critical because the 3D structure of molecules, particularly their spatial arrangement, directly influences their chemical behaviors and biological functions. In response, 3D GNNs have been rapidly developed and shown to outperform their 2D counterparts in numerous tasks (Schütt et al., 2017; Anderson et al., 2019; Fuchs et al., 2020; Batzner et al., 2022; Satorras et al., 2021; Wang et al., 2022; Liu et al., 2022; Jamasb et al., 2024).

As GNNs have shown great results in molecular learning, the need for explainability and interpretability has become increasingly important. Molecular systems are inherently complex, and GNNs are often treated as black-box models, making it difficult to understand how specific structural features contribute to predictions, which raises significant concerns regarding transparency in the decision-making process. GNN explanation methods aim to illuminate these decision-making processes by identifying key substructures of the graph (the

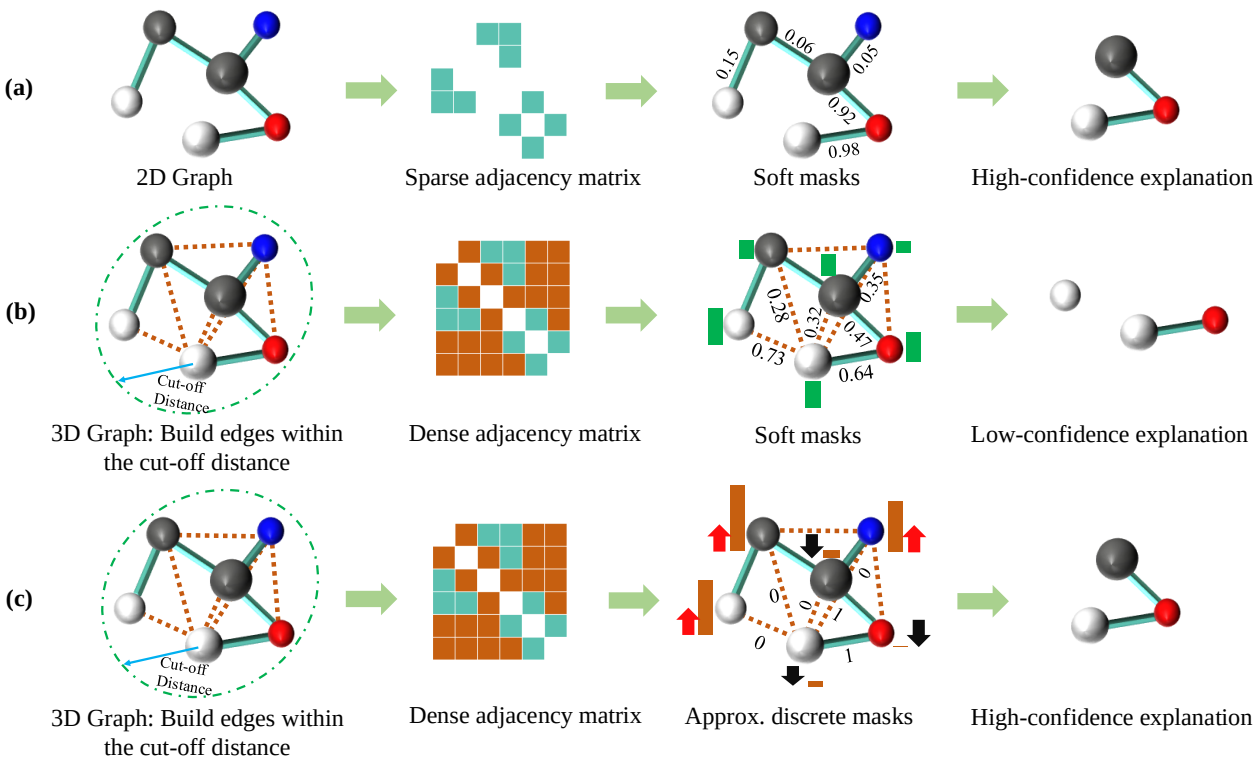


Figure 1: An illustration of the structural differences between 2D and 3D GNN explanations, as well as the challenges posed by existing methods. In the third column, the bars represent the energies for nodes and the arrows represent the operations to "push up" and "push down" the energies, while the numbers correspond to the edge masks derived from the energies of the nodes. Specifically, due to the differing assumptions in 3D GNNs, our aim is to identify a subset of nodes for the explanation. However, using soft masks typically results in explanations of low confidence, which undermines explanation fidelity. To address this issue, we assign energies to each node. By pushing up and down these energies, we can obtain approximately discrete masks that provide more confident explanations. Further details are provided in Sec. 3.

molecule) that influence the model's predictions. Groundbreaking research has addressed these challenges, advancing our understanding of graph learning mechanisms across various contexts (Ying et al., 2019; Yuan et al., 2020; Luo et al., 2020; Pope et al., 2019; Huang et al., 2023).

While existing explanation methods are effective for 2D GNNs, there is an urgent need for explanation techniques specifically designed for 3D GNNs. Current methods face challenges with 3D GNNs due to the construction of edges based on cut-off distances, leading to an exponentially large number of edges (Schütt et al., 2017; Gasteiger et al., 2020a; Wang et al., 2022; Liu et al., 2022; Satorras et al., 2021). Our experiments in Sec. 4 clearly demonstrate the inability of 2D methods to effectively explain 3D GNNs. In our study, we identify the sources of errors in explanations and break them down into two components, informed by a derived upper bound that relates the optimized masks to the actual subgraph. This gap is particularly significant for 3D GNNs due to the large volume of edges involved. In 2D GNN explanations, there are typically at most a few edges with mask values around 0.5, indicating uncertainty about their inclusion or exclusion in the final explanatory graph. However, in 3D GNN explanations, the number of uncertain edges grows rapidly, leading to suboptimal results and ambiguous explanation results that complicate the decision-making process rather than explaining it.

To enhance explanation fidelity, our method aims to bridge this gap by assigning two energy values to each atom in a 3D molecular graph. One energy reflects the scenario where this node is important in making the decision, while the other represents the scenario where it is unimportant. Drawing an analogy to physics, we assert that nodes with lower energy values correspond to greater stability in the explanatory results; thereby,

we are more confident about the scenario with which it is associated. Current explanation models (Ying et al., 2019; Luo et al., 2020; Miao et al., 2022) only optimize the first term in our derived bound leading to *over-smoothing of the soft masks*. Our approach seeks to push the lower energy down and push the higher energy up to simultaneously optimize both components of the derived error bound, thereby reducing discrepancies between the identified explanatory subgraph and the associated edge masks. By achieving a lower energy state, we can accurately and confidently identify a stable subgraph that exhibits high explanation fidelity. An illustration of our method to mitigate challenges caused by the key structural differences between 2D and 3D GNN explanations is presented in Fig. 1. Experiments are conducted on several backbone networks and two widely used datasets, the QM9 (Ramakrishnan et al., 2014) and the EC (Wang et al., 2023), validate the efficacy of our method, demonstrating its capacity to deliver accurate, stable, and reliable explanations for 3D molecular graphs.

We summarize our contributions as follows:

- Leveraging the structural differences between 2D and 3D GNNs, we reformulate graph explanations specifically for 3D GNNs.
- We establish an error bound for graph explanations, dividing them into two components: the first is the focus of existing methods, while the second has been overlooked.
- Based on the derived upper bound, we introduce Energy-based Discrete Mask Approximation to address this bottleneck, optimizing both components simultaneously.
- Experimental results demonstrate that our method is effective and highly generalizable for explaining 3D GNNs.

