A Dataset Statistics

The statistics of datasets used for evaluating TAGE are shown in Table 6.

Table 6: Statistics of multitask datasets used for explanation quality evaluation. The column "Total" under MoleculeNet indicates the total number of commonly studied tasks from MoleculeNet.

	MoleculeNet					PPI	EPN
	HIV	BBBP	BACE	Sider	Total		
# of Graphs	41127	2039	1513	1427	–	24	1
Avg. # of Nodes	25.53	24.05	34.12	33.64	–	56,944	5.86 mn.
Avg. # of Edges	27.48	25.94	36.89	35.36	–	818,716	63.07 mn.
# of Tasks	1	1	1	27	227	121	3

B Implementation Details

Structure of explainer. Our implementation is based on Pytorch [19], Pytorch Geometric [2], and Dive-into-graphs [14]. We implement the explainer with a linear projection f_p that maps the condition vector $\boldsymbol{p}$ to the same dimension as concatenated embeddings, and a 2-layer MLP with ReLU activation that maps concatenated embeddings with the mask to the important score.

Implementation of training objectives. We adopt the Jason-Shannon Estimator as the lower bound for mutual information maximization for the two public datasets. For graph-level tasks, given a mini-batch of N samples, we consider the embeddings of a graph G and its subgraph G_s as a positive pair (with N positive pairs in total), and the embeddings of a graph G_i and the subgraph $G_{j,s}$ of another sample as a negative pair (with $N^2 - N$ pairs in total). For node-level tasks, we still randomly sample N nodes from the entire graph at each iteration and compute the contrastive losses on original embeddings of the N nodes, and embedding of the N nodes when the important subgraph is selected, respectively for each node. Similar to graph-level tasks, we consider embeddings of the same node (in the original graph or in the subgraph) as a positive pair, and embeddings of node i in the full graph and node j in the selected subgraph as a negative pair ($N^2 - N$ in total).

Impirical observation of Laplace distribution. We show examples of gradient distribution of multiple tasks in Figure 5. The absolute value of gradients of three tasks are shown in orange, blue, and green, respectively.

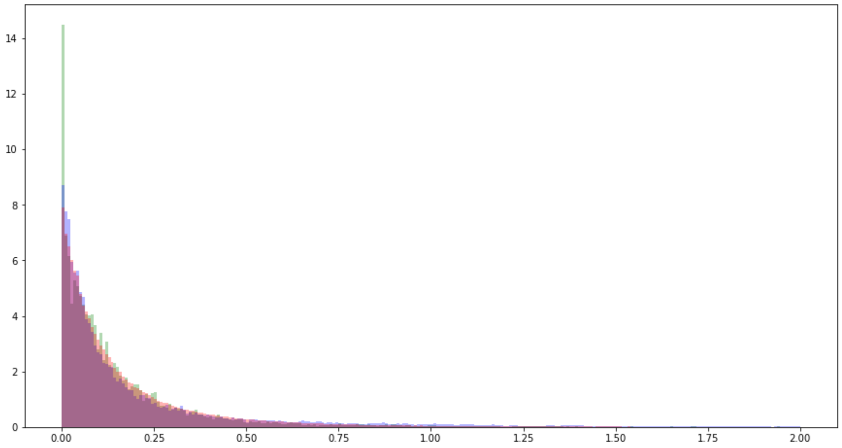

Figure 5: Distribution of downstream model gradient absolute values on three different tasks from PPI.

Training configurations. We set the hyperparameters in the size regularization term to $\lambda_s = 0.05$ and $\lambda_e = 0.002$, respectively. For the graph-level explanation on MoleculeNet, we train the embedding explainer on ZINC-2M with a learning rate of $1e - 4$ and mini-batch size of 256 for one epoch. The random condition vectors are generated from Laplace distribution $Laplace(0, 0.2)$. For the node-level explanation on PPI, we train the embedding explainer on PPI without labels with a learning rate of $5e-6$ and a mini-batch size of 4 for one epoch. The random condition vectors are generated from Laplace distribution $Laplace(0, 0.1)$. For the EPN dataset, we train the embedding explainer with InfoNCE loss, learning rate $1e-4$, and mini-batch size 16. The random condition vectors are generated from Laplace distribution $Laplace(0, 0.25)$. The hyperparameters in the size regularization term are set to $\lambda_s = 0.5$ and $\lambda_e = 0$ for the stable training with InfoNCE.

