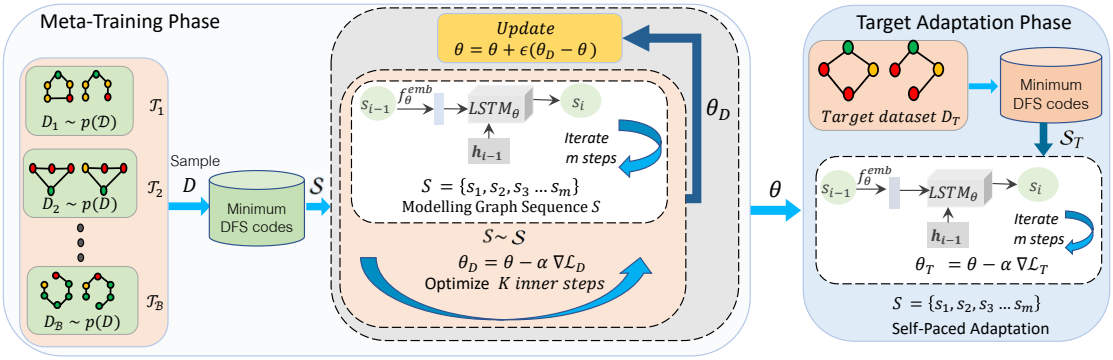

Figure 3: Architecture of the proposed approach (Task 2.1): a few shot generative model.

Analogously, a delete operations from an original node to a feature node represents, increasing the edge-weight by $\Delta \times \sigma$. Note that by making the perturbation granularity a function of the standard deviations, we automatically adapt to heterogeneous distributions in the feature space.

2.1.3 Task 1.3: Extending counter-factual reasoning to graph space

We will build inductive methods by extending our proposed framework for *the graph classification task*. Our counter-factual reasoning framework for node level tasks can be extended to graph classification by having a label over graphs and a GNN that performs a prediction on the entire graph instead of individual nodes. Typically, such GNNs work by aggregating all node embeddings into a single graph-level embedding, which is finally passed over a multi-layered perceptron (*MLP*) followed by a *Sigmoid* layer. In this proposal, we will study the problem for both node and graph classification.

To compute a graph-level embedding, we propose to introduce a special "summary" node to the graph. This summary node acts as a virtual node that is connected to all nodes from the original graph, representing the global information of the entire graph. Each edge from the summary node to an original node serves as a channel for aggregating information from that node. To aggregate node embeddings at the summary node, we propose to use an *attention mechanism* to compute attention coefficients that determine the importance of each node's embedding in the aggregation process.

Let us consider N as the total number of nodes (excluding the summary node) in the original graph, $\mathbf{h}_i$ the node embedding for node i, where $i = 1, 2, \cdots, N$, and $\mathbf{h}_{\text{summary}}$ the graph-level embedding to be computed at the summary node. The attention contribution of any node i towards the graph embedding is computed as, $e_i = \mathbf{a}^T(\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_{\text{sumpool}})$, where $\mathbf{a}$ is a learnable parameter vector, and $\mathbf{W}$ is a learnable weight matrix, and $\mathbf{h}_{sumpool} = \sum_{\forall i} \mathbf{h}_i$ is a *SumPool* over all original nodes. This operation assigns an importance score to all nodes of the graph as a function of its own embedding and the embeddings of the remaining nodes in the graphs. To transform the raw attention weights into a distribution, we apply *SoftMax activation*: $\alpha_i = \text{softmax}(e_i)$. The final graph representation is computed through a weighted aggregation over all original nodes of the graph, i.e., $\mathbf{h}_{\text{summary}} = \sum_{i=1}^{N} \alpha_i \mathbf{h}_i$.

With the ability to compute graph-level embeddings, the rest of the methodology for node-level reasoning naturally translates to graph-level reasoning. Since the graph embedding is also a function of the local topologies around each of its constituent nodes, and their feature characterizations, the action space remains the same as described earlier.

2.2 Research Thrust 2: Counterfactual Reasoning Beyond Classification

Goal. In this thrust, we will (1) use generative approaches to develop innovative techniques that can generate counterfactuals that are valid and feasible within the context of the domain; (2) build counterfactual explainers for GNN based models for graph combinatorial problems.