## 2 Background and Related Work

In this section, we begin by presenting the formal definition of the graph explanation task in Sec. 2.1. Following that, in Sec. 2.2, we provide a comprehensive review of the key methodologies that have been proposed to generate explanations. Finally, Sec. 2.3 delves into the definition and formulation of Energy-Based Models (LeCun et al., 2006), outlining their role in enhancing the interpretability of GNNs and their applications in providing insights into molecular structures and behaviors.

### 2.1 Graph Explanation

A 2D molecular graph $G$ is represented as $G = (\mathcal{V}, \boldsymbol{X}, \mathbf{A})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ denotes a set of $n$ nodes. $\boldsymbol{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}]^T \in \mathbb{R}^{n \times d_v}$ is the node feature matrix with each $\mathbf{x_i} \in \mathbb{R}^{d_v}$, where $d_v$ represents the dimension of the node features. $\mathbf{A} = \{a_{ij} \mid i, j \in \mathcal{V} \text{ and } i \neq j\} \in \{0,1\}^{n \times n}$ is the adjacency matrix, where each $a_{ij} \in \{0,1\}$ denotes whether there is an edge from node $i$ to node $j$. Graph neural networks (GNNs) utilize the edge set $\mathbf{A}$ to facilitate message passing and aggregation between nodes. A GNN $\Phi$ is a mapping from a graph $G$ to a prediction $\hat{Y}$ in relation to the target variable $Y$. This target can represent discrete labels in a graph classification task or continuous values in a regression task. In this study, we present the method on the graph regression setting without loss of generality. The experiments in Sec. 4 cover both regression and classification tasks.

**Graph Explanation:** Following the definition in Ying et al. (2019), the objective of instance-level graph explanation is to identify a subgraph $G_S \subseteq G$ that is important to the target $Y$. This is formally expressed as:

$$G_S^* = \underset{G_S \subseteq G}{\arg\min}\, \mathcal{L}(Y; \Phi(G_S)) \quad \text{s.t.} \quad |G_S| \leq B, \tag{1}$$

where $\mathcal{L}$ denotes the task-dependent loss function, and $B$ represents a size constraint on the subgraph to avoid trivial solutions. Eq. (1) can be rewritten as:

$$\begin{aligned} G_S^* &= \underset{\mathbf{M}}{\arg\min}\ \mathcal{L}(Y; \Phi(\mathbf{X}, \mathbf{M} \odot \mathbf{A})) \\ \text{s.t.} \quad \mathbf{M} &\in \{0,1\}^{n \times n}, \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{M}_{ij} \leq B. \end{aligned} \tag{2}$$

Directly solving Eq. (2) leads to a computationally intractable combinatorial optimization problem with complexity $O(2^n)$. Exiting works relaxed the discrete (hard) masks with discrete values 0 and 1 to soft masks with values between 0 and 1:

$$\begin{aligned} G_S^* &= \arg\min_{\mathbf{M}'} \ \mathcal{L}(Y; \Phi(\mathbf{X}, \mathbf{M}' \odot \mathbf{A})) \\ \text{s.t.} \quad \mathbf{M}' &\in [0,1]^{n \times n}, \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{M}'_{ij} \leq B. \end{aligned} \tag{3}$$

Such relaxation enables gradients to be back-propagated; thus, gradient descent can be used to efficiently solve this problem.

### 2.2 Existing Explanation Methods

GNN explanation methods can be classified according to several criteria: transductive or inductive explanations, instance-level explanations (Ying et al., 2019; Schlichtkrull et al., 2021) versus model-level explanations (Yuan et al., 2020), model-specific approaches (Pope et al., 2019; Dai & Wang, 2021) compared to model-agnostic methods (Luo et al., 2020), and node-level (Ying et al., 2019; Pope et al., 2019) versus graph-level explanations (Yuan et al., 2020). In terms of explanation strategies, four primary categories emerge: (1) Gradient-based methods (Zhou et al., 2016) compute the gradients of target predictions with respect to inputs via back-propagation but often impose structural constraints on GNNs; (2) Decomposition methods (Pope et al., 2019) assign importance scores to input features by analyzing model parameters to reveal relationships between inputs and outputs; (3) Surrogate methods (Huang et al., 2023) use interpretable models to explain the behavior of complex GNNs, though they often encounter difficulties with the discrete and topological nature of 3D graphs; (4) Perturbation-based methods (Ying et al., 2019; Luo et al., 2020) identify important subgraphs by perturbing edges or nodes with masks and analyzing output prediction changes.

### 2.3 Energy-based Model

The central concept of Energy-Based Models (EBMs) is the use of Boltzmann distributions to assess the likelihood of input samples. This involves defining a function $\mathcal{E}(\mathbf{x_i}) : \mathbb{R}^{d_v} \to \mathbb{R}$ that assigns a non-probabilistic scalar known as *energy* to each configuration of the input data. Influential works (Xie et al., 2016) have significantly shaped research in this domain. EBMs have achieved notable success in various applications, including classification (Li et al., 2022; Grathwohl et al., 2019), regression tasks (Danelljan et al., 2020), structured prediction (Rooshenas et al., 2019; Belanger & McCallum, 2016), and out-of-distribution (OOD) detection (Liu et al., 2020). Additionally, (Pang & Wu, 2021) have explored the use of EBMs in latent space for generation, while others have applied them to unsupervised learning (Ranzato et al., 2007) and concept-based modeling (Xu et al., 2024).

## 3 Energy-based Discrete Mask Approximation

In this section, we present the Energy-based Discrete Mask Approximation (EDMA), a principled approach for 3D GNN explanation. We begin by analyzing the differences between 2D and 3D graph explanations in Sec. 3.1. With such differences, we reformulate 3D graph explanations and identify an upper bound on the explanation loss in Sec. 3.2. In Sec. 3.3, we provide a detailed presentation of our method to simultaneously optimize all terms in the derived upper bound.

### 3.1 2D v.s. 3D Graph explanation

The primary distinction between 2D and 3D graph explanations arises from the structural differences inherent in 2D and 3D graphs. In 2D graphs, each node is associated with a set of edges that connect these nodes in a planar layout. In 3D graphs, nodes are represented in three-dimensional space, allowing for a more accurate depiction of the physical arrangement and spatial relationships between entities, while edges are typically determined from the coordinates of the nodes by a cut-off distance (Schütt et al., 2017; Gasteiger et al., 2020a; Wang et al., 2022; Liu et al., 2022).

More concretely, for small molecular structures, the number of bonds (edges) between atoms (nodes) is typically limited, resulting in a rather sparse graph. However, 3D GNNs do not utilize chemical bonds as edges; instead, the 3D spatial configurations of nodes are used to construct edges, resulting in an exponentially large number of edges. As a result, 3D GNN explanation poses significant challenges to existing methods, partially illustrated in Fig. 1 and detailed below:

1. **Differing Assumptions in 2D and 3D Explanations**: The assumptions underlying graph explanations differ between 2D and 3D graphs. For instance, in 2D graph explanations, methods such as those proposed by (Ying et al., 2019; Luo et al., 2020) assume that the graph being explained is a random graph (ERDdS & R&wi, 1959), with edges considered independent of one another. However, this assumption does not hold for complex networks like molecular systems, where force field theory models both intramolecular interactions and intermolecular terms, contributing to the total energy of a molecule (Leach, 2001).