Evaluation. In molecule and protein property prediction, we are usually interested in the positive samples, *i.e.*, the existence of what substructure leads to a certain property. For learning-based baseline methods, we find it common that only one class of the two has a good explanation, and the class with higher explanation quality is not necessarily the positive class. For example, PGExplainer has a near-to-zero fidelity score for the positive class of SIDER. We hence compare only the higher fidelity score among the two classes for all explanation methods and datasets.

C Fidelity and Sparsity

Given a set of graphs $\{G_i\}$ and node masks $\boldsymbol{m}$ predicted by the explainer, the fidelity score and the sparsity score are computed as follows.

$$Fidelity^{\ prob} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left[f(G_i)_{c_i} - f(G_i^{\mathbf{1}-\boldsymbol{m}_i})_{c_i} \right], \tag{7}$$

$$Sparsity\ = \frac{1}{N} \sum\nolimits_{i=1}^{N} |\boldsymbol{m}_i|/|V_i|, \tag{8}$$

where N denotes the number of graphs or nodes to be explained, f denotes the GNN model associated with a specific downstream task, c_i denotes the class of interest, which can be either the labeled class or the original predicted class, G_i and $G_i^{\mathbf{1}-\boldsymbol{m}_i}$ denote the original graph and graph with important nodes removed, respectively. Explanations with both scores higher are better.

D Additional Results for Universal Explanation Ability

Comparison of explanation performance when trained on different datasets. Specifically for the MoleculeNet dataset, as there is a larger unlabeled dataset, ZINC, available for the first stage training of the encoder, the training of our explainer is also performed on the ZINC dataset. For a more strict comparison with the baseline explainer who is trained on individual MoleculeNet datasets, we additionally evaluate the explanation quality when the same individual MoleculeNet dataset is used to train TAGE. The results are shown in Table 7. When trained on the same datasets individually, TAGE still performs better than the baseline explainer in terms of fidelity scores. In the individual dataset case, we need to train different explainers, similarly to the training of PGExplainer, as the datasets for the four tasks are different.

Table 7: An ablation on training TAGE on different datasets (ZINC v.s. individual MoleculeNet datasets).

Method	BACE	HIV	BBBP	SIDER
PGExplainer	0.252 ±0.340	0.473 ±0.404	0.182 ±0.169	0.444 ±0.391
TAGE (individual)	**0.402 ±0.281**	0.541 ±0.330	**0.202 ±0.157**	0.516 ±0.292
TAGE (ZINC)	0.378 ±0.293	**0.595 ±0.321**	0.193 ±0.161	**0.521 ±0.278**

Comparison with the causality-based explainer GEM. On the BACE dataset and task, we additionally compare TAGE with another recent SOTA learning-based method GEM [12] whose explainer is trained based on the Granger causality in Table 8. Note that GEM is not originally proposed under our setting. It assumes that there is a fixed number of important nodes when performing explanation and hence the final explanation is a boolean selection of nodes. We adapt GEM to compute fidelity scores under different sparsity scores by varying the threshold when generating explanation ground-truth with Granger causality.

Table 8: A comparison between TAGE, GEM, and PGExplainer on BACE in terms of fidelity scores when fixing the sparsity scores. For GEM, we vary the threshold when generating explanation ground-truth with Granger causality to obtain explanations with different sparsity scores.

Sparsity	0.90	0.85	0.80	0.75
TAGE	**0.3349**	**0.4992**	**0.5383**	**0.5309**
GEM [12]	0.2829	0.3607	0.4260	0.4035
PGExplainer	0.2521	0.3207	0.4605	0.5161

E Discussion and additional results of visualizations

While there are no ground-truth explanations for the molecular datasets, the validity of results produced by TAGE can be evidenced by multiple domain research. Take BACE for example, Jain and Jadhav [8] study multiple BACE-1 inhibitors that are similar to one presented in our results (Figure 4 - line 3). Inhibitors in Table 1–3 and 8 of [1] share the common "2-imidazoline" structure as explained by TAGE, whereas structures such as =O and -OCF$_3$ as explained by GNNE and PGE are not necessarily in an inhibitor. Moreover, inhibitors studied by Huang et al. [7] share the common "-C(=O)-C-N(H)-C(OH)-" chain structure as present in the explanation results by TAGE (Figure 4 - lines 1 and 2), whereas structures explained by other explainers are not necessarily for a molecule to be a BACE-1 inhibitor. Nevertheless, it's still fidelity scores that give the most reliable evaluation.