2.2.1 Task 2.1: Feasible Counterfactuals through Few-shot Graph Generative Modeling

Motivation. While generating new counterfactuals, it is crucial to ensure that the perturbations respect the underlying graph's constraints and maintain the graph's structural and functional integrity. For instance, in the case of molecular graphs, introducing arbitrary perturbations might lead to chemically infeasible molecules that violate molecular properties such as valency rules. Similarly, in communication networks, introducing unrealistic connections between nodes can undermine the validity of the counterfactual analysis.

Preliminary Results. Our initial evaluation examines whether existing algorithms preserve the topological properties of the test set. We compare the number of graphs forming a single connected component in the test set with those in their corresponding counterfactual explanations. Connectedness is a significant aspect of consideration, particularly in domains such as molecules, where disconnected graphs might not be meaningful. The results for RCExplainer [5], the state-of-the-art counterfactual explainer, are presented in Table 1. We observe statistically significant deviations in two out of four molecular datasets. This suggests a heightened probability of predicting counterfactuals that do not correspond to feasible molecules. Importantly, this finding underscores a limitation of counterfactual explainers, which has received limited attention within the research community.

Dataset	Expected Count	Observed Count	p-value
Mutagenicity	243.06	96	< 0.00001
Proteins	11.68	11	0.84
Mutag	10	8	0.53
AIDS	17.6	9	0.04

Table 1: **Feasibility:** Assessing the statistical significance of deviations in the number of connected graphs between the test set (expected counts) and their corresponding counterfactual explanations (observed) on molecular datasets. Statistically significant deviations with p-value< 0.05 are highlighted through shading.

Proposed Research

In this proposal, we aim to address the challenge of generating feasible counter-factual graphs over GNNs. Our goal is to develop innovative techniques that generate counter-factual graphs with minimal perturbations while ensuring that the resulting graphs remain valid and feasible within the context of the given graph domain. Towards that goal, we will explore neural graph generative models.

Overview of a Graph Generative Model. A neural graph generative model is a type of graph generative model that uses neural networks to learn and generate graphs [12, 14, 46, 58]. Unlike traditional graph generative models, which may rely on handcrafted features or heuristics to generate graphs, neural graph generative models leverage the power of neural networks to automatically learn the underlying patterns and structures present in the data. The primary components of a neural graph generative model include:

- **Graph Encoder**: The graph encoder takes an input graph and converts it into a low-dimensional latent representation. This step involves processing the graph's nodes, edges, and adjacency matrix using neural network layers to capture relevant features.
- **Latent Space:** The latent space is a lower-dimensional representation of the input graph obtained from the graph encoder. It captures the essential characteristics of the graph in a compact form.
- **Graph Decoder:** The graph decoder takes the latent space representation and decodes it to reconstruct the original graph. This involves using neural network layers to generate nodes, edges, and adjacency information based on the learned latent representation.
- **Training:** During the training phase, the model is fed with a set of observed graphs, and the graph encoder and decoder parameters are optimized to minimize the difference between the original graphs and their reconstructed counterparts.
- **Generation:** Once the model is trained, it can generate new graphs by sampling from the latent space and decoding the samples to obtain valid graph structures.

The team has expertise on the topic of graph generative models [12, 14]. In order to ensure the structural and functional integrity of the generated counterfactual graphs, we propose employing a generative model to learn from the graphs used during inductive counterfactual training.

Modified loss function to ensure feasibility: In the context of the Markov Decision Process (MDP), the transition from state X to state Y is influenced by the expected reward. In contrast to the reward function introduced in Thrust 1, an additional factor is introduced, namely the likelihood of the graph corresponding to state Y, as determined by the learned generative model. The formulation is as follows:

$$\mathcal{R}_v^t(a) = \frac{1}{\mathcal{L}_{v,pred}^{t+1} + \beta \times (t+1) + \frac{1}{LL(\mathcal{G}^{t+1})}}, \tag{1}$$

where $\mathcal{L}_{v,pred}^t = \sum_{\forall c \in \mathcal{C}} \mathbb{1}_{l(v)=c} \log\left(\Phi\left(\mathcal{G}^t, v, c\right)\right)$.