2. **Dense Adjacency Matrix**: The adjacency matrix $\mathbf{A}$ in a 3D graph is typically dense, unlike the sparse adjacency matrix commonly observed in a 2D graph. As one can imagine, this leads to a problem of combinatorial complexity with respect to the number of edges with discrete masks. Even with soft mask relaxation, the large number of edges introduces a substantial lack of confidence in identifying the explanatory subgraph, often resulting in suboptimal explanation outcomes. Specifically, with low confidence in distinguishing important and unimportant sub-parts, the soft-masked "subgraph" deviates a lot from the final discrete explanatory sub-graph. The optimization process might find a soft-masked "sub-graph" with minimal loss in Eq. (3); however, when we decide the explanatory subgraph from soft-masks, the lack of confidence in the soft masks leads to poor final explanation performance.

### 3.2 Reformulating 3D Graph Explanation

The inputs to 3D GNNs consist of nodes with 3D spatial coordinates, and edges are constructed based on this spatial information. The common assumption of random graph structures no longer holds in this context. Simply applying current graph explanation methods without accounting for the structural differences is unlikely to yield explanations of scientific meanings, which is demonstrated in our experiments in Sec. 4 where methods not specifically designed for 3D GNNs yield worse performance.

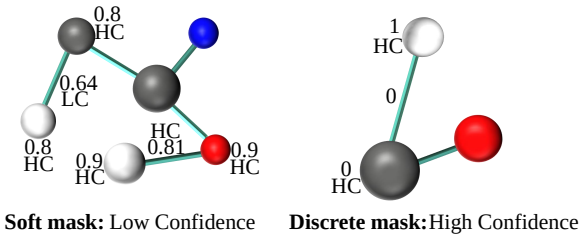


Figure 2: A comparison between soft masks and discrete masks. Numbers are mask values, **HC** denotes High Confidence, and **LC** denotes Low Confidence. The edge masks used for message passing are constructed from node masks. However, soft masks can lead to great discrepancies between the optimization objective and the final explanatory subgraph, as indicated in Eq. (5).

*As discussed in the first challenge outlined in Sec. 3.1, we should define the explanatory substructure to be a subset of nodes.* To this end, we place masks on the nodes, which are then transformed into edge masks. For node $i$, there will be an associated soft-mask value $m_i \in [0, 1]$, and $\mathbf{m} \in [0, 1]^n$ denotes the set of all node masks. The edge masks are then constructed by $\mathbf{M}^{'} = \mathbf{m} \otimes \mathbf{m}$, where $\otimes$ denotes the outer product. Specifically, Eq. (3) can be rewritten as:

$$\begin{aligned} G_S^* &= \arg\min_{\mathbf{M}'} \mathcal{L}(Y; \Phi(\mathbf{X}, \mathbf{M}^{'} \odot \mathbf{A})) \\ \text{s.t.} \quad & \mathbf{m} \in [0, 1]^n, \quad \mathbf{M}^{'} = \mathbf{m} \otimes \mathbf{m}, \quad \|\mathbf{m}\|_1 \leq K, \end{aligned} \tag{4}$$

where $\|\mathbf{m}\|_1 = \sum_{j=1}^{n} \mathbf{m}_i$ is the 1-norm of $\mathbf{m}$, $K$ is the budget on the number of nodes in the final explanatory substructure. To this end, we would like to introduce a shortcoming in relaxing discrete masks to soft-masks: The discrepancy between the optimized soft-masked "subgraph" and the final explanatory subgraph. Mathematically,

$$G_S^* = \arg\min_{G_S \subseteq G} \mathcal{L}(Y; \Phi(G_S)) \leq \underbrace{\mathcal{L}(Y; \Phi(\mathbf{X}, \mathbf{M}^{'} \odot \mathbf{A}))}_{\text{soft-mask explanation loss}} + \underbrace{\mathcal{L}(\Phi(\mathbf{X}, \mathbf{M}^{'} \odot \mathbf{A}); \Phi(\mathbf{X}, \mathbf{M} \odot \mathbf{A}))}_{\text{discrepancy between soft and discrete masks}}. \tag{5}$$

In this bound, the first term represents the soft-mask explanation loss in the relaxed optimization Eq. (3), which we solve through gradient descent. The second term depicts the discrepancy between soft and discrete masks, and this has been overlooked in existing GNN explanation methods. Existing studies term this issue as "introduced evidence". Any value in masks that is not strictly zero or one can introduce new semantics or noise into the explanation, potentially impacting the results (Dabkowski & Gal, 2017). For instance, even if the value of $\mathbf{M}^{'}{}_{ij}$ is small, the edge $e_{ij}$ may still facilitate message passing between node $i$ and $j$. We will refer to this as the confidence of the soft masks, where a mask value close to 0 or 1 indicates high confidence about the substructure's contribution to the decision-making process. *While this bound generally applies to both 2D and 3D GNNs, 3D GNNs suffer much more from this issue for reasons given in the second challenge outlined in Sec. 3.1.* In 3D GNNs, there are exponentially many edges, and the accumulation of information passed during message passing can significantly influence the explanation results even with edge masks of high confidence. Even worse, due to the intrinsic nature of 3D GNNs, the node masks will largely decrease the confidence and stability in final explanations as illustrated in Fig. 2. To this end, we are ready to present the Energy-based Discrete Mask Approximation method to mitigate this issue.

### 3.3 EDMA for Confident 3D Graph Explanation

We now present our method EDMA for confident 3D graph explanation that simultaneously minimizes both terms in Eq. (5).

Instead of using soft masks for the selection of explanatory nodes, we treat the selection of nodes as states within a system, where the energy levels of these states determine their probability of being part of the explanatory subgraph. The EBM function $\mathcal{E}(\mathbf{e_i}) : \mathbb{R}^d \to \mathbb{R}$ maps the node embedding $\mathbf{e_i}$ to a scalar value known as *energy*. Following Liu et al. (2020), the energy for a node with respect to class $c$ is defined as $\mathcal{E}(\mathbf{e_i}, c) = \mathcal{E}_c(\mathbf{e_i}) = \frac{-\phi_c(\mathbf{e_i})}{T}$, where $\phi_c$ extracts logits for class $c$ and $0 < T < 1$ serves as a control hyper-parameter analogous to the temperature in physics. It will push up the larger energy and push down the smaller energy. To see this, suppose $\phi_0 = 2$ and $\phi_1 = 5$ then the difference between them is 3. With $T = 0.1$, the energies will be 20 and 50, respectively, and the difference between them is 30. With a smaller value of $T$, we further amplify the difference between these two energies, leading to a more confident explanation with our explainer as defined in Eq. (6). This process is analogous to the temperature in physics that, when $T$ is small, the system is in a low-energy state, leading to probabilities closer to 0 or 1; in other words, a more confident selection of nodes.

