MEGAN: Multi Explanation Graph Attention Network

Anonymous Author(s)
Affiliation
Address
email

Abstract

Besides increasing trust in the human-AI relationship, XAI methods have the potential to promote new scientific insight. Graph neural networks (GNNs) have recently established themselves as a valuable tool in chemistry and material sciences. Various XAI methods have already been applied to gain new understanding of real-world scientific questions in these application domains. To that end we propose MEGAN, a multi-explanation graph attention network. Unlike common post-hoc XAI methods, our model is self-explaining and features multiple explanation channels, which can be chosen independent of the task specifications. We first validate our model on a synthetic graph regression dataset. We then apply our model to the prediction of water solubility for chemical compounds. We find that our model learns to produce explanations consistent with human intuition, opening the way to learning from our model in less well-understood tasks.

1 Introduction

Explainable AI (XAI) methods aim to provide additional explanations for a model's predictions to make it's complex inner workings more transparent to humans with the intention to improve trust within the human-AI relationship, provide tools for model analysis, and comply with anti-discrimination laws (Doshi-Velez and Kim, 2017). One important subclass of XAI methods are attention-based models, which usually derive an attributional explanation from the internal attention weights. In contrast to most post-hoc explainability procedures, the differentiable nature of attention models offers the potential for explanation supervised training. Here the explanations generated by the model are trained to match a given reference alongside the main prediction value. Recently several successes for explanation supervision have been reported in the domains of natural language processing (Fernandes et al., 2022; Stacey et al., 2022) and image processing (Linsley et al., 2019; Qiao et al., 2018).

Attention is increasingly applied to graph neural networks as well (Veličković et al., 2018; Brody et al., 2022; Pan et al., 2022). However, its potential to be used in explainable models is not fully exploited yet, when compared to other state-of-the-art graph XAI methods such as GNNExplainer (Ying et al., 2019) or Grad-CAM (Pope et al., 2019). We hypothesize that this is partly due to the limited expressiveness of attention-derived explanations. While Grad-CAM for example can generate attributional explanations which feature positive and negative values for each output of a model separately, attention-based models usually can only generate a single explanation with values in the $[0, 1]$ range. Considering a typical node attribution explanation for a regression model, which assigns each node an importance value between 0 and 1: It highlights which nodes were important for the model to make a given prediction, but it does not explain how the node correlates with the actual numerical prediction. We show in this paper that a substantial gain in interpretability can be achieved when using attention to generate multi-channel explanations.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

We propose the *multi-explanation graph attention network* (MEGAN) architecture, which allows the number of explanation channels to be chosen as an independent parameter. The network generates multiple channels of node and edge attribution explanations, building on the recently proposed GATv2 layer (Brody et al., 2022) for edge explanations and a weighted global pooling operation for the node explanations. While the number of explanation channels is a free parameter, in this work we first investigate the special choice of two explanation channels for regression tasks. This amounts to a linear interpretation of the problem, where one channel is providing positive explanations and the other negative explanations with respect to a chosen reference value. For the classification case, we choose the number of explanation channels the same as the number of classes. To promote the channels to behave according to these intended interpretations, we introduce a modified training procedure where each training step is split into an explanation-only step first and a prediction step afterward.

We validate the functionality of our model on a synthetic graph regression dataset as well as real-world datasets. We firstly show that the special single-explanation case of our model produces explanations with quality similar to GNNExplainer. Switching to two explanations significantly improves prediction performance and interpretability of explanations. We furthermore show that explanation-supervised training produces explanations with the highest similarity to the ground truth explanations as well as fidelity w.r.t. the intended interpretations of the respective channels. At last, we demonstrate our model for two real-world datasets, namely the water solubility prediction for molecular graphs and sentiment classification for movie reviews. We can show that our model produces explanations consistent with human intuition for these tasks, potentially opening the way to learning from our model in less well-understood real-world tasks related to chemistry and material sciences (see Appendix D).

2 Related Work

Graph explanations. Yuan et al. (2022) provides an overview of XAI methods that were either adopted or specifically designed for graph neural networks (GNNs). Notable ones include GradCAM (Pope et al., 2019), GraphLIME (Huang et al., 2022) and GNNExplainer (Ying et al., 2019). Jiménez-Luna et al. (2020) presents another overview of XAI methods used for the application domain of drug discovery. (Sanchez-Lengeling et al., 2020) evaluate many common graph XAI methods for tasks of chemical property prediction. Henderson et al. (2021) for example introduce regularization terms to improve GradCAM generated explanations for chemical property prediction. Most of the approaches presented here are classified as post-hoc methods, which aim to explain the decision of of existing models in hindsight. Our model on the other hand is *self-explaining*, as Jiménez-Luna et al. (2020) describe it. Due to being based on an attention mechanism, this provides the potential for explanation-supervised training.

Explanation supervision. During explanation-supervised training, the explanations generated by the model are trained to match a given dataset of usually human-generated explanations alongside the main prediction task. Linsley et al. (2019), Qiao et al. (2018) and Boyd et al. (2022) for example have demonstrated promising results for explanation supervision in the image processing domain. Likewise, Fernandes et al. (2022), Pruthi et al. (2022) and Stacey et al. (2022) for example demonstrate this for the language processing domain. Recently, Gao et al. (2021) demonstrated a method to perform GNN explanation supervision for GradCAM explanations. However, their implementation currently does not support batching and the extent of the explanation is coupled to the task specifications. In contrary, our MEGAN architecture supports batching and features a choice of explanation channels independent of task specifications.

3 Multi-Explanation Graph Attention Network

3.1 Task Description

We assume a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is represented by a set of node indices $\mathcal{V} \subset \mathbb{N}^V$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \subset \mathbb{R}^E$, where a tuple $(i, j) \in \mathcal{E}$ denotes an edge from node i to node j. Every node i is associated with a vector of initial node features $\mathbf{h}_i^{(0)} \in \mathbb{R}^{N_0}$, combining into the initial node feature tensor $\mathbf{H}^{(0)} \in \mathbb{R}^{V \times N_0}$. Each edge is associated with a feature vector $\mathbf{u}_i \in \mathbb{R}^M$, combining into the edge feature tensor $\mathbf{U} \in \mathbb{R}^{E \times M}$. We consider the problems of graph classification and graph

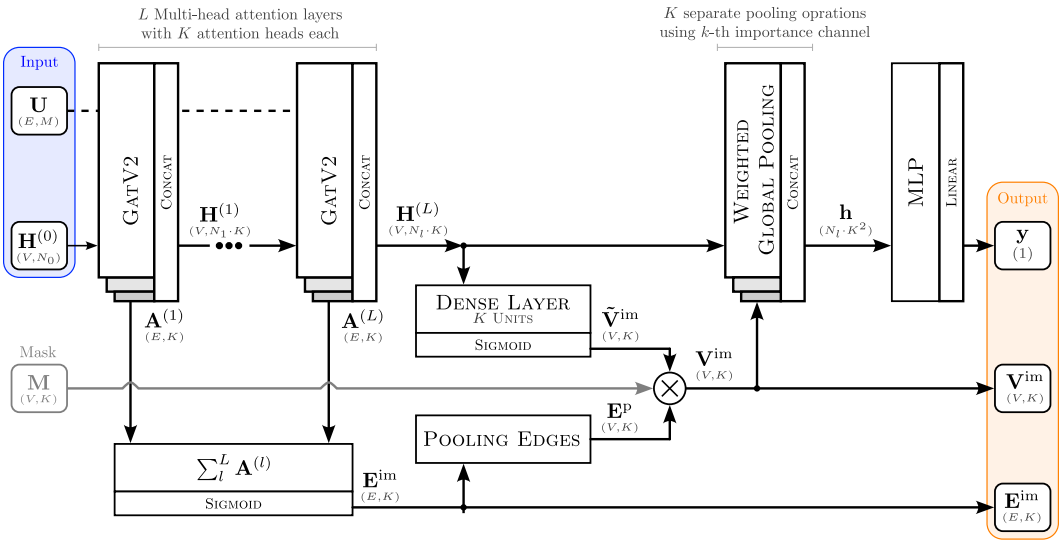

Figure 1: Multi-explanation graph attention network (MEGAN) architecture overview. Rectangle boxes represent layers; arrows indicate layer interconnections. Rounded boxes represent tensors. Intermediate tensors are also named annotated arrows. Tuples beneath variable names indicate the tensor shape, with batch dimension omitted, but implicitly assumed as the first dimension for all.

regression, i.e. output vectors $\mathbf{y}^{\text{class}} \in \{0,1\}^C$ (with $|\mathbf{y}^{\text{class}}| = 1$, such that the highest value of that vector identifies the class $\hat{c} =_c \mathbf{y}_c^{\text{class}}$) or single real output values $y^{\text{reg}} \in \mathbb{R}$, respectively.
In addition, MEGAN outputs node and edge attribution explanations alongside each prediction. We define explanations as priority masks by assigning $[0,1]$ values to each node and each edge, representing the importance of the corresponding element towards the outcome of the prediction. We generally assume that any prediction may be explained by K individual importance channels. The node explanations are given as the *node importance* tensor $\mathbf{V}^{\text{im}} \in [0,1]^{V \times K}$ and the edge explanations are given as the *edge importance* tensor $\mathbf{E}^{\text{im}} \in [0,1]^{E \times K}$.