In this equation, $LL(\mathcal{G}^{t+1})$ represents the likelihood of the graph $\mathcal{G}^{t+1}$ being generated by the graph generative model learned from the training set. In other words, the transition probability is enhanced for states that correspond to graphs with a high likelihood of being generated by the generative model. The parameter α serves as a weight parameter to determine the relative importance of feasibility compared to the perturbation size and the probability of achieving the desired prediction class.

Few-shot graph generative modeling: For generative modeling, while we can use off-the-shelf generative models, such as [46] and [12], existing techniques requires huge volumes of training data to be accurate. To overcome this limitation, we propose to develop new algorithms for few-shot graph generative modeling, i.e., the ability to learn from small volumes of training data.

The proposed methodology for few shot graph generative modeling derives intuition from the observation that although the availability of graphs exhibiting a specific desired property may be limited, it may be possible to identify other graph repositories exhibiting similar properties. Hence, we can train model for general properties using large graph repositories but fine-tune the model for specific properties using small specific dataset. To elaborate, we may not have access to a large set of molecules exhibiting activity against COVID-19. However, million-scale repositories of chemical compounds are widely available [17], from which the broad characteristics of chemical compounds such as valency rules, correlated functional groups, etc. may be learned. Hence, potentially, the learning task from the smaller COVID-19 repository could be focused only on features that are unique to this set.

Empowered with this observation we propose the architecture outlined in Fig. 3. Given a set of auxiliary datasets $D_1, \cdots, D_{\mathcal{B}}$, first, we would first learn *initial* model parameters θ. θ is learned in a strategic manner such that, at inference time, when an *unseen target dataset* D_T containing a small number of graphs is provided as input, we can fine-tune θ to new θ_T where $p_{\theta_T}(D_T)$ best approximates the true distribution of D_T. Finally, to generate graphs, we sample from $p_{\theta_T}(D_T)$.

In order to learn a generative model over an auxiliary dataset of labeled graphs D, we first convert graphs to its *DFS encodings*. This choice is motivated by the observation that *minimum DFS codes*, which is an instance of DFS encoding, provides one-to-one mapping from graphs to sequences. In contrast, in BFS encoding, the same graph may have multiple sequence representations, and may be exponential with respect to the graph size. Consequently, one-to-one mapping is an attractive feature that our model can exploit, and as others have shown, it also improves the scalability and fidelity of graph generative modeling [13].

Once graphs are converted into sequences via *minimum DFS codes*, as shown in Fig. 3, *meta-learning* is conducted on the sequence representations to learn parameter set θ. To model sequences, we use LSTM as shown in Fig. 3. Finally, during target-adaptation phase, the target graph database D_T is converted to the equivalent sequence representation $\mathcal{S}_T$, followed by fine-tuning to learn θ_T. To generate graphs, we sample sequences from $p_{\theta_T}(\mathcal{S}_T)$, which are then converted to graphs. The conversion back from a sequence to its graph representation is trivial since our DFS-encoding enables one-to-one mapping. Hence, this conversion can be performed in $O(|E|)$ time, where E is the set of edges.

2.2.2 Task 2.2: Designing Counterfactual Explainer for Combinatorial problems

Motivation. Graph combinatorial problems pose greater challenges compared to graph and node classification tasks. Consequently, the models designed for learning combinatorial algorithms on graphs are

References

[1] C. Abrate and F. Bonchi. Counterfactual graphs for explainable classification of brain networks. In KDD, 2021.

[2] C. Abrate and F. Bonchi. Counterfactual graphs for explainable classification of brain networks. In KDD, page 2495–2504, 2021.

[3] A. Arora, S. Galhotra, and S. Ranu. Debunking the myths of influence maximization: An in-depth benchmarking study. In Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD '17, page 651–666, New York, NY, USA, 2017. Association for Computing Machinery.