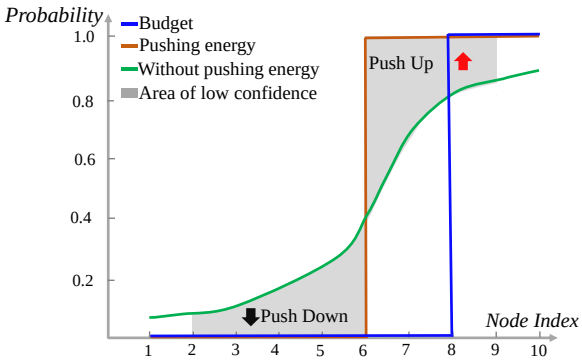


Figure 3: An illustration of the effects of the explainer function $f$. The node indices are arranged based on their probability values. By pushing up and down the energies, the masks become approximately discrete, enhancing confidence in the explanatory substructure. Moreover, varying the values of the stretching parameters ($\gamma$, $\zeta$ in Eq. (6)) enables us to better control the budget.

Then, an explainer function $f$ is used to compute the probabilities that each node $i$ belongs to the explanatory subgraph, represented as $P_i(c = 1) = f(\mathcal{E}_\mathbf{0}(\mathbf{e_i}), \mathcal{E}_1(\mathbf{e_i})))$. Inspired by the hard concrete distribution (Louizos et al., 2017), $f : \mathbb{E} \times \mathbb{E} \mapsto [0, 1]$, with $\mathbb{E}$ being the potential energy state space for all nodes, is a function

that takes both energies and produces a single scalar value indicating probabilities defined as:

$$\begin{aligned} & f\big(\mathcal{E}_0(\mathbf{e_i}), \mathcal{E}_1(\mathbf{e_i})\big) \\ = \quad & \min\Big(1, \max\big(0, \frac{1}{1+e^{\mathcal{E}_0(\mathbf{e_i})-\mathcal{E}_1(\mathbf{e_i})}}(\zeta-\gamma)+\gamma\big)\Big), \end{aligned} \tag{6}$$

where $\gamma < 0$, $\zeta > 1$ are hyper-parameters to stretch the probability to the interval $(\gamma, \zeta)$ and then truncate the value to the range $[0, 1]$. Together with the temperature, this will help us obtain more confident explanations in terms of having all the probabilities close to either 0 or 1 and better control the budget $K$ as we can set appropriate values of $\gamma$ and $\zeta$ to obtain the desired number of probabilities closer to 1. In Fig. 3, we illustrate how our energy-based explainer function generates approximately discrete masks while effectively managing the budget. Our explainer function takes energy values as input and outputs the probability that a node belongs to the explanatory substructure. By increasing the energies of important nodes and decreasing those of unimportant ones, we enhance their distinction, resulting in approximately discrete masks that yield more confident explanatory substructures. Furthermore, the stretching parameters in Eq. (6) regulate the number of nodes for which energies are pushed up.

Without loss of generality, we denote $m_i = f\big(\mathcal{E}_\mathbf{0}\left(\mathbf{e_i}\right), \mathcal{E}_1\left(\mathbf{e_i}\right)\big)$ as the explanation mask. Finally, the mutual information term in Eq. (4) and the explainer function in Eq. (6) are then jointly optimized to classify nodes and determine whether they belong to the explanatory subgraph. The final optimization function for our proposed method is as follows:

$$\mathcal{L}_{final} = \mathcal{L}(Y; \Phi(\mathbf{X}, \mathbf{M}^{'} \odot \mathbf{A})) + \alpha \| f(\mathcal{E}_\mathbf{0}(\mathbf{e_i}), \mathcal{E}_1(\mathbf{e_i})))\|_1, \tag{7}$$

where $\alpha$ is a parameter that balances the information loss and the explainer function loss that controls the budget. With this particular formulation, we simultaneously optimize both terms in our derived bound in Eq. (5), leading to approximately discrete, i.e., confident, probabilities for the inclusion or exclusion of a certain node in the final explanatory subgraph.

### 3.4 Discreteness by Design v.s. Regularization

A common alternative to promote discreteness is the introduction of regularization terms to constrain deviations of the masks from discreteness. This approach is used in most GNN explanation methods, e.g., GNNExplainer (Ying et al., 2019) and PGExplainer (Luo et al., 2020). Specifically, the optimization objective is:

$$G_S^* = \arg\min_{\mathbf{M}'} \mathcal{L}(Y; \Phi(\mathbf{M}^{'} \odot \mathbf{A}, \mathbf{X})) + \alpha \cdot \|\mathbf{M}^{'}\|_1 + \beta \cdot \mathbb{H}[\mathbf{M}^{'}], \tag{8}$$

where $\alpha$ and $\beta$ are coefficients that balance different losses. $\alpha \cdot \|\mathbf{M}^{'}\|_1$ enforces the budget and $\beta \cdot \mathbb{H}[\mathbf{M}^{'}]$ represents the entropy of the mask values, ensuring that they remain close to binary. Most existing explanation methods adopt this setting and include the entropy loss to promote discreteness.

However, this regularization-based approach has several limitations. First, the introduction of additional loss terms creates a trade-off between enforcing discreteness and optimizing the primary objective, requiring careful hyperparameter tuning to achieve an appropriate balance. Poorly chosen regularization strengths can lead to suboptimal solutions where the mask is neither sufficiently discrete nor effective in explaining the model's predictions. Second, regularization introduces additional complexity into the optimization process; for example, competing loss terms can create conflicting gradients, making convergence slower and less stable. In contrast, our method enforces discreteness by design; it directly enforces binary mask constraints without relying on auxiliary regularization terms. These complications become worse for 3D graph explanations due to exponentially many edges, making the regularization-based approach highly ineffective for 3D graphs. As demonstrated in the ablation study in Sec. 4.4, removing the entropy loss does not result in noticeable performance degradation; the performance gains from our method significantly outweigh the degradation from the absence of entropy loss in existing explainers, demonstrating the superiority of our method as a principled approach to enforce discreteness by design.

# 4 Experimental Studies

We begin by outlining the experimental setup in Sec. 4.1. Sec. 4.2 presents a comparative analysis of the quantitative results of our method against baseline approaches. In Sec. 4.3, we offer a qualitative analysis to further illustrate the interpretability and effectiveness of the proposed method. Finally, an ablation study is conducted in Sec. 4.4 to evaluate the contributions and significance of various components.

## 4.1 Experimental Setup

**Dataset and Backbone 3D GNNs.** In this work, we conduct experiments on two widely used datasets—the QM9 (Ramakrishnan et al., 2014) dataset and the EC dataset (Wang et al., 2023). QM9 is a comprehensive 3D molecular dataset frequently used to predict various molecular properties; we use the version available in PyTorch Geometric (PyG) directly, along with its predefined training and test splits. Our study targets the prediction of two key properties: the dipole moment ($\mu$) and the free energy at 298.15K (denoted as $G_f$ to avoid confusion with the graph notation $G$). EC is a multi-classification task (60 classes) for *enzymes* that classifies enzyme commission (EC) numbers based on the reactions catalyzed by enzymes. Compared to the QM9 dataset, the EC dataset is a classification task, instead of regression, with larger and more complex graph structures, *demonstrating the generalization ability of our proposed method*. The EC dataset contains $5,481$ graphs, which are split into $4,149$ for training and $1,332$ for testing. We adopt the pretrained SchNet (Schütt et al., 2017) and DimeNet++ (Gasteiger et al., 2020b;a) from PyG as the backbone 3D GNN models for the QM9 dataset. The EC dataset is a multi-class classification task for *proteins*, whereas SchNet and DimeNet++ for the QM9 dataset are designed specifically for small molecules. Therefore, we adopt ProNet (Wang et al., 2023), a well-known 3D GNN for protein learning, as the backbone model.