3.2 Architecture Overview

To solve the previously defined task we propose the following *multi-explanation graph attention network* (MEGAN) architecture. Figure 1 provides a visual overview of this architecture. The network consists of L attention layers, where the number of layers L and the hidden units of each layer are hyperparameters. Each of these layers consists of K individual, yet structurally identical GATv2 (Brody et al., 2022) attention heads, one for each of the K expected explanation channels. Assuming the attention heads in the l-th layer have N_l hidden units, then each attention head produces it's own node embeddings $\mathbf{H}^{(l,k)}$, where $k \in \{1, \ldots, K\}$ is the head index. The final node embeddings $\mathbf{H}^{(l)} \in \mathbb{R}^{V \times N_l \cdot K}$ of layer l are then produced by concatenating all these individual matrices along the feature dimension:

$$\mathbf{H}^{(l)} = \mathbf{H}^{(l,1)} \,||\, \mathbf{H}^{(l,2)} \,||\, \ldots \,||\, \mathbf{H}^{(l,K)} \tag{1}$$

This node embedding tensor will then be used as the input to *each* of the K attention heads of layer $l+1$. Aside from the node embeddings, each attention head also produces a vector $\mathbf{A}^{(l,k)} \in \mathbb{R}^E$ of attention logits which are used to calculate the attention weights

$$\boldsymbol{\alpha}^{(l,k)} = (\mathbf{A}^{(l,k)}) \tag{2}$$

of the k-the attention head in the l-th layer. The edge importance tensor $\mathbf{E}^{\text{im}} \in [0,1]^{E \times K}$ is calculated from the concatenation of these attention logit tensors in the feature dimension and summed up over the number of layers:

$$\mathbf{E}^{\text{im}} = \left(\sum_{l=1}^{L} \left(\mathbf{A}^{(l,1)} \,||\, \mathbf{A}^{(l,2)} \,||\, \ldots \,||\, \mathbf{A}^{(l,K)} \right) \right) \tag{3}$$

Based on this, a local pooling operation is used to derive the pooled edge importance tensor $\mathbf{E}^{\text{p}} \in [0,1]^{V \times K}$ for the *nodes* of the graph.
The final node embeddings $\mathbf{H}^{(L)}$ are then used as the input to a dense network, whose final layer is set to have K hidden units, producing the node importance embeddings $\tilde{\mathbf{V}}^{\text{im}} \in [0,1]^{V \times K}$. The node importance tensor is then calculated as the product of those node importance embeddings $\tilde{\mathbf{V}}^{\text{im}} \in [0,1]^{V \times K}$ and the pooled edge importance tensor $\mathbf{E}^{\text{p}} \in [0,1]^{V \times K}$:

$$\mathbf{V}^{\text{im}} = \tilde{\mathbf{V}}^{\text{im}} \cdot \mathbf{E}^{\text{p}} \cdot \mathbf{M}. \tag{4}$$

The mask $\mathbf{M}$ introduced in Fig. 1 is needed to compute the fidelity metric, which will be discussed later.

At this point the edge and node importance matrices, which represent the explanations generated by the network, are already accounted for, which leaves only the primary prediction to be explained. The first remaining step is a global sum pooling operation which turns the node embedding tensor $\mathbf{H}^{(L)}$ into vector of global graph embeddings. For this, K separate weighted global sum pooling operations are performed, one for each explanation channel. Each of these pooling operations uses the same node embeddings $\mathbf{H}^{(L)}$ as input, but a different slice $V^{\text{im}}_{:,k}$ of the node importance matrix as weights. In that way, K separate graph embedding vectors

$$\mathbf{h}^{(k)} = \sum_{i=0}^{V} \left(\mathbf{H}^{(L)} \cdot \mathbf{V}^{\text{im}}_{:,k} \right)_{i,:} \tag{5}$$

are created, which are then concatenated into a single graph embedding vector

$$\mathbf{h} = \mathbf{h}^{(1)} \,||\, \mathbf{h}^{(2)} \,||\, \ldots \,||\, \mathbf{h}^{(K)} \tag{6}$$

where $\mathbf{h} \in \mathbb{R}^{N_L \cdot K^2}$. This graph embedding vector is then passed through a generic MLP whose final layer either has linear activation for graph regression or softmax activation for graph classification to create an appropriate output

$$\mathbf{y} = \text{MLP}(\mathbf{h}) \tag{7}$$

3.3 Explanation Co-Training

With the architecture as is it is up to this point, there is no mechanism to ensure that individual explanation channels learn the appropriate explanations according to their intended interpretation. We use a modified training procedure to ensure that each channel learns the appropriate explanations. This is illustrated in Figure 2. Each overall train step of the model is split into two parts: The explanation step and the prediction step. The explanation step is based only on the node importances produced by the network. A global sum pooling operation is used to turn the importance values of each separate channel into a single *alternate output tensor* $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times K}$, where B is the training batch size. This alternate output tensor is then used to solve an approximation of the original prediction problem. This can be seen as a reduction of the problem into a set of K separate subgraph counting problems, where each of those only uses the subset of training batch samples that aligns with the respective channel's intended interpretation.

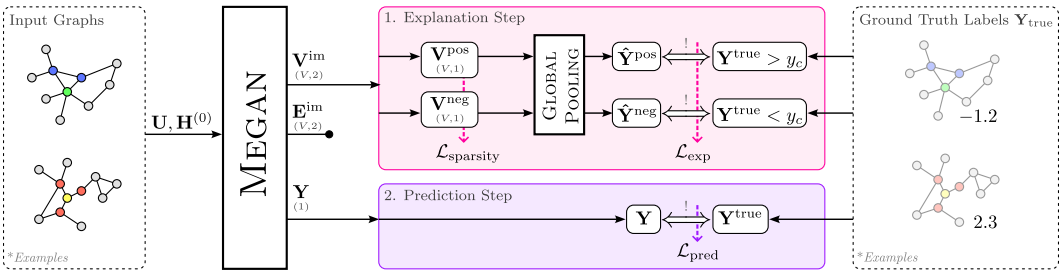

Figure 2: Illustration of split training procedure for the regression case. The explanation-only train step attempts to find an approximate solution to the main prediction task, by using only a globally pooled node importance tensor. After the weight update for the explanation step was applied to the model, the prediction step performs another weight update based on the actual output of the model and the ground truth labels.

Regression For regression, we assume $K = 2$, where the first channel represents the negative and the second channel the positive influences relative to the reference value y_c. We select all samples of the current training batch lesser and greater than the reference value and use these to calculate a mean squared error (MSE) loss:

$$\mathcal{L}_{\text{exp}} = \frac{1}{2 \cdot B} \sum_{b=1}^{B} \begin{cases} (\hat{\mathbf{Y}}_{b,0} - y_c - \mathbf{Y}_b^{\text{true}})^2 & \text{if } \mathbf{Y}_b^{\text{true}} < y_c \\ (\hat{\mathbf{Y}}_{b,1} - y_c - \mathbf{Y}_b^{\text{true}})^2 & \text{if } \mathbf{Y}_b^{\text{true}} > y_c \end{cases} \tag{8}$$

Classification We assume the number of channels $K = C$ is equal to the number of possible output classes C. We use the alternate output channel to compute an individual binary cross entropy (BCE) loss for each channel:

$$\mathcal{L}_{\text{exp}} = \mathcal{L}_{\text{BCE}}(\mathbf{Y}^{\text{class}}, \sigma(\hat{\mathbf{Y}})) \tag{9}$$

For regression as well as classification, the total loss during the explanation step consists of these task-specific terms and an additional term for explanation sparsity:

$$\mathcal{L}_{\text{total}} = \gamma \mathcal{L}_{\text{exp}} + \beta \mathcal{L}_{\text{sparsity}} \tag{10}$$

where γ and β are hyperparameters of the training process. Explanation sparsity $\mathcal{L}_{\text{sparsity}}$ is calculated as L1 regularization over the node importance tensor. Based on this loss the gradients are calculated and the model weights are updated.

After the explanation step, a normal prediction step is executed as well, which uses the main model output and the entirety of the training batch to perform another weight update.

3.4 Implementation

The model is implemented using KGCNN framework (Reiser et al., 2021) which implements graph neural networks in TensorFlow and Keras. The source code is publicly available and can be found at `https://github.com/awa59kst120df/graph_attention_student`.

4 Computational Experiments

In this section we only give a brief overview of the used datasets and experiments. For more detailed information, we refer to Appendix A and Appendix B, respectively.