[4] M. Bajaj, L. Chu, Z. Y. Xue, J. Pei, L. Wang, P. C.-H. Lam, and Y. Zhang. Robust counterfactual explanations on graph neural networks. arXiv preprint arXiv:2107.04086, 2021.

[5] M. Bajaj, L. Chu, Z. Y. Xue, J. Pei, L. Wang, P. C.-H. Lam, and Y. Zhang. Robust counterfactual explanations on graph neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems, 2021.

[6] R. Bhattoo, S. Ranu, and N. Krishnan. Learning articulated rigid body dynamics with lagrangian graph neural network. Advances in Neural Information Processing Systems, 35:29789–29800, 2022.

[7] R. Bhattoo, S. Ranu, and N. A. Krishnan. Learning the dynamics of particle-based systems with lagrangian graph neural networks. Machine Learning: Science and Technology, 2023.

[8] S. Bishnoi, R. Bhattoo, S. Ranu, and N. Krishnan. Enhancing the inductive biases of graph neural ode for modeling dynamical systems. ICLR, 2023.

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

[10] P. Chakraborty, S. Ranu, K. S. I. Mantri, and A. De. Learning and maximizing influence in social networks under capacity constraints. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pages 733–741, 2023.

[11] P. D. Dobson and A. J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. Journal of molecular biology, 330(4):771–783, 2003.

[12] N. Goyal, H. V. Jain, and S. Ranu. Graphgen: a scalable approach to domain-agnostic labeled graph generation. In Proceedings of The Web Conference 2020, pages 1253–1263, 2020.

[13] N. Goyal, H. V. Jain, and S. Ranu. Graphgen: a scalable approach to domain-agnostic labeled graph generation. In Proceedings of The Web Conference 2020, pages 1253–1263, 2020.

[14] S. Gupta, S. Manchanda, S. Bedathur, and S. Ranu. Tigger: Scalable generative modelling for temporal interaction graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 6819–6828, 2022.

[15] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc.

[16] Z. Huang, M. Kosan, S. Medya, S. Ranu, and A. Singh. Global counterfactual explainer for graph neural networks. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pages 141–149, 2023.

[17] J. J. Irwin and B. K. Shoichet. Zinc- a free database of commercially available compounds for virtual screening. Journal of chemical information and modeling, 45(1):177–182, 2005.

[18] S. Ivanov, S. Sviridov, and E. Burnaev. Understanding isomorphism bias in graph data sets. Geometric Learning and Graph Representations ICLR Workshop, 2019.

[19] M. Jiang, Z. Li, S. Zhang, S. Wang, X. Wang, Q. Yuan, and Z. Wei. Drug–target affinity prediction using graph neural network and contact maps. RSC advances, 10(35):20701–20712, 2020.

[20] J. Kakkad, J. Jannu, K. Sharma, C. Aggarwal, and S. Medya. A survey on explainability of graph neural networks. IEEE Data Engineering Bulletin, 2023.

[21] A. Karczmarz, A. Mukherjee, P. Sankowski, and P. Wygocki. Improved feature importance computations for tree models: Shapley vs. banzhaf. arXiv preprint arXiv:2108.04126, 2021.

[22] J. Kazius, R. McGuire, and R. Bursi. Derivation and validation of toxicophores for mutagenicity prediction. Journal of medicinal chemistry, 48(1):312–320, 2005.

[23] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song. Learning combinatorial optimization algorithms over graphs. Advances in neural information processing systems, 30, 2017.

[24] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[25] D. Ley, S. Mishra, and D. Magazzeni. Globe-ce: A translation-based approach for global counterfactual explanations. In ICML, 2023.

[26] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli. Graph matching networks for learning the similarity of graph structured objects. In International conference on machine learning, pages 3835–3845. PMLR, 2019.

[27] W. Lin, H. Lan, and B. Li. Generative causal explanations for graph neural networks. In International Conference on Machine Learning, pages 6666–6679. PMLR, 2021.

[28] A. Lucic, M. A. Ter Hoeve, G. Tolomei, M. De Rijke, and F. Silvestri. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In AISTATS, pages 4499–4511, 2022.