**Baselines.** We compare our approach against several state-of-the-art baselines. GNNExplainer (Ying et al., 2019) and PGExplainer (Luo et al., 2020) are ***leading explanation methods for 2D GNNs***, designed for transductive and inductive tasks, respectively. However, due to structural differences discussed in Sec. 3.1, these methods are not directly applicable to 3D GNNs. To adapt them for 3D molecular graphs, we place masks on nodes, generate edges in a manner similar to our approach, and use these to perturb node embeddings and generate explanations. These adapted methods, referred to as GNNExplainer-Dense and PGExplainer-Dense, serve as key baselines for evaluating our proposed method.

In addition, we include LRI (Miao et al., 2022) in our comparisons, as it is currently the only method specifically designed for geometric graph explanations. We employ the LRI-Bernoulli variant, which identifies key nodes relevant to downstream regression tasks, making it a strong baseline for explaining 3D molecular graph data. All baseline methods are implemented using PyG with necessary adjustments to ensure consistency in the experimental setup.

**Evaluation Metrics** Following standard practices, for the regression task with the QM9 data, we use Mean Absolute Error (MAE) to evaluate the performance of 3D GNN predictions against ground-truth molecular properties. A lower MAE indicates better results. In the context of explaining 3D molecular graphs, let $\text{MAE}_W$ represent the prediction error using the entire graph, while $\text{MAE}_S$ denotes the prediction error using the optimal subgraph selected for explanation, as described in Eq. (2). Naturally, $\text{MAE}_S$ is expected to be higher than $\text{MAE}_W$, as the complete graph generally yields better predictions than the subgraph, given the same pretrained 3D GNN model (such as SchNet or DimeNet++). We define explanation fidelity as $\text{Fidelity}^- = \text{MAE}_S - \text{MAE}_W$, which measures the quality of explanations produced by different methods. ***A lower Fidelity$^-$ indicates that the method provides a more accurate and reliable explanation ($-$ indicates that lower is better).***

Similarly, for the classification task with the EC data, we use accuracy to evaluate the performance of 3D GNN predictions against ground-truth molecular labels. Following Faber et al. (2021), to ensure a more reliable comparison of results, we only select molecular graphs that have been correctly classified by the backbone model for explanation, thereby excluding the influence of misclassified instances. We define explanation fidelity $\text{Fidelity}^+$ as the percentage of explanatory subgraphs that produce the same labels as the original whole graphs. ***A higher Fidelity$^+$ indicates that the method provides a more accurate and reliable explanation ($+$ indicates the higher the better).*** Note that the reported results are averaged across all molecules in the test set. Since the standard deviation is two orders of magnitude smaller

than the average fidelity, we omit it from our reported results. Further details about experimental settings are presented in the Appendix.

### 4.2 Quantitative Results

**Regression Task on the QM9 Dataset.** We conduct experiments to demonstrate the effectiveness of our method by comparing it to baseline approaches on the QM9 dataset. The results using SchNet are presented in Tables 1 and 2. Notably, all baseline methods select the top-$k$ nodes as explanations, with $k$ ranging from 2 to 9. Our method consistently outperforms these baselines, producing explanatory subgraphs with the lowest explanation fidelity (where lower fidelity indicates better performance). This suggests that by optimizing the two components based on the derived upper bound, we achieve a closer alignment between the desired discrete masks and the approximate discrete masks generated via the EBM. By appropriately controlling the loss function, we can either increase or decrease the energy of each atom, amplifying the distances between explanatory and non-explanatory parts, making them easier to identify. Similar results are observed in Tables 3 and 4 using DimeNet++ as the backbone.

Table 1: Explanation Fidelity$^-$ (the lower the better) for all baseline methods and our proposed EDMA method using SchNet as the backbone on the property $\mu$ (dipole moment). The best results are highlighted in bold.

| **Top-$k$** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|
| GNNExplainer-Dense | 3.88 | 5.62 | 7.28 | 8.05 | 8.27 | 8.00 | 7.59 | 6.87 |
| PGExplainer-Dense | 2.91 | **3.73** | 4.83 | 6.09 | 6.62 | 6.55 | 6.81 | 6.08 |
| LRI-Bernoulli | 3.50 | 4.84 | 6.16 | 6.88 | 7.10 | 7.29 | 7.43 | 7.32 |
| **EDMA** | **2.74** | **3.73** | **4.31** | **4.83** | **5.08** | **5.47** | **5.72** | **5.31** |

Table 2: Explanation Fidelity$^-$ (the lower the better) for all baseline methods and our proposed EDMA method using SchNet as the backbone on the property $G_f$ (atomization free energy). The results are presented in units scaled by a factor of 1000. The best results are highlighted in bold.

| **Top-$k$** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|
| GNNExplainer-Dense | 9.66 | 8.48 | 7.24 | 6.03 | 4.78 | 3.51 | 2.26 | 1.09 |
| PGExplainer-Dense | 10.26 | 9.56 | 8.68 | 7.74 | 6.52 | 5.40 | 4.21 | 3.94 |
| LRI-Bernoulli | 9.39 | 8.39 | 7.38 | 6.59 | 5.93 | 5.31 | 4.78 | 4.09 |
| **EDMA** | **8.66** | **7.45** | **6.23** | **5.07** | **3.74** | **2.55** | **1.36** | **0.21** |

Table 3: Explanation Fidelity$^-$ (the lower the better) for all baseline methods and our proposed EDMA method using DimeNet++ as the backbone on the property $\mu$ (dipole moment). The best results are highlighted in bold.

| **Top-$k$** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|
| GNNExplainer-Dense | 2.50 | 2.29 | 2.07 | 1.80 | 1.53 | 1.27 | 1.04 | 0.82 |
| PGExplainer-Dense | 2.52 | 2.46 | 2.13 | 2.07 | 1.84 | 1.67 | 1.50 | 1.40 |
| LRI-Bernoulli | 2.58 | 2.42 | 2.22 | 2.04 | 1.87 | 1.69 | 1.51 | 1.40 |
| **EDMA** | **2.40** | **2.07** | **1.76** | **1.47** | **1.22** | **1.02** | **0.85** | **0.71** |

**Classification Task on the EC Dataset.** Additionally, we conduct experiments on the EC dataset with ProNet (Wang et al., 2023) as the backbone model. The results are presented in Table 5 with budget $k$ defined over the range $[80, 160]$ with a step size of 10. Our proposed method outperforms all the baselines by a large margin. Only our method can faithfully explain more than half of the graphs in the dataset, which clearly illustrates the superiority of our proposed method in generalizing to larger and more complex structures.