4.1 Datasets

RbMotifs. We create a synthetic graph regression dataset consisting of 5000 randomly generated graphs, where each node has three node features representing a RGB color value. Edges are undirected and unweighted. Some of these randomly generated colored graphs are additionally seeded with specific subgraph motifs, which consist of pre-defined color combinations and are associated with a constant value. When a motif appears in a graph, the associated value is added to that graph's overall value.

Solubility. Approx. 8000 molecular graphs of the AqSolDB dataset (Sorkun et al., 2019). The target value for each graph is the measured logS value representing water solubility of the corresponding chemical compound. Node and edge features are generated by RDKit (Landrum, 2010).

MovieReviews. A dataset of 2000 movie reviews, classified by sentiments "positive" and "negative". Originally a text-based dataset from the ERASER benchmark (DeYoung et al., 2020), we pre-processed it into a graph dataset by representing each word as a node and connecting adjacent nodes by undirected and unweighted edges using a sliding window of size 2 as it was done by Rathee et al. (2022). Node features for each node were generated by using a pre-trained GLOVE (Pennington et al., 2014) model with 50 dimensions.

4.2 Metrics

We measure similarity to the ground truth explanations using AUROC, as it is done by McCloskey et al. (2019). We measure sparsity on binarized explanations, which are generated by thresholding

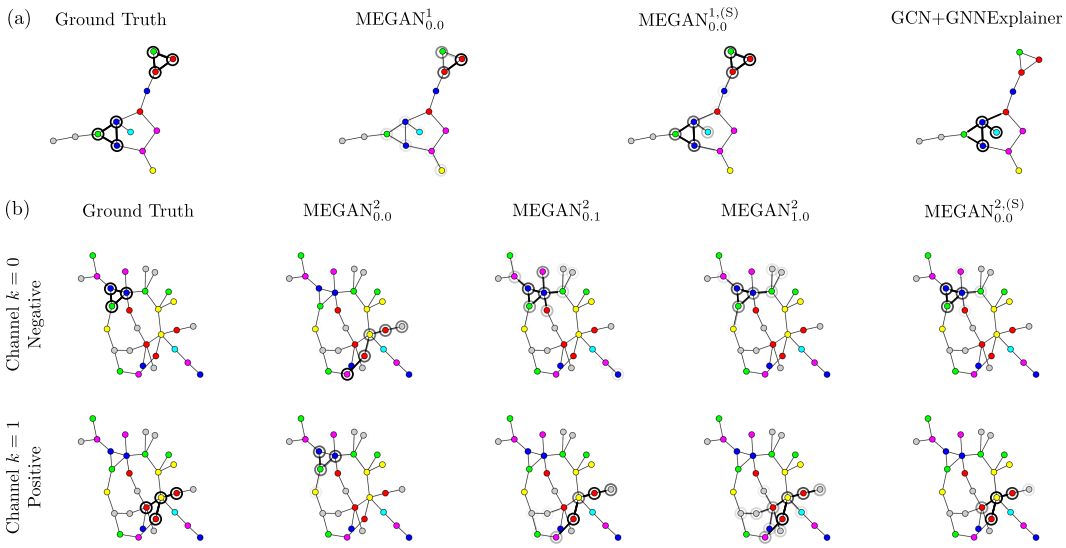

Figure 3: Examples for experiments with synthetic dataset RbMotifs and MEGAN$^{K}_{\gamma}$. (a) Single-explanation case, where GNNExplainer and MEGAN reproduce ground truth explanation consisting of a single motif, while explanation-supervised version finds both motifs (b) Dual-explanation case, where baseline MEGAN model finds the correct motifs, but assigns them to the wrong channels w.r.t to the intended channel interpretations. Models with $\gamma \neq 0$ and the explanation-supervised version find the motifs and assign them to the correct channels.

the model's continuous attributions. We also measure explanation fidelity as described in Yuan et al. (2022) and additionally provide the fidelity of random explanations for comparison. More details can be found in Appendix B.2.

For the multi-explanation models, we define the Fidelity* metric: Considering regression as an example, we argue that the positive explanation channel is faithful to the predicted output, exactly if the model output becomes significantly more *negative* when the positive channel is withheld from the model, as it is then missing all supposedly positive information about the graph. The same applies for the negative channel, which when withheld, should produce a more positive output. In this sense we calculate a deviation Δy^k for each channel by supplying a corresponding binary importance mask $\mathbf{M}^k$ (see Figure 1) which completely blocks channel k. A channel's deviation then contributes positively to the overall value if that deviation is along the expected direction as defined before:

$$\text{Fidelity*} = \frac{1}{K} \sum_{k=1}^{K} \begin{cases} +\Delta y^k & \text{if direction of deviation as expected for channel } k \\ -\Delta y^k & \text{if direction of deviation not as expected for channel } k \end{cases} \tag{11}$$

Consequently, positive values of Fidelity* show good alignment of explanations with their respective channel's intended interpretation, while low and negative values indicate misalignment.

4.3 Synthetic Dataset - Single Explanations

We demonstrate the base capabilities of our model for the special single-explanation ($K = 1$) case. The experiment is conducted on the synthetic RbMotifs graph regression dataset. We compare explanations of a baseline MEGAN$^{K=1}_{\gamma=0.0}$ model, an explanation-supervised MEGAN$^{K=1,(S)}_{\gamma=0.0}$ model and the explanations of GNNExplainer based on a simple GCN network with similar layer structure. The experiment is repeated independently 50 times, where a new random dataset split is chosen each time. We report our results in Table 1 and show examples in Figure 3.

The results show that both MEGAN models greatly outperform the GCN baseline in terms of the main predictions (MSE, R^2). The baseline MEGAN$^{1}_{0.0}$ and GNNExplainer perform comparably in terms of similarity to ground truth, while the explanation-supervised MEGAN$^{1}_{0.0}$ clearly outperforms both, nearly achieving a perfect match. We also point out, that explanations of MEGAN models show

Table 1: Results for computational experiments with synthetic graph regression dataset. We report the median value for 50 independent experiment repetitions in black, as well as the 75th percentile (upper) and 25th percentile (lower) of the distribution in gray. We underline the best result for each column. The first section of the table shows results for the single-explanation experiments and the second section shows the results for the dual-explanation experiments.
$^{(S)}$ Model trained explanation-supervised with ground truth explanations.
$^{(*)}$ Multi-explanation case measures Fidelity* metric introduces in Section 4.2.

Model	MSE	R^2	Node AUC	Edge AUC	Sparsity	Fidelity	$\text{Fidelity}_{\text{rand}}$
GNNX_{GCN}	$1.11 \mid {}^{1.26}_{0.93}$	$0.62 \mid {}^{0.69}_{0.57}$	$0.76 \mid {}^{0.81}_{0.73}$	$0.76 \mid {}^{0.79}_{0.73}$	$0.14 \mid {}^{0.21}_{0.08}$	$0.14 \mid {}^{0.24}_{0.06}$	$0.33 \mid {}^{0.55}_{0.17}$
$\text{MEGAN}^1_{0.0}$	$0.25 \mid {}^{0.32}_{0.20}$	$0.92 \mid {}^{0.93}_{0.89}$	$0.84 \mid {}^{0.87}_{0.79}$	$0.80 \mid {}^{0.82}_{0.76}$	$0.15 \mid {}^{0.30}_{0.09}$	$1.77 \mid {}^{3.08}_{0.77}$	$0.91 \mid {}^{1.69}_{0.45}$
$\text{MEGAN}^{1,(S)}_{0.0}$	$0.25 \mid {}^{0.33}_{0.20}$	$0.91 \mid {}^{0.93}_{0.89}$	$0.97 \mid {}^{0.97}_{0.97}$	$\underline{0.99} \mid {}^{0.99}_{0.99}$	$0.16 \mid {}^{0.24}_{0.11}$	$1.42 \mid {}^{2.37}_{0.67}$	$0.91 \mid {}^{1.57}_{0.52}$
$\text{MEGAN}^2_{0.0}$	$0.21 \mid {}^{0.25}_{0.18}$	$\underline{0.93} \mid {}^{0.94}_{0.91}$	$0.74 \mid {}^{0.91}_{0.59}$	$0.70 \mid {}^{0.81}_{0.58}$	$0.06 \mid {}^{0.10}_{0.04}$	$1.20^{(*)} \mid {}^{10.6}_{-5.1}$	-
$\text{MEGAN}^2_{0.1}$	$0.23 \mid {}^{0.28}_{0.18}$	$0.92 \mid {}^{0.94}_{0.91}$	$0.93 \mid {}^{0.94}_{0.92}$	$0.88 \mid {}^{0.93}_{0.85}$	$0.12 \mid {}^{0.17}_{0.08}$	$2.63^{(*)} \mid {}^{3.33}_{1.93}$	-
$\text{MEGAN}^2_{1.0}$	$0.25 \mid {}^{0.29}_{0.21}$	$0.92 \mid {}^{0.93}_{0.90}$	$0.94 \mid {}^{0.95}_{0.93}$	$0.92 \mid {}^{0.94}_{0.87}$	$0.10 \mid {}^{0.15}_{0.07}$	$2.64^{(*)} \mid {}^{3.23}_{2.20}$	-
$\text{MEGAN}^{2,(S)}_{0.0}$	$\underline{0.19} \mid {}^{0.23}_{0.16}$	$\underline{0.93} \mid {}^{0.94}_{0.92}$	$\underline{0.98} \mid {}^{0.99}_{0.98}$	$\underline{0.99} \mid {}^{0.99}_{0.98}$	$0.08 \mid {}^{0.12}_{0.05}$	$\underline{2.94}^{(*)} \mid {}^{3.92}_{2.33}$	-

noticeably larger median fidelity than the random baseline, while explanations of GNNExplainer don't, indicating that explanations produced by MEGAN actually faithfully reflect the model's predictive behavior.