Additional visualization results on HIV and SIDER are shown in Figure 6 and Figure 7, respectively.

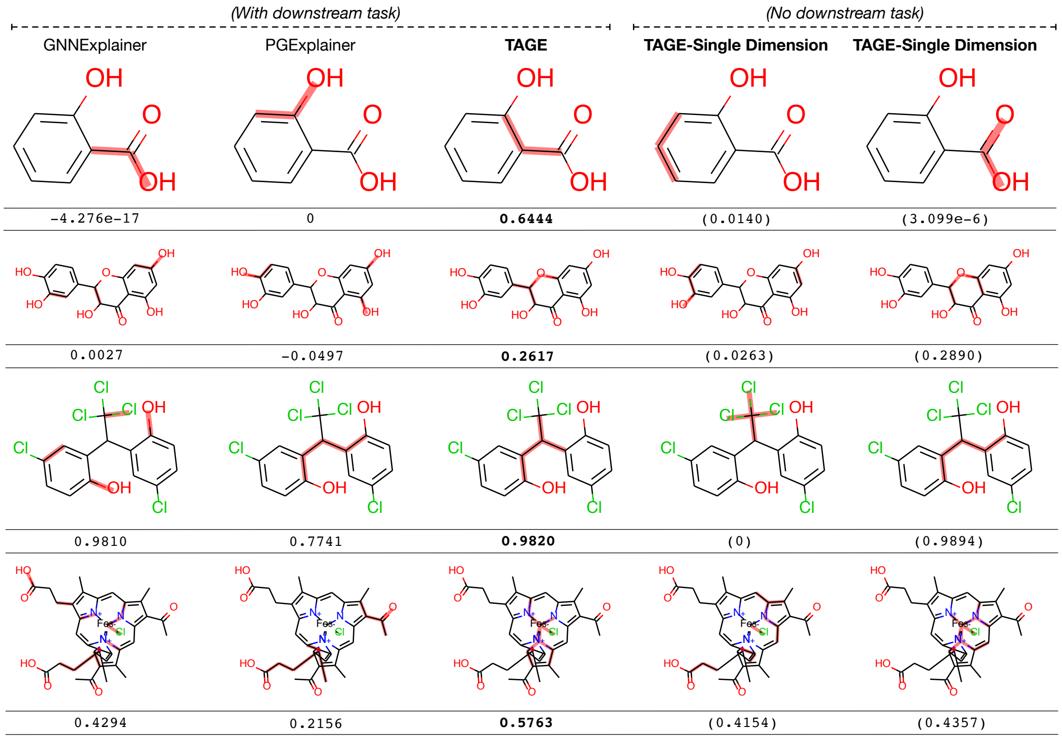

Figure 6: Visualizations on explanations to the GNN model for the HIV task. The top 10% important edges are highlighted with red shadow. The numbers below molecules are fidelity scores when masking out the top 10% important edges. The right two columns are explanations for two certain embedding dimensions without downstream tasks.