### 4.3 Qualitative Results

In this section, we present qualitative results of the baselines and our approach in finding the important substructures that contribute to the properties of the molecules the most. Functional groups play a significant role in determining the chemical properties of molecules; thus, an explanation method that generates

Table 4: Explanation Fidelity$^-$ (the lower the better) for all baseline methods and our proposed EDMA method using DimeNet++ as the backbone on the property $G_f$ (atomization free energy). The best results are highlighted in bold.

| **Top-$k$** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|
| GNNExplainer-Dense | 65.74 | 62.56 | 58.91 | 54.65 | 49.15 | 43.18 | 36.69 | 30.53 |
| PGExplainer-Dense | 64.32 | 61.25 | 55.33 | 51.64 | 46.13 | 40.33 | 34.35 | 29.41 |
| LRI-Bernoulli | 64.76 | 60.46 | 55.29 | 49.97 | 44.66 | 39.41 | 34.19 | 29.49 |
| **EDMA** | **63.83** | **59.42** | **54.48** | **49.26** | **43.84** | **38.15** | **33.12** | **28.47** |

Table 5: Explanation Fidelity$^+$ (the higher the better) for all baseline methods and our proposed EDMA method using ProNet as the backbone on the EC dataset. The best results are highlighted in bold.

| **Top-$k$** | **80** | **90** | **100** | **110** | **120** | **130** | **140** | **150** | **160** |
|---|---|---|---|---|---|---|---|---|---|
| GNNExplainer-Dense | 24.32 | 26.35 | 29.13 | 30.48 | 35.96 | 36.71 | 43.09 | 46.70 | 49.70 |
| PGExplainer-Dense | 26.95 | 24.02 | 27.18 | 30.41 | 28.53 | 30.41 | 31.61 | 37.76 | 40.92 |
| LRI-Bernoulli | 12.09 | 15.77 | 19.37 | 21.77 | 24.55 | 26.58 | 30.48 | 32.21 | 34.08 |
| **EDMA** | **30.41** | **32.28** | **36.94** | **40.09** | **42.34** | **46.40** | **49.92** | **52.48** | **56.38** |

results with high fidelity is more likely to accurately identify these functional groups. We provide chemical explanations derived from domain knowledge and elaborate on the contributions of functional groups. For each molecule, we match the number of atoms identified by the explanation methods to those in the actual chemical explanations across all methods for a fair comparison. Since functional groups are not necessarily connected, we do not impose a requirement for the explanatory subgraphs to be connected. In Fig. 4, we visualize the identified substructures from various explanation methods on several real molecules from the QM9 dataset, focusing on the property $\mu$ with DimeNet++ as the backbone. Clearly, EDMA is effective in identifying explanatory substructures that align closely with established chemical knowledge, providing more meaningful and interpretable explanations. In contrast, other baseline methods rely on softer masks, which often lead to less precise and less reliable explanations. By adopting approximately discrete masks, EDMA ensures a higher fidelity to chemical understanding, resulting in more scientifically coherent and robust explanations.

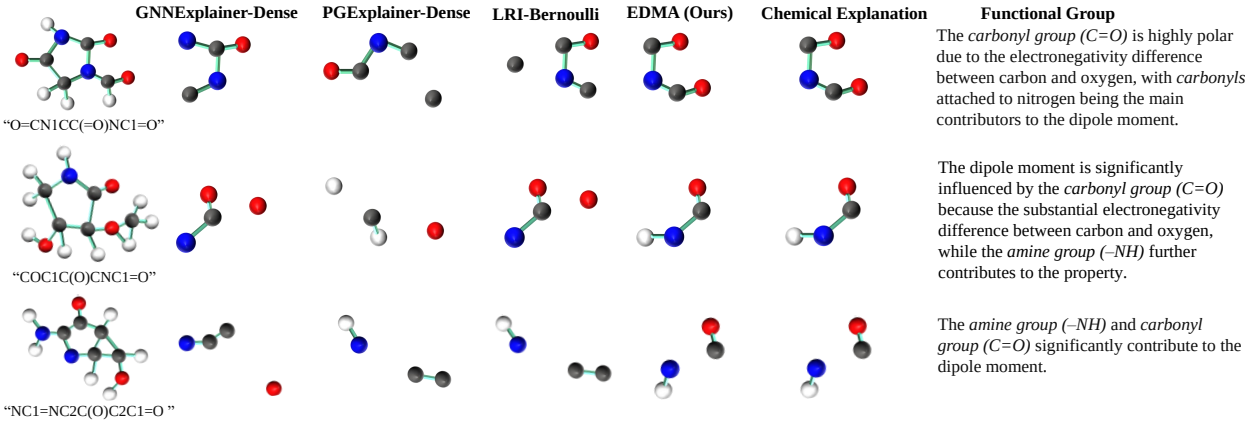


Figure 4: The first column showcases molecules from the QM9 dataset, along with corresponding SMILES strings. The following columns present the explanation results from various baseline methods alongside our EDMA method. Finally, the last two columns offer insights into the chemical explanations and the effects of functional groups. **It is evident that our method, EDMA, delivers the most accurate explanation results, aligning closely with chemical priors.**

### 4.4 Ablation Study

We conduct ablation studies to investigate three fundamental questions: (1) Is the discreteness of masks the primary factor driving improvements in explanation results for 3D graphs, or does the formulation of EBM also contribute? (2) Is there a clear need for methods that *achieve discreteness by design*, such as our proposed EDMA, over regularization-based approaches? (3) Is the outperformance of our proposed method consistent when using equivariant models as backbones?

To address the first question, we isolate the EBM formulation by reducing the effect of discrete masks in our proposed EDMA pipeline. Specifically, we keep all components unchanged except for the parameters related to energy adjustments. In particular, we use a larger $T$, which diminishes the amplification of the energy difference discussed in Sec. 3.3, thereby resulting in relatively less "discrete masks". We refer to this variant as EDMA-soft. The results are presented in Table 6. Clearly, without adjusting the temperature to enforce discreteness, the explanation fidelity of EDMA-soft is much worse than that of EDMA, suggesting that the discreteness of masks is the primary factor in improving explanation fidelity.

Table 6: Experimental results comparing explanation Fidelity$^-$ of the EDMA method and its soft mask variant, EDMA-soft (achieved by adjusting the temperature hyper-parameter), are presented for the property $\mu$ (dipole moment) using SchNet as the backbone model.

| **Top-$k$** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|
| **EDMA** | 2.74 | 3.73 | 4.31 | 4.83 | 5.08 | 5.47 | 5.72 | 5.31 |
| **EDMA-soft** | 3.32 | 4.52 | 6.14 | 7.20 | 7.88 | 8.09 | 8.02 | 7.44 |

To address the second question, we remove the regularization term (cross-entropy loss) in Eq. (8) from the baseline models. If the inclusion of such a regularization term is effective, its removal should result in a significant performance downgrade. However, as presented in Table 7, excluding the regularization term from the optimization objective does not lead to any noticeable decline in explanation fidelity; in fact, in some cases, it even outperforms the version with regularization. This is caused by the complexities in the optimization process, as discussed in Sec. 3.4. On the other hand, our proposed EDMA method consistently outperforms the baselines, as it achieves discreteness by design rather than by introducing regularization terms that complicate the optimization objective.

Table 7: Explanation Fidelity$^-$ for all baseline methods with and without (marked with $^*$) entropy loss and our proposed EDMA method using SchNet as the backbone on the property $\mu$ (dipole moment).

| **Top-$k$** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|
| GNNExplainer-Dense | 3.88 | 5.62 | 7.28 | 8.05 | 8.27 | 8.00 | 7.59 | 6.87 |
| GNNExplainer-Dense$^*$ | 3.82 | 5.90 | 7.40 | 8.11 | 8.09 | 7.83 | 7.44 | 6.75 |
| PGExplainer-Dense | 2.91 | 3.73 | 4.83 | 6.09 | 6.62 | 6.55 | 6.81 | 6.08 |
| PGExplainer-Dense$^*$ | 2.84 | **3.61** | 4.86 | 6.16 | 6.73 | 7.05 | 6.74 | 6.22 |
| LRI-Bernoulli | 3.50 | 4.84 | 6.16 | 6.88 | 7.10 | 7.29 | 7.43 | 7.32 |
| LRI-Bernoulli$^*$ | 3.70 | 5.28 | 6.46 | 6.95 | 7.34 | 7.40 | 7.49 | 7.22 |
| **EDMA** | **2.74** | 3.73 | **4.31** | **4.83** | **5.08** | **5.47** | **5.72** | **5.31** |

To address the third question, we compare the performance of EDMA with GNNExplainer using an equivariant architecture, EGNN (Satorras et al., 2021), on property $G_f$ of the QM9 dataset. The results are presented in Table 8. Our method consistently outperforms GNNExplainer, demonstrating its effectiveness with equivariant models.