Generally, in the single-explanation case, we often observe that that with samples where the graph contains two opposing motifs (red and blue) the explanations of baseline $\text{MEGAN}^1_{0.0}$ and GNNExplainer usually focus only on either of the motifs, as it can be seen in Figure 3(a). Only the explanation-supervised $\text{MEGAN}^{1,(S)}_{0.0}$ correctly highlights both motifs.

4.4 Synthetic Dataset - Multi Explanations

In the second experiment, we demonstrate our model's capability to learn dual explanations ($K = 2$). The experiment is conducted on the synthetic RbMotifs graph regression dataset. We compare multiple architecturally identical MEGAN^K_γ models and vary the value of the explanation training hyperparameter γ, including a baseline model with $\gamma = 0$, which performs no additional explanation training step at all, and an explanation-supervised version $\text{MEGAN}^{2,(S)}_{0.0}$

Each experiment is independently repeated 50 times, where a new random dataset split is chosen each time. We report our results in Table 1 and show an example in Figure 3.

The results show that the baseline $\text{MEGAN}^2_{0.0}$ model achieves the lowest similarity to the ground truth explanations, with $\text{MEGAN}^{2,(S)}_{0.0}$ showing the highest value, achieving a near perfect match for nodes and edges. For the prediction performance on the other hand there is no clear difference between the models, all of them achieving nearly identical R^2 values. The Fidelity* is also the highest for the explanation-supervised model $\text{MEGAN}^{2,(S)}_{0.0}$, while the models with $\gamma \neq 0$ result in similarly high median values, although the distributions show higher deviation for lower values of γ. The baseline model $\text{MEGAN}^2_{0.0}$ shows the lowest median Fidelity* value with the highest fluctuations.

We firstly conclude that explanation-supervision can be applied effectively to train a MEGAN model to emulate pre-defined explanations. Secondly, the proposed modified training procedure also leads to the development of explanations which have high similarity to ground truth and are faithful to the intended channel interpretations when compared to the baseline. In this context it is also important to note that the baseline model $\text{MEGAN}^2_{0.0}$ does not perform significantly differently on the prediction task. In fact, we generally observe that this model learns the correct explanations as well, but without any additional guidance the alignment of these explanations with the two available channels is arbitrary. This often leads the correct explanations to develop in the wrong channels (w.r.t their intended interpretation) as it can be seen in Figure 3.

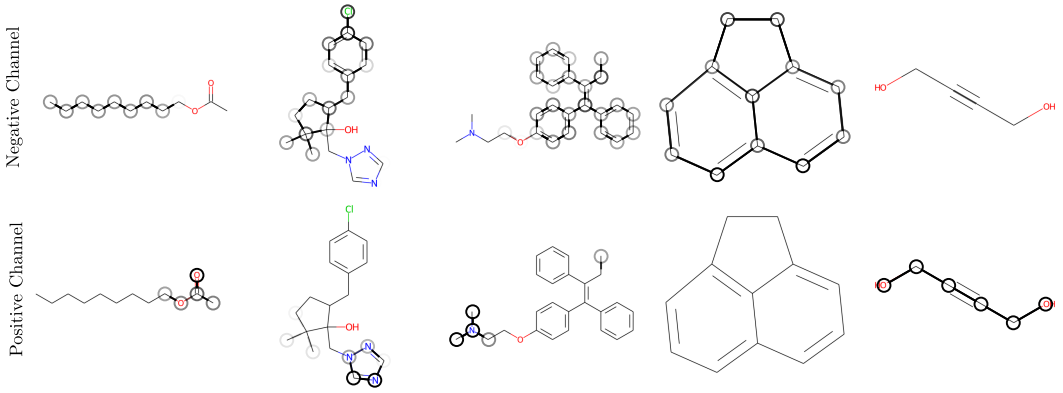

Figure 4: Selected examples for the prediction of water solubility for molecular graphs. Darker highlights for nodes and edges indicate higher predicted importance values. Examples generally show an association of long carbon chains and carbon rings with low water solubility. Groups including nitrogen and oxygen are associated with higher solubility.

4.5 Real-World Datasets

In addition to the synthetic datasets, we conduct experiments with real world datasets, namely the prediction of water solubility for molecular graphs and sentiment classification for movie reviews. For more details on hyperparameter configurations in both cases, we refer to Appendix B.

For the solubility dataset, some illustrated examples can be found in Figure 4 (more in Appendix C.1). The overall predictivity is high (r^2 correlation coefficient of 0.91 on test set). We observe that the negative channel usually highlights long chains of carbon atoms and carbon rings, while the positive channel is usually associated with functional groups including oxygen and nitrogen atoms.
This is consistent with chemistry knowledge, as non-polar carbon chains and rings are associated with low water solubility, while polar chemical groups including O and N atoms increase water solubility. Overall, the models explanations are surprisingly convincing, opening the possibility to obtain non-trivial insight into data sets where little prior knowledge exists.

For the movie reviews dataset we show one example in Table 2 (more in Appendix C.2). In general we observe that the positive channel foremost tends to highlight positive adjectives and the negative channel highlights negative adjectives respectively. This leads to the fact that in some cases the classification result of the network can be guessed by looking at the importance activation map due to the fact that positive reviews tend to contain a lot of positive adjectives and negative reviews accordingly.
However, the explanations of our model are unable to grasp the concepts of sarcasm and negations.

Table 2: Example for movie review sentiment classification dataset. Punctuation and capitalization were removed in pre-processing. Higher node importance values are indicated by more intense green highlights.

Negative	Positive
overall avengers endgame was a remarkable movie and a worthy culmination of the mcu up to this point there were some genuinely heartbreaking moments and breathtaking action sequences but to be honest some of the movies i had to sit through to get here were not worth it some of the early mcu movies and series leading up to this finale i found rather bland unfunny and sometimes just downright bad but this movie was one of the best movies i have seen in a while	overall avengers endgame was a remarkable movie and a worthy culmination of the mcu up to this point there were some genuinely heartbreaking moments and breathtaking action sequences but to be honest some of the movies i had to sit through to get here were not worth it some of the early mcu movies and series leading up to this finale i found rather bland unfunny and sometimes just downright bad but this movie was one of the best movies i have seen in a while

Even if positive adjectives are negated they are still in the positive explanation channel and the same is true for sarcastic remarks. We believe that this is part of the reason why our model does not reach the benchmark classification accuracy provided by DeYoung et al. (2020) usually achieved by language models. However, our model's accuracy is consistent with that achieved by other GNNS as reported by Rathee et al. (2022).

5 Limitations

Despite the encouraging experimental results, there are limitations to the proposed MEGAN architecture: Firstly, there is no guarantee that each channel's explanations align correctly according to their predetermined interpretations. This alignment is mainly promoted through an additional training step, whose influence on the network is dependent on the hyperparameter γ. We observed "explanation leakage" and "explanation flipping" on rare occasions even with reasonable values of γ. In those cases explanations of one channel may faintly appear in the opposite channel or a particularly disadvantageous initialization of the network causes explanations to develop in the exact opposite channel relative to their assigned interpretation. The second limitation is in the design of the explanation training step itself, which uses only the importance values of each channel to approximately solve an augmentation of the main prediction task. This process essentially amounts to reducing the problem to a subgraph detection / counting task. While there are important real-world applications that can be approximated as such, it still presents an important limit to the expressiveness of our model.

6 Conclusion

In this paper we present MEGAN, a multi-explanation graph attention network architecture. Our model uses attention mechanisms to produce node and edge attribution explanations along multiple channels for graph classification and regression tasks. Thereby, the number of explanations can be chosen independently of the task specifications. This may be an important step in developing richer and more interpretable explanations for graph neural networks in the future. In this work, we investigate one specific choice of explanation channels first: For graph regression tasks we use two explanation channels to capture explanations that explain positive and negative influences relative to a value of reference. For classification tasks we choose the number of explanation channels to be equal to the number of classes, such that each class receives its own explanation. We promote the individual explanation channels to behave according to those predetermined interpretations by applying a modified training procedure which includes an additional self-supervised explanation-only training step in each iteration.