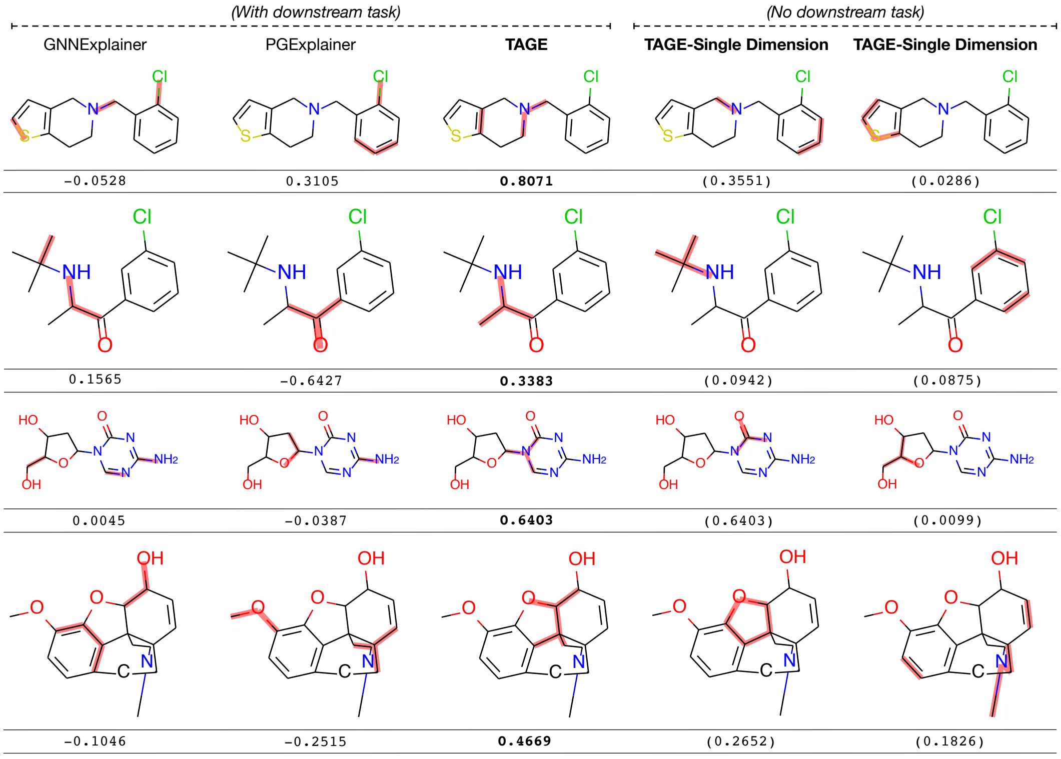

Figure 7: Visualizations on explanations to the GNN model for the SIDER task. The top 10% important edges are highlighted with red shadow. The numbers below molecules are fidelity scores when masking out the top 10% important edges. The right two columns are explanations for two certain embedding dimensions without downstream tasks.

F Experimental studies on the synthetic datasets BA-Shapes

We perform an additional evaluation on the BA-Shapes synthetic datasets used in GNNExplainer [34] and provided by Pytorch-Geometric [2]. The synthetic dataset is less complicated compared to real-world datasets. We train a 3-layer GCN for node classification with a training accuracy of 0.95. The AUC score (for importance edges) of TAGE is 0.999 compared to 0.963 and 0.925 of PGExplainer and GNNExplainer, respectively. Note that the baseline scores are from the PGExplainer paper. Some re-implementations[4][5] of PGExplainer can also achieve an AUC score of 0.999. Our purpose to show our score on BA-Shapes is to demonstrate that TAGE is on par with its baselines even when considering the typical single-task setting. Figure 8 visualizes 20 examples of explanations. TAGE is able to provide accurate explanations for all 20 examples.

G Discussion of limitations and potential solutions

Inductive learning of explanations. Our study focus on the setting of inductive learning of the explanation, *i.e.*, to train the explainer on a given dataset and perform inference on new coming data. There are many work conducted under the inductive setting, such as PGExplainer. All methods under this setting may have a potential limitation that the explainer may suffer from some dataset bias when training data and the data to be explained are inconsistent. This is an interesting problem that requires further investigation. However, we believe that this is a separate problem and applies to all inductive learning methods. In addition, the size of graph could be inconsistent for training and inference. To

[4] https://github.com/LarsHoldijk/RE-ParameterizedExplainerForGraphNeuralNetworks
[5] https://openreview.net/forum?id=tt04glo-VrT