[29] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang. Parameterized explainer for graph neural network. arXiv preprint arXiv:2011.04573, 2020.

[30] S. Manchanda, A. Mittal, A. Dhawan, S. Medya, S. Ranu, and A. Singh. Gcomb: Learning budget-constrained combinatorial algorithms over billion-sized graphs. Advances in Neural Information Processing Systems, 33:20000–20011, 2020.

[31] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev. Reinforcement learning for combinatorial optimization: A survey. Computers & Operations Research, 134:105400, 2021.

[32] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. Information Retrieval, 3(2):127–163, 2000.

[33] S. Medya, T. Ma, A. Silva, and A. Singh. A game theoretic approach for core resilience. In International Joint Conferences on Artificial Intelligence Organization, 2020.

[34] S. Medya, S. Ranu, J. Vachery, and A. Singh. Noticeable network delay minimization via node upgrades. Proceedings of the VLDB Endowment, 11(9):988–1001, 2018.

[35] J. Pearl. Causality: Models, reasoning, and inference. Cambridge University Press, 2009.

[36] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. Advances in Neural Information Processing Systems, 35, 2022.

[37] R. Ranjan, S. Grover, S. Medya, V. Chakaravarthy, Y. Sabharwal, and S. Ranu. Greed: A neural framework for learning graph distance functions. In Advances in Neural Information Processing Systems, 2022.

[38] K. Rawal and H. Lakkaraju. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. 2020.

[39] K. Riesen and H. Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), pages 287–297. Springer, 2008.

[40] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. AI magazine, 29(3):93–93, 2008.

[41] K. Sharma, I. A. Gillani, S. Medya, S. Ranu, and A. Bagchi. Balance maximization in signed networks via edge deletions. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pages 752–760, 2021.

[42] K. Sharma, S. Verma, S. Medya, A. Bhattacharya, and S. Ranu. Task and model agnostic adversarial attack on graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 15091–15099, 2023.

[43] J. Tan, S. Geng, Z. Fu, Y. Ge, S. Xu, Y. Li, and Y. Zhang. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In WebConf, 2022.

[44] J. Tan, S. Geng, Z. Fu, Y. Ge, S. Xu, Y. Li, and Y. Zhang. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In Proceedings of the ACM Web Conference 2022, WWW '22, page 1018–1027, 2022.

[45] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. International Conference on Learning Representations, 2018. accepted as poster.

[46] C. Vignac, I. Krawczuk, A. Siraudin, B. Wang, V. Cevher, and P. Frossard. Digress: Discrete denoising diffusion for graph generation. In The Eleventh International Conference on Learning Representations, 2023.

[47] P. Voigt and A. Von dem Bussche. The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing, 10(3152676):10–5555, 2017.

[48] M. Vu and M. T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. Advances in neural information processing systems, 33:12225–12235, 2020.

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

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

[51] J. Xiong, Z. Xiong, K. Chen, H. Jiang, and M. Zheng. Graph neural networks for automated de novo drug design. Drug Discovery Today, 26(6):1382–1393, 2021.

[52] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In International Conference on Learning Representations, 2019.

[53] T. Yan and A. D. Procaccia. If you like shapley then you'll love the core. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 5751–5759, 2021.

[54] P. Yanardag and S. Vishwanathan. Deep graph kernels. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pages 1365–1374, 2015.

[55] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. Do transformers really perform badly for graph representation? In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems, 2021.

[56] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems, 32:9240, 2019.

[57] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In KDD, page 974–983, 2018.

[58] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In ICML, 2018, volume 80 of Proceedings of Machine Learning Research, pages 5694–5703. PMLR, 2018.

[59] H. Yuan, J. Tang, X. Hu, and S. Ji. Xgnn: Towards model-level explanations of graph neural networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 430–438, 2020.

[60] H. Yuan, H. Yu, S. Gui, and S. Ji. Explainability in graph neural networks: A taxonomic survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[61] T. Zhao, G. Liu, D. Wang, W. Yu, and M. Jiang. Learning from counterfactual links for link prediction. In International Conference on Machine Learning, pages 26911–26926. PMLR, 2022.