We validate our model on a synthetic graph regression dataset and show that it produces explanations with quality similar to GNNExplainer for the single-explanation case. When switching to the dual-explanation case, we argue that explanations become more interpretable. Furthermore, we show that when using the additional explanation training step, explanations become more faithful with respect to the outcome of the network. Furthermore, we can show that an explanation-supervised version of our model achieves the highest similarity to explanation ground truth as well as fidelity w.r.t. to the respective channel's intended interpretation. Finally, we demonstrate the capabilities of our model on two real-world datasets: Prediction of water solubility for molecular graphs and sentiment classification for movie reviews. We show that the explanations generated by our model are coherent with human intuition for both tasks.

Inspired by the recent successes of explanation supervision (Linsley et al., 2019; Fernandes et al., 2022) we intend to leverage the differentiability of our model, in order to investigate the potential benefits of explanation supervision using human-generated explanations for more complex real-world graph related tasks.

References

Boobier, S., Osbourn, A., and Mitchell, J. B. O. (2017). Can human experts predict solubility better than computers? *Journal of Cheminformatics*, 9(1):63.

Boyd, A., Tinsley, P., Bowyer, K., and Czajka, A. (2022). CYBORG: Blending Human Saliency Into the Loss Improves Deep Learning. arXiv:2112.00686 [cs].

Brody, S., Alon, U., and Yahav, E. (2022). How Attentive are Graph Attention Networks?

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., and Wallace, B. C. (2020). ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Doshi-Velez, F. and Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]*. arXiv: 1702.08608.

Fernandes, P., Treviso, M., Pruthi, D., Martins, A. F. T., and Neubig, G. (2022). Learning to Scaffold: Optimizing Model Explanations for Teaching. arXiv:2204.10810 [cs].

Friederich, P., Krenn, M., Tamblyn, I., and Aspuru-Guzik, A. (2021). Scientific intuition inspired by machine learning-generated hypotheses. *Machine Learning: Science and Technology*, 2(2):025027. Publisher: IOP Publishing.

Gao, Y., Sun, T., Bhatt, R., Yu, D., Hong, S., and Zhao, L. (2021). GNES: Learning to Explain Graph Neural Networks. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 131–140. ISSN: 2374-8486.

Henderson, R., Clevert, D.-A., and Montanari, F. (2021). Improving Molecular Graph Neural Network Explainability with Orthonormalization and Induced Sparsity. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4203–4213. PMLR. ISSN: 2640-3498.

Huang, Q., Yamada, M., Tian, Y., Singh, D., and Chang, Y. (2022). GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–6. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Huuskonen, J. (2000). Estimation of Aqueous Solubility for a Diverse Set of Organic Compounds Based on Molecular Topology. *Journal of Chemical Information and Computer Sciences*, 40(3):773–777. Publisher: American Chemical Society.

Jiménez-Luna, J., Grisoni, F., and Schneider, G. (2020). Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks.

Krenn, M., Pollice, R., Guo, S. Y., Aldeghi, M., Cervera-Lierta, A., Friederich, P., Gomes, G. d. P., Häse, F., Jinich, A., Nigam, A., Yao, Z., and Aspuru-Guzik, A. (2022). On scientific understanding with artificial intelligence. arXiv:2204.01467 [physics].

Landrum, G. (2010). RDKit: Open-source cheminformatics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Linsley, D., Shiebler, D., Eberhardt, S., and Serre, T. (2019). Learning what and where to attend.

McCloskey, K., Taly, A., Monti, F., Brenner, M. P., and Colwell, L. J. (2019). Using attribution to decode binding mechanism in neural network models for chemistry. *Proceedings of the National Academy of Sciences*, 116(24):11624–11629. Publisher: Proceedings of the National Academy of Sciences.

OpenAI (2022). DALL·E 2.

Pan, X., Song, S., Chen, Y., Wang, L., and Huang, G. (2022). PLAM: A plug-in module for flexible graph attention learning. *Neurocomputing*, 480:76–88.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., and Hoffmann, H. (2019). Explainability Methods for Graph Convolutional Neural Networks. pages 10772–10781.

Pruthi, D., Bansal, R., Dhingra, B., Soares, L. B., Collins, M., Lipton, Z. C., Neubig, G., and Cohen, W. W. (2022). Evaluating Explanations: How Much Do Explanations from the Teacher Aid Students? *Transactions of the Association for Computational Linguistics*, 10:359–375.

Qiao, T., Dong, J., and Xu, D. (2018). Exploring Human-Like Attention Supervision in Visual Question Answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. page 12.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8821–8831. PMLR. ISSN: 2640-3498.

Rathee, M., Funke, T., Anand, A., and Khosla, M. (2022). BAGEL: A Benchmark for Assessing Graph Neural Network Explanations. arXiv:2206.13983 [cs].

Reiser, P., Eberhard, A., and Friederich, P. (2021). Graph neural networks in TensorFlow-Keras with RaggedTensor representation (kgcnn). *Software Impacts*, 9:100095.

Sanchez-Lengeling, B., Wei, J., Lee, B., Reif, E., Wang, P., Qian, W., McCloskey, K., Colwell, L., and Wiltschko, A. (2020). Evaluating Attribution for Graph Neural Networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 5898–5910. Curran Associates, Inc.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489. Number: 7587 Publisher: Nature Publishing Group.

Sorkun, M. C., Khetan, A., and Er, S. (2019). AqSolDB, a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds. *Scientific Data*, 6(1):143. Number: 1 Publisher: Nature Publishing Group.

Sorkun, M. C., Koelman, J. M. V. A., and Er, S. (2021). Pushing the limits of solubility prediction via quality-oriented data selection. *iScience*, 24(1):101961.

Stacey, J., Belinkov, Y., and Rei, M. (2022). Supervising Model Attention with Human Explanations for Robust Natural Language Inference. arXiv:2104.08142 [cs].

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph Attention Networks.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354. Number: 7782 Publisher: Nature Publishing Group.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. (2019). GNNExplainer: Generating Explanations for Graph Neural Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yuan, H., Yu, H., Gui, S., and Ji, S. (2022). Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–19. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

A Dataset Details

A.1 RbMotifs

RbMotifs (red & blue motifs) is a synthetic graph regression dataset. It consits of 5000 randomly generated graphs with node counts between 10 and 40. Each node is associated with three node feature values in the range $[0, 1]$, which represent RGB color values. Node colors are randomly sampled from a predefined set of 7 colors. Edges are undirected and unweighted. Additionally to the random nodes, graphs may be seeded with one of four subgraph motifs, each associated with a constant value. The target value for each graph is then calculated as the sum of all the motif-specific values of all motifs which are contained in the graph and a random component $\delta \sim \mathcal{U}(-0.5, 0.5)$:

$$y^{\text{true}} = \sum_{m} y^{m} + \delta \tag{12}$$

where y^m is the constant value associated with the m-th motif contained in the graph. The four subgraph motifs consist of specifc combinations of colored nodes, where two motifs form a similar pair, whose nodes are either dominated by red or blue nodes. This means there are two possible red motifs and two blue motifs. The red motifs are associated with a positive constant value, while the blue motifs are associated with negative values. Figure 5 illustrates this and shows some examples graphs from the dataset. Figure 6 shows the distribution of color values as well as the distribution of how many motifs are contained within the graphs.

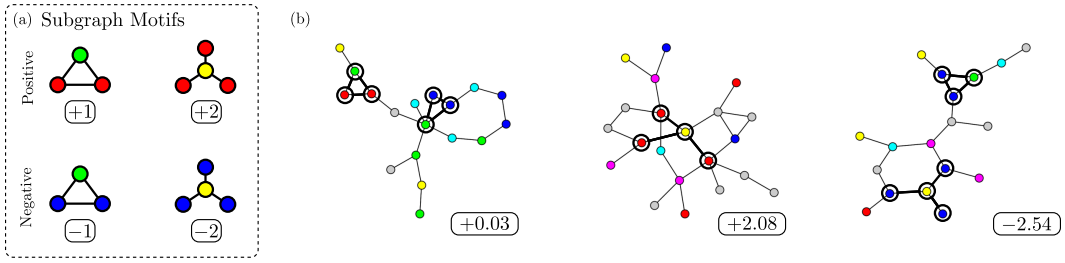

Figure 5: (a) The subgraph motifs used in the RbMotifs dataset and their associated values. (b) Example graphs from the dataset annotated with their target value

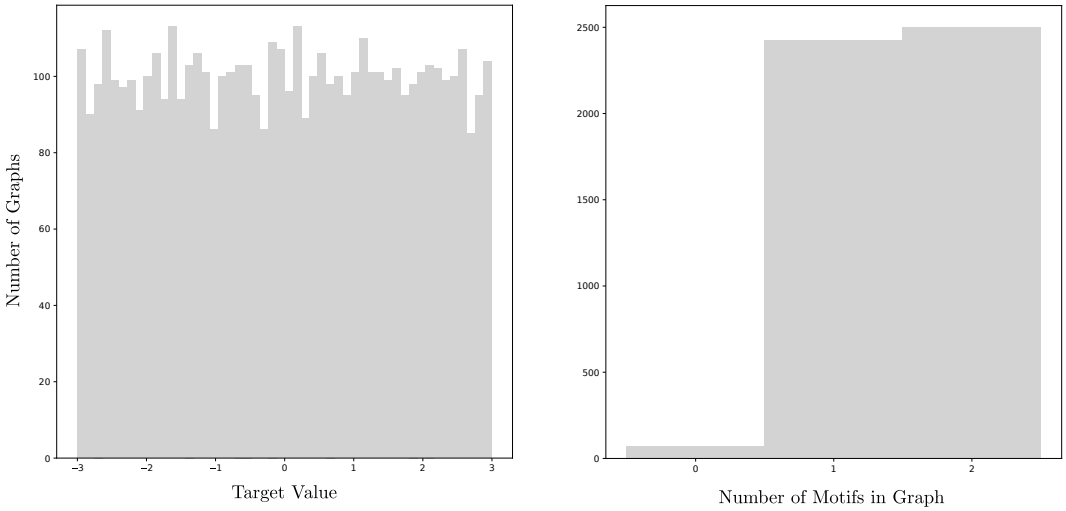

Figure 6: (a) Distribution of target values for the RbMotifs dataset. The limits -3 and $+3$ were imposed as hard limits during the generation process. (b) The distribution of the number of motifs contained within graphs. For 2 motifs, any combination exists.