Table 8: Explanation Fidelity$^-$ (the lower the better) for GNNExplainer and our proposed EDMA method using EGNN as the backbone on the property $G_f$ (atomization free energy). The better results are highlighted in bold.

| **Top-$k$** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|
| GNNExplainer-Dense | 52.18 | 72.29 | 92.91 | 114.02 | 135.42 | 157.03 | 178.78 | 200.55 |
| **EDMA** | **46.45** | **65.15** | **84.79** | **104.98** | **125.56** | **146.62** | **166.85** | **187.44** |

## 5 Conclusions and Future work

In conclusion, we introduce a principled approach specifically designed to explain 3D GNNs. By recognizing the varying assumptions in 3D GNNs, we identify a common bottleneck in all existing methods for 3D explanations: the gap between the optimized soft masks and the final explanatory substructures due to the non-discreteness of masks. To address this, we propose a novel energy-based explanation function that generates, by pushing up and down the energies, masks that are approximately discrete and highly confident, leading to improved explanation fidelity. Experimental results on both small and large molecules demonstrate the effectiveness and generalizability of our method.

**Future Work.** Our method represents an early effort in 3D molecular graph explanation. It would be interesting to explore whether incorporating additional chemical domain knowledge could further enhance explanation fidelity and generalizability. Moreover, it is worth investigating applying our findings to more complex molecular tasks, such as utilizing the property-informed subgraphs extracted by EDMA for drug editing and molecular optimization tasks, to enable more efficient drug discovery.

## References

Brandon Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *Proceedings of the 33st International Conference on Neural Information Processing Systems*, pp. 14537–14546, 2019.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):1–11, 2022.

David Belanger and Andrew McCallum. Structured prediction energy networks. In *International Conference on Machine Learning*, pp. 983–992. PMLR, 2016.

Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *Advances in neural information processing systems*, 30, 2017.

Enyan Dai and Suhang Wang. Towards self-explainable graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge ManageOn explainability of graph neural networks via subgraph explorationsment*, CIKM '21, pp. 302–311, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482306. URL `https://doi.org/10.1145/3459637.3482306`.

Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7183–7192, 2020.

P ERDdS and A R&wi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. When comparing to ground truth is wrong: On evaluating gnn explanation methods. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 332–341, 2021.

Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6533–6542, 2017.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

Johannes Gasteiger, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020a.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020b.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Rohan Gupta, Devesh Srivastava, Mehar Sahu, Swati Tiwari, Rashmi K Ambasta, and Pravir Kumar. Artificial intelligence to deep learning: machine intelligence approach for drug discovery. *Molecular diversity*, 25:1315–1360, 2021.

Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6968–6972, 2023. doi: 10.1109/TKDE.2022.3187455.

Arian R Jamasb, Alex Morehead, Chaitanya K Joshi, Zuobai Zhang, Kieran Didi, Simon Mathis, Charles Harris, Jian Tang, Jianlin Cheng, Pietro Liò, et al. Evaluating representation learning on the protein structure universe. *ArXiv*, pp. arXiv–2406, 2024.

Andrew R Leach. *Molecular modelling: principles and applications*. Pearson education, 2001.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Shuang Li, Yilun Du, Gido Van de Ven, and Igor Mordatch. Energy-based models for continual learning. In *Conference on lifelong learning agents*, pp. 1–22. PMLR, 2022.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3D molecular graphs. In *International Conference on Learning Representations*, 2022.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through $l_0$ regularization. *arXiv preprint arXiv:1712.01312*, 2017.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33: 19620–19631, 2020.

Siqi Miao, Yunan Luo, Mia Liu, and Pan Li. Interpretable geometric deep learning via learnable randomness injection. In *NeurIPS 2022 AI for Science: Progress and Promises*, 2022.

Bo Pang and Ying Nian Wu. Latent space energy-based model of symbol-vector coupling for text generation and classification. In *International Conference on Machine Learning*, pp. 8359–8370. PMLR, 2021.

Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10764–10773, 2019. doi: 10.1109/CVPR.2019.01103.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Marc'Aurelio Ranzato, Y-Lan Boureau, Sumit Chopra, and Yann LeCun. A unified energy-based framework for unsupervised learning. In *Artificial Intelligence and Statistics*, pp. 371–379. PMLR, 2007.

Amirmohammad Rooshenas, Dongxu Zhang, Gopal Sharma, and Andrew McCallum. Search-guided, lightly-supervised training of structured prediction energy networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting graph neural networks for {nlp} with differentiable edge masking. In *International Conference on Learning Representations*, 2021.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, pp. 991–1001, 2017.

Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. ComENet: Towards complete and efficient message passing for 3D molecular graphs. In *The 36th Annual Conference on Neural Information Processing Systems*, pp. 650–664, 2022.

Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. Learning hierarchical protein representations via complete 3D graph networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=9X-hgLDLYkQ`.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International conference on machine learning*, pp. 2635–2644. PMLR, 2016.

Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery ; Data Mining*, KDD '20. ACM, August 2020. doi: 10.1145/3394486.3403085.

Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, Keir Adams, Maurice Weiler, Xiner Li, Tianfan Fu, Yucheng Wang, Haiyang Yu, YuQing Xie, Xiang Fu, Alex Strasser, Shenglong Xu, Yi Liu, Yuanqi Du, Alexandra Saxton, Hongyi Ling, Hannah Lawrence, Hannes Stärk, Shurui Gui, Carl Edwards, Nicholas Gao, Adriana Ladera, Tailin Wu, Elyssa F. Hofgard, Aria Mansouri Tehrani, Rui Wang, Ameya Daigavane, Montgomery Bohde, Jerry Kurtin, Qian Huang, Tuong Phung, Minkai Xu, Chaitanya K. Joshi, Simon V. Mathis, Kamyar Azizzadenesheli, Ada Fang, Alán Aspuru-Guzik, Erik Bekkers, Michael Bronstein, Marinka Zitnik, Anima Anandkumar, Stefano Ermon, Pietro Liò, Rose Yu, Stephan Günnemann, Jure Leskovec, Heng Ji, Jimeng Sun, Regina Barzilay, Tommi Jaakkola, Connor W. Coley, Xiaoning Qian, Xiaofeng Qian, Tess Smidt, and Shuiwang Ji. Artificial intelligence for science in quantum, atomistic, and continuum systems. *Foundations and Trends® in Machine Learning*, 18(4):385–912, 2025. ISSN 1935-8237. doi: 10.1561/2200000115. URL `http://dx.doi.org/10.1561/2200000115`.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

## A Experimental Details

To ensure a fair comparison between our proposed model and existing methods, we performed extensive hyperparameter tuning for both our approach and the baseline methods. For the baseline methods, we employed the grid search to systematically explore their respective hyperparameter spaces. In the case of the QM9 dataset, each model's performance was evaluated based on the Mean Absolute Error (MAE) on the test set, with the objective of identifying the optimal parameter configurations.