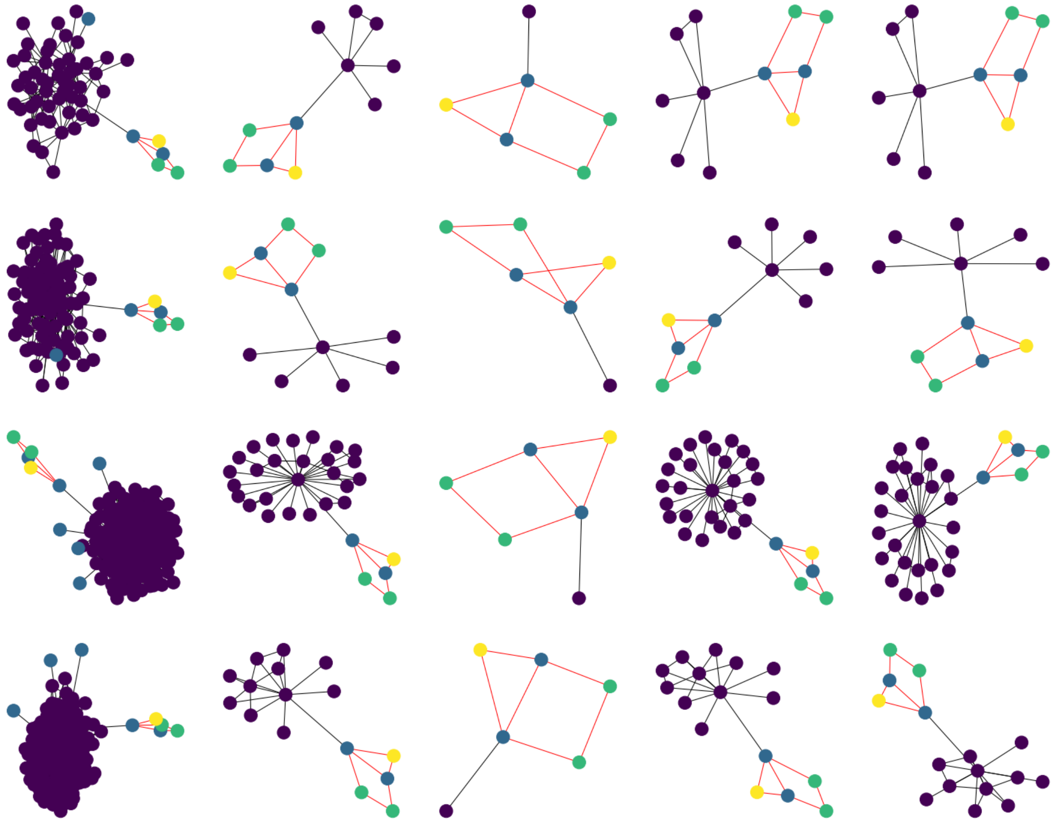

Figure 8: Visualizations on explanations to the synthetic dataset BA-Shapes.

tackle this issue, we obtain the substructure by selecting top k percentage of edges according to their important scores.

Expressiveness of explainers. The proposed method is the most suitable under the two-stage and multi-task settings. Our experiments show that even compared on the single task setting and on common datasets, TAGE can still have the same or better performance than baseline task-specific explanation methods. However, when the model, task, or datasets to be explained become too complicated, it is possible that the embedding explainer in TAGE may require more parameters to have enough expressiveness for the task-specific explanation. In those cases, one may adopt a similar fine-tuning approach as described by Wang et al. [27], or use task-specific explainers which are more efficient.

Black-box explanations. Similarly to our baseline method PGExplainer, our explainer relies on node embeddings as inputs to the explainer. In particular, the node embeddings serve as representations to allow explainers identify each node. It is required by any (inductively) learning based explanations to tell neural network-based explainers which edge they are looking at. A limitation of the inductive methods is that when the node embeddings may become unavailable when explaining a black-box model. The study of explaining black-box models (where only output is available) is a different direction of study in scenarios like attacking. Many current SOTA explanation approaches, such as Grad-Cam, GNN-LRP, and PGExplainer, fail under the black-box setting. However, if one would like to adapt our approach to the black-box setting, it is still feasible by adopting a surrogate model for the black-box model and perform explanation on the surrogate model. In addition, as mentioned above, the node embeddings are mainly used to identify which node the explainer is looking at, we does not necessarily require the original embedding. When node embedding are unavailable, we can still use any representation of nodes as long as it can identify the node based on its feature and topology.