A.2 Solubility

We use the AqSolDB dataset introduced by Sorkun et al. (2019), which consists of 9982 chemical compounds, annotated with measurements of water solubility (logS) at room temperature. The dataset was created by merging multiple existing datasets. We follow the instructions in (Sorkun et al., 2021) and and use all of the 1290 compounds originally contained within the so-called *Huuskonen dataset* (Huuskonen, 2000) as the test set for all experiments. For the remaining elements we apply the same pre-processing steps:

- Remove all compounds which do not contain a carbon atom
- Remove adjoined mixtures
- Remove compounds that contain charged atoms

After the application of these pre-processing steps, that leaves approx. 7000 molecules in the training dataest. We process all of the SMILES strings into molecular graphs using RDKit (Landrum, 2010).

A.3 MovieReviews

MovieReviews is a real world text classification datasets consisting of 2000 movie reviews from IMDB. Each movie review is to be classified into one of the two sentiment classes "positive" and "negative". Within the dataset, both classes are represented equally. We take this dataset from the ERASER benchmark (DeYoung et al., 2020) and process it in a similar way as it was done in Rathee et al. (2022) as well.

First, the strings are converted into token lists, where tokens are individual words and other sentence elements such as punctuation. In this step we remove punctuation and capitalization. The string tokens are then converted into a 50 dimensional numeric feature vector by using a pre-trained GLOVE (Pennington et al., 2014) model. We convert the token lists into graphs by interpreting every token as a graph node and connecting each node with it's neighboring nodes (according to the order of the token list / sentence order) using a sliding window of size 2. This means that every word is connected to the 2 neighboring words in both directions. Edges are undirected and unweighted. For all experiments we use the canonical train-test split as defined by DeYoung et al. (2020).

B Experiment Details

B.1 Explanation Pre-processing

We pre-process all explanations of MEGAN an GNNExplainer by normalizing the attribution values to a $[0, 1]$ value range. This is done w.r.t to all explanation channels, which means that the relative differences between the explanation channels remain the same.

B.2 Evaluation Metrics

Explanation Accuracy. For cases where definite ground truth explanations are available, we measure the accuracy of the generated explanations by computing the are under the ROC curve (AUROC), as it is done in McCloskey et al. (2019), for the entire validation set. The AUROC value is in range $[0, 1]$, where 1.0 indicates a perfect classifier and 0.5 indicates a random classifier.

Sparsity. We calculate the sparsity as the percentage of nodes / edges contained in the binary version of the predicted node / edge importance vector. The binary version of these vectors is calculated with a threshold at 0.5.

Fidelity. For single-explanation cases, we calculate Fidelity as described in Yuan et al. (2022): The Fidelity value is the deviation of the predicted output when the given binary explanation is removed from the input of a particular sample. In this case removal means setting all the feature values of the corresponding input elements to zero. The binary version of the explanation vectors is calculated with a simple threshold at 0.5.

Since changes in a regressed value are harder to put into context than for a normalized classification output, which is always in the $[0, 1]$ value range, we also provide $\text{Fidelity}_{\text{rand}}$ as a reference. This

is the Fidelity value which results from a randomly generated explanation mask, which has the same sparsity value as the original explanation mask. Consequently, regression explanations can be considered faithful if their Fidelity values are significantly higher than the $\text{Fidelity}_{\text{rand}}$ baseline.

B.3 Synthetic Dataset Experiments

The first experiment on the RbMotifs dataset compares GNNExplainer with MEGAN models, which only use a single explanation channel ($K = 1$). In this experiment, the ground truth explanations are considered to consist of the union of all subgraph motifs that appear in the respective graph. The individual model and training parameters are reported in Table 3.
The GNNExplainer explanations are based on a standard multi-layer GCN (Kipf and Welling, 2017) network. The GCN network is trained on the same train set as the MEGAN models. Afterwards a GNNExplainer optimization is performed for each element of the test set to obtain the explanations. We note that we use a slightly modified implementation of GNNExplainer contained in KGCNN library Reiser et al. (2021).
The MEGAN models used in this first experiment use only the one explanation channel. Since the $K = 1$ case is not covered in Section 3.3, these models do not use any additional explanation step at all ($\gamma = 0$).

The second experiment compares different MEGAN configurations for the dual-explanation case $K = 2$. In this case, ground truth explanations are split into two channels. The first channel contains all blue (negative) motifs that appear in the graph and the second channel contains all red (positive) motifs. The individual model and training parameters are reported in Table 3.

The results of 50 independent repetitions of these experiments can be found in Table 1.

Table 3: Model and Training parameters used for the experiments with the synthetic dataset RbMotifs. The columns from left to right are: The model name, the learning rate, the batch size, the number of training epochs, the number of convolutional layers used for the network, the hidden units used for each of the convolutional layers, the hidden units used for the MLP layers, the sparsity coefficient and the total number of parameters of the model.

Model	LR	BS	Epochs	Depth	Conv. Units	MLP Units	β	# Param.
GCN	0.004	512	250	3	$(9, 9, 9)$	$(3, 1)$	-	2343
MEGAN[1]	0.004	512	250	3	$(9, 9, 9)$	$(3, 1)$	0.0	723
MEGAN[2]	0.01	512	250	3	$(3, 3, 3)$	$(3, 1)$	0.1	413
GNNExplainer	0.1	1	100	-	-	-	0.1	0

Table 4: Model and Training parameters used for the experiments with the real-world datasets. The columns from left to right are: The model name, the learning rate, the batch size, the number of training epochs, the number of convolutional layers used for the network, the hidden units used for each of the convolutional layers, the hidden units used for the MLP layers, the sparsity coefficient and the total number of parameters of the model. The specified dropout percentages indicate the dropout which is applied after *each* layer.

Dataset	LR	BS	Epochs	Depth	Conv. Units	MLP Units	β	# Param.
Solubility	0.001	512	250	5	$(45, 40, 35, 30, 25)$ 20% Dropout	$(30, 20, 10, 1)$ 0% Dropout	0.1	80907
MovieR.	0.001	50	100	5	$(45, 40, 35, 30, 25)$ 20% Dropout	$(30, 20, 10, 2)$ 0% Dropout	0.1	71558

B.4 Real-World Dataset Experiments

Additional to the experiments for the synthetic dataset we also perform experiments with two real-world datasets: The prediction of water solubility for chemical compounds and the sentiment classification of movie reviews. For both experiments we only report one configuration of the MEGAN architecture. The model and training parameters can be found in Table 4. For each experiment we briefly optimize the hyperparameters manually. Most importantly, we find that dropout regularization proves increasingly useful for increasing numbers of node features and layers. The given dropout percentages are applied after each layer.

The results of 50 independent repetitions of the solubility experiment can be found in Table 5. We also report the results of Sorkun et al. (2021) for the same test set.
Over all repetitions, our model performs consistently well in terms of predictivity ($R^2 = 0.91$), although the results are slightly worse than those achieved by the consensus model employed by Sorkun et al. (2021). However, we especially point out the high Fidelity* value for our model. On the one hand this indicates that the explanation train step effectively promotes the learned explanations to remain truthful to the intended interpretations of the respective channels. On the other hand this also indicates that the explanations which are found by the model can indeed be interpreted as positive and negative influences for the concept of solubility in general.

Table 5: Results for computational experiments with solubility dataset. We report the median value for 50 independent experiment repetitions in black, as well as the 75th percentile (upper) and 25th percentile (lower) of the distribution in gray.