**SchNet**. For property $\mu$: we tested GNNExplainer-Dense using the following parameter settings: the coefficient for size loss varied from 1.0 to 5.5 with a step size of 2.0, the coefficient for entropy loss ranged from 0.1 to 1.1 with a step size of 0.4, and the number of training epochs ranged from 50 to 500 with a step size of 200. The optimal parameters were determined as follows: a size loss coefficient of 1.0, an entropy loss coefficient of 0.9, and 50 training epochs. Similarly, for PGExplainer-Dense, the training parameters were set with a training epoch of 40, a size loss coefficient of 30.0, and an entropy loss coefficient of 1.6. For the LRI-Bernoulli method, the training epoch was set to 50, with a prediction loss coefficient of 5.0 and an information loss coefficient of 1.0. The EDMA model's training epoch was established at 300, with the parameter $\alpha$ set to 1.0.

For the property $G_f$, the following parameters were established for GNNExplainer-Dense: the coefficient for size loss was set to 300.0, the coefficient of entropy loss was also set to 300.0, and the number of training epochs was fixed at 300. In the case of PGExplainer-Dense, we set the training epoch to 100, with a coefficient of size loss of 520.0 and entropy loss coefficient of 300.0. For LRI-Bernoulli, the training epoch was established at 300, with a prediction loss coefficient of 1.0 and an information loss coefficient of 3.0. The training epoch for EDMA was similarly set to 300, with the parameter $\alpha$ assigned to a value of 500.0. Additionally, due to the presence of a shortcut embedding preceding the final readout layer in the PyG implementation for SchNet on property $G_f$, the node mask was multiplied by this embedding layer to ensure the validity of the experimental setup.

**DimeNet++**. For the property $\mu$ on DimeNet++, we established the following for GNNExplainer-Dense: the coefficient for feature size loss was set to 1.5, the entropy loss coefficient was set to 0.5, and the number of training epochs was fixed at 200. In the case of PGExplainer-Dense, the training epoch was set to 150, with a size loss coefficient of 0.5 and an entropy loss coefficient of 2.5. For LRI-Bernoulli, the training epoch was set to 500, with a prediction loss coefficient of 1.0 and an information loss coefficient of 0.5. The training epoch for EDMA was similarly established at 300, with the parameter $\alpha$ assigned a value of 3.0.

For the property $G_f$, we established the following parameters for GNNExplainer-Dense: the size loss coefficient was set to 0.5, the entropy loss coefficient was set to 5.0, and the number of training epochs was fixed at 300. For PGExplainer-Dense, we set the training epoch to 100, with both the size and entropy loss coefficient set to 5.0. In the case of LRI-Bernoulli, the training epoch was set to 500, with a prediction loss coefficient to 1.0 and an information loss coefficient of 5.0. The training epoch for EDMA was similarly established at 500, with the parameter $\alpha$ assigned a value of 8.0. It is important to note that while we use the same notation for $G_f$, in the PyG package for DimeNet++ this property specifically refers to the atomization free energy.

**ProNet**. For the EC dataset, we established the following parameters for GNNExplainer-Dense: the size loss coefficient was set to 10.0, the entropy loss coefficient was set to 8.0, and the number of training epochs was fixed at 500. For PGExplainer-Dense, we set the training epoch to 20, with the size loss coefficient set to 0.05 and entropy loss coefficient set to 1.0. In the case of LRI-Bernoulli, the training epoch was set to 20, with both the prediction and information loss coefficient to 1.0. The training epoch for EDMA was similarly established at 500, with the parameter $\alpha$ assigned a value of 8.0.

## B Descriptions of Baseline Explanation Methods

Graph explanation methods seek to identify a subgraph that maximally retains the predictive signal of the original graph $G$ under a budget constraint $B$ on the size of the subgraph (Ying et al., 2019). The subgraph is identified through masking out unwanted edges in the original graph, and to allow gradient-

based optimization, existing works relax the discrete (hard) masks with discrete values 0 and 1 to soft masks with values between 0 and 1. Mathematically, this is expressed as:

$$G_S^* = \arg\min_{\mathbf{M}^{'}} \ \mathcal{L}(Y; \Phi(\mathbf{X}, \mathbf{M}^{'} \odot \mathbf{A}))$$
$$\text{s.t.} \quad \mathbf{M}^{'} \in [0,1]^{n \times n}, \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{M}^{'}{}_{ij} \leq B.$$

Practically, all the baseline explanation methods are based on this expression; the only difference is in how they realize the masks $\mathbf{M}^{'}$.

GNNExplainer (Ying et al., 2019) directly uses a set of learnable parameters followed by a sigmoid function to represent these masks. Mathematically, $\mathbf{M}^{'} = \sigma(\mathbf{\Theta})$, where $\sigma$ is the sigmoid function to constrain the values to be between 0 and 1 and $\mathbf{\Theta} \in \mathbb{R}^{n \times n}$ is the set of learnable parameters. It is worth noting that GNNExplainer can explain only one graph per training (the transductive setting).

PGExplainer (Luo et al., 2020) extends GNNExplainer to the inductive setting by using an MLP to generate edge masks from learned feature embeddings extracted from the final message passing layer. Once trained, PGExplainer can explain any graph from the same distribution (the inductive setting).

Learnable Randomness Injection (LRI) (Miao et al., 2022) introduces two distinct approaches: LRI-Bernoulli and LRI-Gaussian for explaining 3D geometric graphs. LRI-Bernoulli applies Bernoulli noise to nodes, that is, applying binary masks to nodes to assess the impact of node existence on the final prediction. Similar to PGExplainer, these masks can be parametrized by MLPs. In contrast, LRI-Gaussian perturbs node positions by adding Gaussian noise, evaluating the significance of geometric features rather than identifying substructures. LRI-Gaussian is not a subgraph extraction method and is out of the scope of this work.

## C Invariant and Equivariant GNNs

When modeling molecular representations, two main approaches can be used: invariant GNN models and equivariant models. The key difference between these two categories lies in how they handle geometry transformations, such as rotation and translation. Invariant models aim to capture atomic representations that are invariant under these geometric transformations. Typically, these invariant representations are achieved by incorporating relative 3D information as scalar inputs, such as Euclidean distances between atom pairs, bond angles, and torsion angles between planes. During the message passing process, these learned invariant representations are used to exchange messages between atoms. In this manner, the final output remains invariant under geometric transformations. SchNet and DimeNet++ are classical examples of invariant models: SchNet (Schütt et al., 2017) incorporates pairwise distances in its representations, while DimeNet++ (Gasteiger et al., 2020b;a) includes additional bond angles to improve expressiveness.

In contrast, equivariant models seek to represent features that are equivariant to the geometry transformations of the input. When the input graph undergoes geometric transformations, the learned representations transform in a predictable manner. High-order tensors, such as spherical harmonics, are often used to represent these features to ensure equivariance. In the message passing process, tensor products are commonly applied for tensor operations, while Clebsch-Gordan coefficients and Wigner-D matrices are frequently used to enforce equivariance. In our study, the classical equivariant model EGNN (Satorras et al., 2021) serves as a baseline for comparing the performance of our method and GNNExplainer. Extensive experimental results demonstrate that, whether using invariant or equivariant models, our model achieves competitive performance.