Source	Model	RMSE	R^2	Node Sparsity	Edge Sparsity	Fidelity*
Sorkun et al. (2021)	Consensus	<u>0.54</u>	<u>0.93</u>	-	-	-
ours	$\text{MEGAN}^2_{0.5}$	$0.59 \left\vert \begin{smallmatrix} 0.60 \\ 0.58 \end{smallmatrix} \right.$	$0.91 \left\vert \begin{smallmatrix} 0.91 \\ 0.91 \end{smallmatrix} \right.$	$0.29 \left\vert \begin{smallmatrix} 0.42 \\ 0.18 \end{smallmatrix} \right.$	$0.25 \left\vert \begin{smallmatrix} 0.37 \\ 0.18 \end{smallmatrix} \right.$	$7.15 \left\vert \begin{smallmatrix} 8.37 \\ 5.13 \end{smallmatrix} \right.$

The results of 50 independent repetitions of the MovieReviews experiment can be found in Table 6. We also report the results of DeYoung et al. (2020), who use a BERT encoder and LSTM network, and Rathee et al. (2022), who use the same pre-processing steps to provide a baseline for a simple GCN network.
Overall, DeYoung et al. (2020) clearly show the best classification performance. We believe this is due to their usage of a state-of-the art BERT language model, which has recently proven very powerful for multiple language processing tasks. To our surprise however, our results are only marginally better than those of a GCN baseline model reported by Rathee et al. (2022). We hypothesize that this is due to our usage of a 50 dimensional GLOVE model instead of the full 300 dimensional version used by Rathee et al. (2022). In the future, we want to investigate different language encoder models in junction with our model.

Table 6: Results for computational experiments with solubility dataset. We report the median value for 50 independent experiment repetitions in black, as well as the 75th percentile (upper) and 25th percentile (lower) of the distribution in gray.

Source	Model	F1	Node Sparsity	Edge Sparsity	Fidelity*
DeYoung et al. (2020)	BERT+LSTM	<u>0.97</u>	-	-	-
Rathee et al. (2022)	GLOVE+GCN	0.85	-	-	-
ours	$\text{MEGAN}^2_{1.0}$	$0.86 \left\vert \begin{smallmatrix} 0.87 \\ 0.84 \end{smallmatrix} \right.$	$0.01 \left\vert \begin{smallmatrix} 0.01 \\ 0.00 \end{smallmatrix} \right.$	$0.00 \left\vert \begin{smallmatrix} 0.00 \\ 0.00 \end{smallmatrix} \right.$	$1.00 \left\vert \begin{smallmatrix} 1.00 \\ 1.00 \end{smallmatrix} \right.$

C Additional Examples

C.1 Solubility

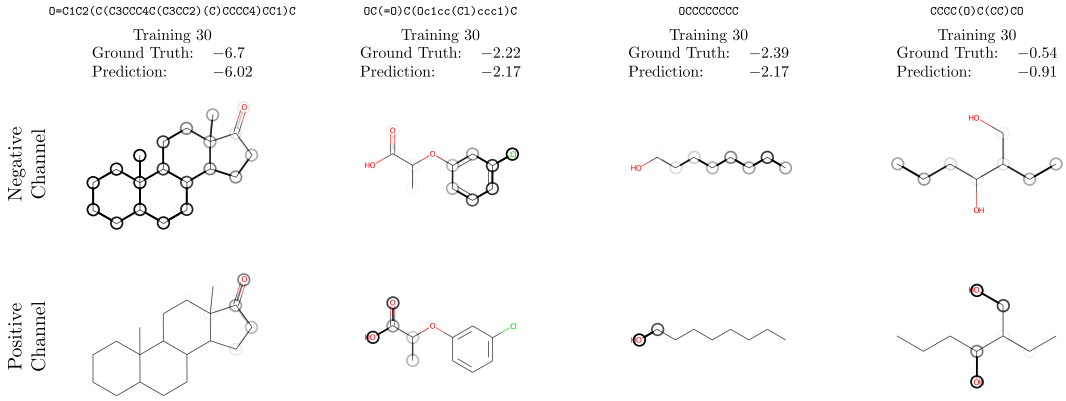

Figure 7: Examples which show that the model learns to associate carbon groups with a negative influence on the solubility value and oxygen functional groups (especially OH groups) with positive influences on the solubility value.

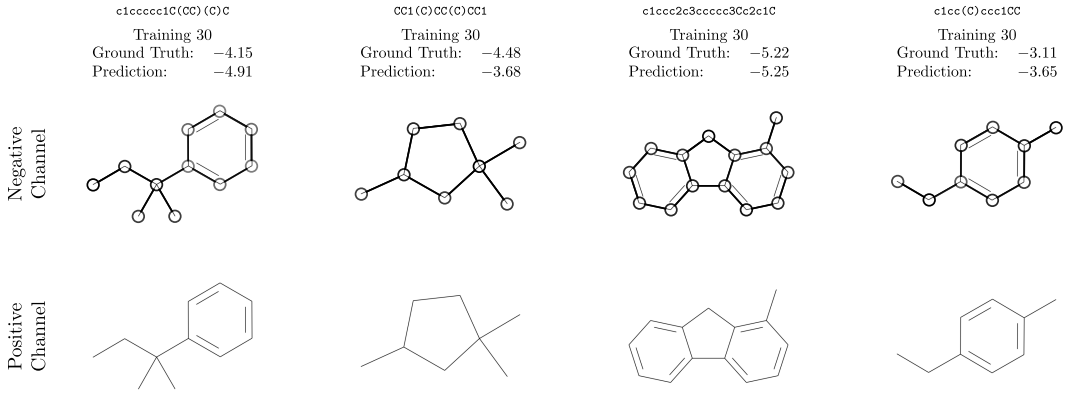

Figure 8: Examples which show that in the absence of any atoms carbon hydrogen, the positive channel is usually not activated at all, further supporting the assumption that the model learns to associate carbon structures with low solubility. We point out that although the model provides a heuristical explanation consistent with chemistry knowledge here, this still shows a limitation of simple attributional explanations: Despite being explained in a similar fashion, the samples shown here still vary considerably in their actual solubility value. We argue that in such cases two simple attributional explanations as used here are not sufficient to accurately communicate the underlying reason for those differences in value.

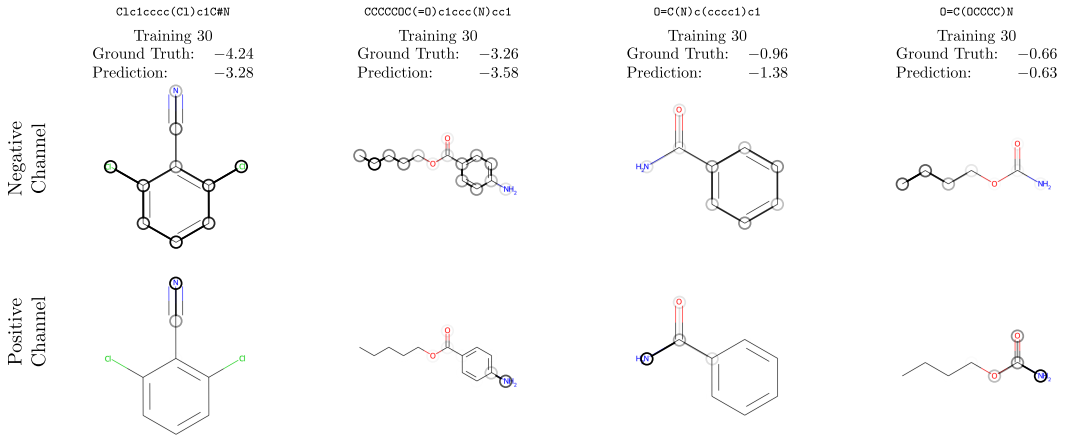

Figure 9: Examples which show that the model learns to associate nitrogen functional groups with positive influences on the solubility value aswell.

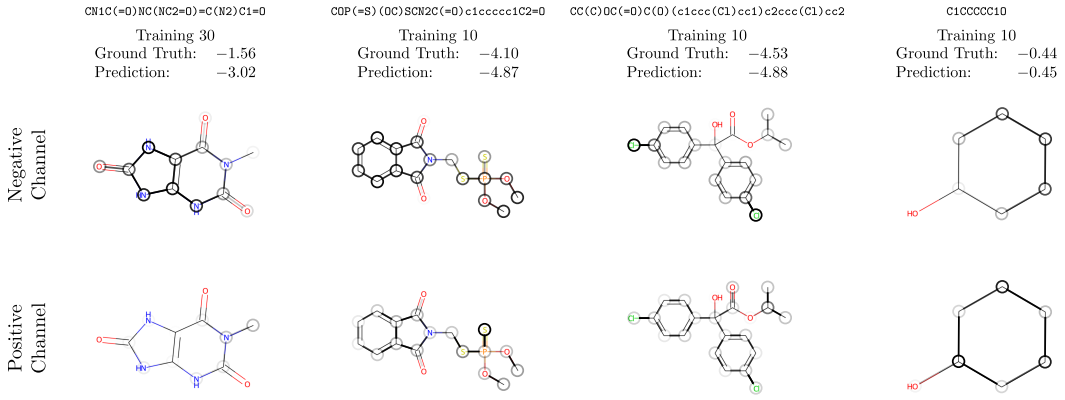

Figure 10: Examples which show that the quality of the explanations can be inconsistent within as well as in between independent repetitions of model training. The first sample is taken from the 30th repetition of the solubility experiment, from which all the good examples of the previous figures have been drawn from as well. It incorrectly shows a strong activation of the negative channel and a weak activation of the positive channel, even though there are many characteristic oxygen and nitrogen functional groups present. In this case, the faulty explanation is actually reflected in the relatively large error in the model's prediction as well. The other three samples were drawn from the 10th repetition, which shows worse explanations overall. Despite the relatively accurate predictions, all three samples show very indiscriminate explanations, that feature a lot of similar activations in both channels.

C.2 Movie Reviews

Table 7: Example for movie review which contains positive and negative sentences, which are correctly sorted into the respective channels. However, this example also shows some sort of a bias by including the first sentence as a negative explanation. Objectively the first sentence does not contain any sentiment. The false explanation of the model is supposedly caused by the word "criminals" which is presumably often used in junction with negative adjectives.

Negative	Positive
a couple of criminals mario van peebles and loretta devine move into a rich family house in hopes of conning them out of their jewels however someone else steals the jewels before they are able to get to them writer mario van peebles delivers a clever script with several unexpected plot twists but director mario van peebles undermines his own high points with haphazard camera work editing and pacing it felt as though the film should have been wrapping up at the hour mark but alas there was still 35 more minutes to go daniel baldwin i ca n't believe i 'm about to type this gives the best performance in the film outshining the other talented members of the cast	a couple of criminals mario van peebles and loretta devine move into a rich family house in hopes of conning them out of their jewels however someone else steals the jewels before they are able to get to them writer mario van peebles delivers a clever script with several unexpected plot twists but director mario van peebles undermines his own high points with haphazard camera work editing and pacing it felt as though the film should have been wrapping up at the hour mark but alas there was still 35 more minutes to go daniel baldwin i ca n't believe i 'm about to type this gives the best performance in the film outshining the other talented members of the cast

Table 8: Example for an exclusively positive review. Due to the overall lack of negative adjectives, the negative channel isn't activated at all.

Negative	Positive
this three hour movie opens up with a view of singer guitar player musician composer frank zappa rehearsing with his fellow band members all the rest displays a compilation of footage mostly from the concert at the palladium in new york city halloween 1979 other footage shows backstage foolishness and amazing clay animation by bruce bickford the performance of titties and beer played in this movie is very entertaining with drummer terry bozzio supplying the voice of the devil frank guitar solos outdo any van halen or hendrix i 've ever heard bruce bickford outlandish clay animation is that beyond belief with zooms morphings etc and actually it does n't even look like clay it looks like meat	this three hour movie opens up with a view of singer guitar player musician composer frank zappa rehearsing with his fellow band members all the rest displays a compilation of footage mostly from the concert at the palladium in new york city halloween 1979 other footage shows backstage foolishness and amazing clay animation by bruce bickford the performance of titties and beer played in this movie is very entertaining with drummer terry bozzio supplying the voice of the devil frank guitar solos outdo any van halen or hendrix i 've ever heard bruce bickford outlandish clay animation is that beyond belief with zooms morphings etc and actually it does n't even look like clay it looks like meat

Table 9: Example which shows that the model currently doesn't understand negations and sarcasm. We point out that the partial sentence "never a bad thing" in the middle of the review is sorted into the negative channel. Another example is the first sentence: The praise it features is meant sarcastically, but it is still sorted into the positive channel.

Negative	Positive
burnt money is the perfect festival film it will show once or twice and then no one thankfully will ever have to hear from it again this film from the seattle international film festival 2001 emerging masters series is easily one of the year worst billed as a gay ' bonnie and clyde this gritty film from director marcelo pi eyro has its only highlight in a well designed title sequence two gay lovers get involved in a bank robbery that makes a gang leader whose plan they screwed up angry this causes the gang leader to send his boys out to get the gay guys one of whom may not actually be gay hiding out in a prostitute apartment the two men must fight off police and gang members in a very long showdown for the movie conclusion if caught they risk losing all the money and their love as an added emotional bonus one of the gay men is dying or something like that everything that happens is so quick and confusing i was completely lost clarity is n't exactly this movie striving virtue so it was a little hard to pick up not much could have really happened though the main events in this long two hour film are explicit homosexual and heterosexual sex graphic drug use extreme violence and strong language lots of explicit material is never a bad thing when there a reason but there no purpose to anything in this film most of the sex and violence scenes come off as silly while the heavy drug use comes off as ridiculous and depressing it appears pi eyro who co wrote with marcelo figueras from a novel by ricardo piglia purposefully adds more blood and lovemaking for his own amusement he makes the actors as sweaty and dirty as possible makes them snort cocaine gives them guns and condoms and lets them go burnt money is pointless the performances are bad it tries to thrill and shock but only causes boredom god forbid it will ever get a distributor another disappointing film from this year so called emerging masters series pass on by	burnt money is the perfect festival film it will show once or twice and then no one thankfully will ever have to hear from it again this film from the seattle international film festival 2001 emerging masters series is easily one of the year worst billed as a gay ' bonnie and clyde this gritty film from director marcelo pi eyro has its only highlight in a well designed title sequence two gay lovers get involved in a bank robbery that makes a gang leader whose plan they screwed up angry this causes the gang leader to send his boys out to get the gay guys one of whom may not actually be gay hiding out in a prostitute apartment the two men must fight off police and gang members in a very long showdown for the movie conclusion if caught they risk losing all the money and their love as an added emotional bonus one of the gay men is dying or something like that everything that happens is so quick and confusing i was completely lost clarity is n't exactly this movie striving virtue so it was a little hard to pick up not much could have really happened though the main events in this long two hour film are explicit homosexual and heterosexual sex graphic drug use extreme violence and strong language lots of explicit material is never a bad thing when there a reason but there no purpose to anything in this film most of the sex and violence scenes come off as silly while the heavy drug use comes off as ridiculous and depressing it appears pi eyro who co wrote with marcelo figueras from a novel by ricardo piglia purposefully adds more blood and lovemaking for his own amusement he makes the actors as sweaty and dirty as possible makes them snort cocaine gives them guns and condoms and lets them go burnt money is pointless the performances are bad it tries to thrill and shock but only causes boredom god forbid it will ever get a distributor another disappointing film from this year so called emerging masters series pass on by

D Relevance to Material Sciences

XAI methods aim to improve trust within the human-AI relationship and thus promote a more widespread adoption of powerful AI solutions into every facet of human society. Beyond this factor of trust in a researcher-AI relationship, we believe that further development of XAI methodology will eventually be able to directly support the process of scientific discovery itself. This is because human researchers and computational models possess very complementary strengths. While humans provide the required creativity and the ability to theorize on higher levels of abstraction, methods of machine learning have an unparalleled capability to learn the hidden relationships within vast amounts of data. Nowadays, the top AI models are able to perform highly complex tasks on a level similar or above to that of average humans. This ranges from the proficiency of playing highly complex games (Silver et al., 2016; Vinyals et al., 2019) to image recognition and generation (Ramesh et al., 2021; OpenAI, 2022), natural language processing (Radford et al., 2018; Lewis et al., 2020) and even various tasks within the natural sciences (Boobier et al., 2017). We argue that if such highly capable AI models were able to communicate their learned knowledge effectively and efficiently to an audience of human researchers that this could provide a valuable tool in the pursuit of new scientific insight (Friederich et al., 2021; Krenn et al., 2022). Especially in the domain of material sciences, a model trained to accurately predict a complex material property could teach us it's hidden structural causality, helping in turn to accelerate the discovery and synthesis of new materials.

We propose our MEGAN architecture as a first step towards this goal. Our model mainly aims to expand the explanatory capabilities of the GNN approach. Unlike the common post-hoc explanability methods our model is *self-explaining* as it is coined by Jiménez-Luna et al. (2020). This means that the explanation mechanism is built into the network architecture itself, where the explanations themselves are being directly involved in the calculation of the network's main prediction. For one thing, this results in explanations which are more faithful in describing the model's actual behavior. Additionally, we design our architecture to have the number of explanation channels be independent of the task specifications. We argue that such an increased flexibility will be important for the development of more expressive and granular forms of explanation in the future. Encouraged by the results obtained for the prediction of water solubility, we specifically plan to investigate methods to meaningfully expand this approach to different choices of explanation channels than the ones presented in this work. We believe that this has the potential to reveal even more fine-grained insights about the structure property relationship of real-world problems in chemistry and material science.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The abstract however focuses more on the relevance to the context of material sciences than the original submission.
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [No] We cannot conceive of any immediate negative social impacts of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 3.4
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Our experimental results are reported in the form of the median as well as the 25th and 75th percentile of the distribution
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Computational experiments do not have any out-of-the-ordinary requirments, they were all executed on a personal computer.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] The origin of all used datasets is cited as well as the main software library used for the implementation.
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] All used datasets are publicly available
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]